User's Manual

32

RX64M Group

User's Manual: Hardware

RENESAS 32-Bit MCU
RX Family / RX600 Series

Renesas Electronics
www.renesas.com

Rev.1.10 Oct 2016

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(2012.4)

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

   ¾ The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   ¾ The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   ¾ The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   ¾ When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   ¾ The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

How to Use This Manual

1. Objective and Target Users

This manual was written to explain the hardware functions and electrical characteristics of this LSI to the target users, i.e. those who will be using this LSI in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logic circuits, and microcomputers.
This manual is organized in the following items: an overview of the product, descriptions of the CPU, system control functions, and peripheral functions, electrical characteristics of the device, and usage notes.

When designing an application system that includes this LSI, take all points to note into account.
Points to note are given in their contexts and at the final part of each section, and in the section giving usage notes.

The list of revisions is a summary of major points of revision or addition for earlier versions. It does not cover all revised items. For details on the revised points, see the actual locations in the manual.

The following documents have been prepared for the RX64M Group. Before using any of the documents, please visit our web site to verify that you have the most up-to-date available version of the document.

Document Type	Contents	Document Title	Document No.
Data Sheet	Overview of hardware and electrical characteristics	RX64M Group Datasheet	R01DS0173EJ
User's manual: Hardware	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation	RX64M Group User's manual: Hardware	This User's manual
User's manual: Software	Detailed descriptions of the CPU and instruction set	RX Family RXv2 Instruction Set Architecture User's Manual: Software	R01US0071EJ
Flash Memory User's Manual: Hardware Interface	Detailed descriptions of the hardware interface of the flash memory	RX64M Group, RX71M Group Flash Memory User's Manual: Hardware Interface	R01UH0435EJ
Application Note	Notes on Printed Circuit Board Patterns	RX Family Hardware Design Guide	R01AN1411EJ
	Examples of applications and sample programs	—	—
Renesas Technical Update	Preliminary report on the specifications of a product, document, etc.	—	—

2. Description of Registers

Each register description includes a bit chart, illustrating the arrangement of bits, and a table of bits, describing the meanings of the bit settings. The standard format and notation for bit charts and tables are described below.

X.X.X ... Register

Address(es): xxxx xxxxh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	...[1:0]		...4	—	—	—	...0
Value after reset:	x	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	...0	.....	0: ..... 1: Setting prohibited	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	...4	.....	0: 1:	R
b6, b5	...[1:0]	.....	0 0: ..... 0 1: ..... Settings other than above are prohibited.	R/(W)*1
b7	—	Reserved	The read value is undefined. Writing to this bit has no effect.	R

(1) R/W: The bit or field is readable and writable.
R/(W): The bit or field is readable and writable. However, writing to this bit or field has some limitations. For details on the limitations, see the description or notes of respective registers.
R: The bit or field is readable. Writing to this bit or field has no effect.

(2) Reserved.
Use the specified value when writing to this bit or field; otherwise, the correct operation is not guaranteed.

(3) Setting prohibited. The correct operation is not guaranteed if such a setting is performed.

3. List of Abbreviations and Acronyms

Abbreviation	Full Form
ACIA	Asynchronous Communications Interface Adapter
bps	bits per second
CRC	Cyclic Redundancy Check
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
GSM	Global System for Mobile Communications
Hi-Z	High Impedance
IEBus	Inter Equipment Bus
I/O	Input/Output
IrDA	Infrared Data Association
LSB	Least Significant Bit
MSB	Most Significant Bit
NC	Non-Connect
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
SIM	Subscriber Identity Module
UART	Universal Asynchronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator

Contents

R01UH0377EJ0110
Rev.1.10
Oct 24, 2016

120-MHz 32-bit RX MCU, on-chip FPU, 240 DMIPS, up to 4-MB flash memory, 512-KB SRAM, various communications interfaces including IEEE 1588-compliant Ethernet MAC, full-speed USB 2.0 with battery charging, SD host interface (optional), quad SPI, and CAN, 12-bit A/D converter, RTC, encryption (optional), serial interface for audio, CMOS camera interface

Features

■ 32-bit RXv2 CPU core
- Max. operating frequency: 120 MHz
  Capable of 240 DMIPS in operation at 120 MHz
- Single precision 32-bit IEEE-754 floating point
- Two types of multiply-and-accumulation unit (between memories and between registers)
- 32-bit multiplier (fastest instruction execution takes one CPU clock cycle)
- Divider (fastest instruction execution takes two CPU clock cycles)
- Fast interrupt
- CISC Harvard architecture with 5-stage pipeline
- Variable-length instructions: Ultra-compact code
- Supports the memory protection unit (MPU)
- JTAG and FINE (one-line) debugging interfaces

■ Low-power design and architecture
- Operation from a single 2.7- to 3.6-V supply
- Low power consumption: A product that supports all peripheral functions draws only 0.3mA/MHz (Typ.).
- RTC is capable of operation from a dedicated power supply.
- Four low-power modes

■ On-chip code flash memory, no wait states
- Supports versions with up to 4 Mbytes of ROM
- 120-MHz operation, 8.3-ns read cycle (no wait states)
- User code is programmable by on-board or off-board programming.
- Programming/erasing as background operations (BGOs)

■ On-chip data flash memory
- 64 Kbytes, reprogrammable up to 100,000 times
- Programming/erasing as background operations (BGOs)

■ On-chip SRAM
- 512 Kbytes of SRAM (no wait states)
- 32 Kbytes of RAM with ECC (one wait state, single-error correction and double error detection)
- 8 Kbytes of standby RAM (backup on deep software standby)

■ Data transfer
- DMAC: 8 channels
- DTC
- EXDMAC: 2 channels
- DMAC for the Ethernet controller: 3 channels for 176- and 177-pin products; 2 channels for 100-, 144-, and 145-pin products

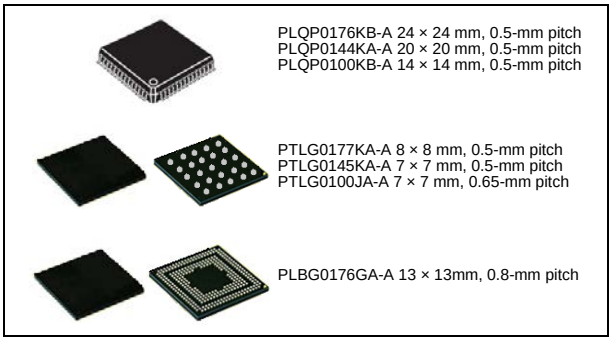
■ Reset and supply management
- Power-on reset (POR)
- Low voltage detection (LVD) with voltage settings

■ Clock functions
- External crystal resonator or internal PLL for operation at 8 to 24 MHz
- Internal 240-kHz LOCO and HOCO selectable from 16, 18, and 20 MHz
- 120-kHz clock for the IWDTa

■ Real-time clock
- Adjustment functions (30 seconds, leap year, and error)
- Real-time clock counting and binary counting modes are selectable
- Time capture function
  (for capturing times in response to event-signal input)

■ Independent watchdog timer
- 120-kHz (1/2 LOCO frequency) clock operation

■ Useful functions for IEC60730 compliance
- Oscillation-stoppage detection, frequency measurement, CRC, IWDTa, self-diagnostic function for the A/D converter, etc.
- Register write protection function can protect values in important registers against overwriting.



■ Various communications interfaces
- IEEE 1588-compliant Ethernet MAC (for 176- and 177-pin products: 2 modules)
- PHY layer for host/function or OTG controller (1) with full-speed USB 2.0 with battery charging transfer (only for 176- and 177-pin products)
- PHY layer (1) for host/function or OTG controller (1) with full-speed USB 2.0 transfer
- CAN (compliant with ISO11898-1), incorporating 32 mailboxes (up to 3 modules)
- SCIg and SCIh with multiple functionalities (up to 9)
  Choose from among asynchronous mode, clock-synchronous mode, smart-card interface mode, simplified SPI, simplified I^2C, and extended serial mode.
- SCIFA with 16-byte transmission and reception FIFOs (up to 4 interfaces)
- I^2C bus interface for transfer at up to 1 Mbps (up to 2 interfaces)
- Four-wire QSPI (1 interface) in addition to RSPIa (1 interface)
- Parallel data capture unit (PDC) for the CMOS camera interface (not in 100-pin products)
- SD host interface (optional: 1 interface) with a 1- or 4-bit SD bus for use with SD memory or SDIO

■ External address space
- Buses for full-speed data transfer (max. operating frequency of 60 MHz)
- 8 CS areas
- 8-, 16-, or 32-bit bus space is selectable per area
- Independent SDRAM area (128 Mbytes)

■ Up to 29 extended-function timers
- 16-bit TPUa, MTU3a, and GPTA: input capture, output compare, PWM waveform output
- 8-bit TMRa (4 channels), 16-bit CMT (4 channels), 32-bit CMTW (2 channels)

■ 12-bit A/D converter
- Two 12-bit units (8 channels for unit 0; 21 channels for unit 1)
- Self diagnosis
- Detection of analog input disconnection

■ 12-bit D/A converter: 2 channels
- On-chip operational amplifier output or direct input selectable

■ Temperature sensor for measuring temperature within the chip

■ Encryption (optional)
- AES (key lengths: 128, 192, and 256 bits)
- DES (key lengths: 56 bits (DES); 3 × 56 bits (T-DES))
- SHA (SHA-1 (128), SHA-2 (224 or 256), HMAC (160, 224, or 256))

■ Up to 127 pins for general I/O ports
- 5-V tolerance, open drain, input pull-up, switchable driving ability

■ Operating temp. range
- –40°C to +85°C

1. Overview

1.1 Outline of Specifications

Table 1.1 lists the specifications in outline, and Table 1.2 gives a comparison of the functions of products in different packages.

Table 1.1 shows the outline of maximum specifications, and the number of peripheral module channels differs depending on the pin number on the package and the code flash memory capacity. For details, see Table 1.2, Comparison of Functions for Different Packages.

Table 1.1 Outline of Specifications (1/9)

Classification	Module/Function	Description
CPU	CPU	• Maximum operating frequency: 120 MHz • 32-bit RX CPU (RXv2) • Minimum instruction execution time: One instruction per state (cycle of the system clock) • Address space: 4-Gbyte linear • Register set of the CPU General purpose: Sixteen 32-bit registers Control: Ten 32-bit registers Accumulator: Two 72-bit registers • Basic instructions: 75 • Floating-point instructions: 11 • DSP instructions: 23 • Addressing modes: 11 • Data arrangement Instructions: Little endian Data: Selectable as little endian or big endian • On-chip 32-bit multiplier: 32 × 32 → 64 bits • On-chip divider: 32 / 32 → 32 bits • Barrel shifter: 32 bits
	FPU	• Single precision (32-bit) floating point • Data types and floating-point exceptions in conformance with the IEEE754 standard
Memory	Code flash memory	• Capacity: 2 Mbytes, 2.5 Mbytes, 3 Mbytes, 4 Mbytes • 120 MHz, no-wait access • On-board programming: Four types • Off-board programming (parallel programmer mode) • The trusted memory (TM) function protects against the reading of programs from blocks 8 and 9.
	Data flash memory	• Capacity: 64 Kbytes • Programming/erasing: 100,000 times
	RAM	• Capacity: 512 Kbytes • 120 MHz, no-wait access • SED (single error detection)
	Unique ID	• 12-byte length ID unique to the device
	RAM with ECC	• Capacity: 32 Kbytes • 120 MHz, single wait access • SEC-DED (single error correction/double error detection)
	Standby RAM	• Capacity: 8 Kbytes • Operation synchronized with PCLKB: Up to 60 MHz, two-cycle access
Operating modes		• Operating modes by the mode-setting pins at the time of release from the reset state Single-chip mode Boot mode (for the SCI interface) Boot mode (for the USB interface) User boot mode • Selection of operating mode by register setting Single-chip mode, user boot mode On-chip ROM disabled extended mode On-chip ROM enabled extended mode • Endian selectable

Table 1.1 Outline of Specifications (2/9)

Classification	Module/Function	Description
Clock	Clock generation circuit	• Main clock oscillator, sub clock oscillator, low-speed/high-speed on-chip oscillator, PLL frequency synthesizer, and IWDT-dedicated on-chip oscillator • The peripheral module clocks can be set to frequencies above that of the system clock. • Main-clock oscillation stoppage detection • Separate frequency-division and multiplication settings for the system clock (ICLK), peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD), flash-IF clock (FCLK) and external bus clock (BCLK) The CPU and other bus masters run in synchronization with the system clock (ICLK): Up to 120 MHz Peripheral modules of MTU3, GPT, RSPI, SCIFA, USBA, ETHERC, EPTPC, EDMAC, and AES run in synchronization with PCLKA, which operates at up to 120 MHz. Other peripheral modules run in synchronization with PCLKB: Up to 60 MHz ADCLK in the SD12AD (unit 0) runs in synchronization with PCLKC: Up to 60 MHz ADCLK in the SD12AD (unit 1) runs in synchronization with PCLKD: Up to 60 MHz Flash IF run in synchronization with the flash-IF clock (FCLK): Up to 60 MHz Devices connected to the external bus run in synchronization with the external bus clock (BCLK): Up to 60 MHz • Multiplication is possible with using the high-speed on-chip oscillator (HOCO) as a reference clock of the PLL circuit
Reset		Nine types of reset • RES# pin reset: Generated when the RES# pin is driven low. • Power-on reset: Generated when the RES# pin is driven high and VCC = AVCC0 = AVCC1 rises. • Voltage-monitoring 0 reset: Generated when VCC = AVCC0 = AVCC1 falls. • Voltage-monitoring 1 reset: Generated when VCC = AVCC0 = AVCC1 falls. • Voltage-monitoring 2 reset: Generated when VCC = AVCC0 = AVCC1 falls. • Deep software standby reset: Generated in response to an interrupt to trigger release from deep software standby. • Independent watchdog timer reset: Generated when the independent watchdog timer underflows, or a refresh error occurs. • Watchdog timer reset: Generated when the watchdog timer underflows, or a refresh error occurs. • Software reset: Generated by register setting.
Power-on reset		If the RES# pin is at the high level when power is supplied, an internal reset is generated. After VCC = AVCC0 = AVCC1 has exceeded the voltage detection level and the specified period has elapsed, the reset is cancelled.
Voltage detection circuit (LVDA)		Monitors the voltage being input to the VCC = AVCC0 = AVCC1 pins and generates an internal reset or internal interrupt. • Voltage detection circuit 0 Capable of generating an internal reset The option-setting memory can be used to select enabling or disabling of the reset. Voltage detection level: Selectable from three different levels (2.94 V, 2.87 V, and 2.80 V) • Voltage detection circuits 1 and 2 Voltage detection level: Selectable from three different levels (2.99 V, 2.92 V, and 2.85 V) Digital filtering (1/2, 1/4, 1/8, and 1/16 LOCO frequency) Capable of generating an internal reset • Two types of timing are selectable for release from reset An internal interrupt can be requested. • Detection of voltage rising above and falling below thresholds is selectable. • Maskable or non-maskable interrupt is selectable Voltage detection monitoring Event linking
Low power consumption	Low power consumption facilities	• Module stop function • Four low power consumption modes Sleep mode, all-module clock stop mode, software standby mode, and deep software standby mode
	Battery backup function	• When the voltage on the VCC pin drops, battery power from the VBATT pin is supplied to keep the real-time clock (RTC) operating.

Table 1.1 Outline of Specifications (3/9)

Classification	Module/Function	Description
Interrupt	Interrupt controller (ICUA)	• Peripheral function interrupts: 293 sources • External interrupts: 16 (pins IRQ0 to IRQ15) • Software interrupts: 2 sources • Non-maskable interrupts: 7 sources • Sixteen levels specifiable for the order of priority • Method of interrupt source selection: The interrupt vectors consist of 256 vectors (128 sources are fixed. The remaining 128 vectors are selected from among the other 156 sources.)
External bus extension		• The external address space can be divided into eight areas (CS0 to CS7), each with independent control of access settings. Capacity of each area: 16 Mbytes (CS0 to CS7) A chip-select signal (CS0# to CS7#) can be output for each area. Each area is specifiable as an 8-, 16-, or 32-bit bus space. The data arrangement in each area is selectable as little or big endian (only for data). • SDRAM interface connectable • Bus format: Separate bus, multiplex bus • Wait control • Write buffer facility
DMA	DMA controller (DMACAa)	• 8 channels • Three transfer modes: Normal transfer, repeat transfer, and block transfer • Request sources: Software trigger, external interrupts, and interrupt requests from peripheral functions
	EXDMA controller (EXDMACa)	• 2 channels Four transfer modes: Normal transfer, repeat transfer, block transfer, and cluster transfer • Single-address transfer enabled with the EDACKn signal • Request sources: Software trigger, external DMA requests (EDREQn), and interrupt requests from peripheral functions
	Data transfer controller (DTCa)	• Three transfer modes: Normal transfer, repeat transfer, and block transfer • Request sources: External interrupts and interrupt requests from peripheral functions
I/O ports	Programmable I/O ports	• I/O ports for the 177-pin TFLGA, 176-pin LFBGA, and 176-pin LFQFP I/O pins: 127 Input pin: 1 Pull-up resistors: 127 Open-drain outputs: 127 5-V tolerance: 19 • I/O ports for the 145-pin TFLGA and 144-pin LFQFP I/O pins: 111 Input pin: 1 Pull-up resistors: 111 Open-drain outputs: 111 5-V tolerance: 18 • I/O ports for the 100-pin TFLGA and 100-pin LFQFP I/O pins: 78 Input pin: 1 Pull-up resistors: 78 Open-drain outputs: 78 5-V tolerance: 17
Event link controller (ELC)		• Event signals such as interrupt request signals can be interlinked with the operation of functions such as timer counting, eliminating the need for intervention by the CPU to control the functions. • 119 internal event signals can be freely combined for interlinked operation with connected functions. • Event signals from peripheral modules can be used to change the states of output pins (of ports B and E). • Changes in the states of pins (of ports B and E) being used as inputs can be interlinked with the operation of peripheral modules.

Table 1.1 Outline of Specifications (4/9)

Classification	Module/Function	Description
Timers	16-bit timer pulse unit (TPUa)	• (16 bits × 6 channels) × 1 unit • Maximum of 16 pulse-input/output possible • Select from among seven or eight counter-input clock signals for each channel • Input capture/output compare function • Output of PWM waveforms in up to 15 phases in PWM mode • Support for buffered operation, phase-counting mode (two phase encoder input) and cascade-connected operation (32 bits × 2 channels) depending on the channel. • PPG output trigger can be generated • Capable of generating conversion start triggers for the A/D converters • Digital filtering of signals from the input capture pins • Event linking by the ELC
Timers	Multifunction timer pulse unit (MTU3a)	• 9 channels (16 bits × 8 channels, 32 bits × 1 channel) • Maximum of 28 pulse-input/output and 3 pulse-input possible • Select from among 14 counter-input clock signals for each channel (PCLKA/1, PCLKA/2, PCLKA/4, PCLKA/8, PCLKA/16, PCLKA/32, PCLKA/64, PCLKA/256, PCLKA/1024, MTCLKA, MTCLKB, MTCLKC, MTCLKD, MTIOC1A) 14 of the signals are available for channel 0, 12 are available for channel 2, 11 are available for channels 1, 3, 4, 6 to 8, and 10 are available for channel 5. • Input capture function • 39 output compare/input capture registers • Counter clear operation (synchronous clearing by compare match/input capture) • Simultaneous writing to multiple timer counters (TCNT) • Simultaneous register input/output by synchronous counter operation • Buffered operation • Support for cascade-connected operation • 43 interrupt sources • Automatic transfer of register data • Pulse output mode Toggle/PWM/complementary PWM/reset-synchronized PWM • Complementary PWM output mode Outputs non-overlapping waveforms for controlling 3-phase inverters Automatic specification of dead times PWM duty cycle: Selectable as any value from 0% to 100% Delay can be applied to requests for A/D conversion. Non-generation of interrupt requests at peak or trough values of counters can be selected. Double buffer configuration • Reset synchronous PWM mode Three phases of positive and negative PWM waveforms can be output with desired duty cycles. • Phase-counting mode: 16-bit mode (channels 1 and 2); 32-bit mode (channels 1 and 2) • Counter functionality for dead-time compensation • Generation of triggers for A/D converter conversion • A/D converter start triggers can be skipped • Digital filter function for signals on the input capture and external counter clock pins • PPG output trigger can be generated • Event linking by the ELC
	Port output enable 3 (POE3a)	• Control of the high-impedance state of the MTU3/GPT's waveform output pins • 5 pins for input from signal sources: POE0, POE4, POE8, POE10, POE11 • Initiation on detection of short-circuited outputs (detection of simultaneous PWM output to the active level) • Initiation by oscillation-stoppage detection or software • Additional programming of output control target pins is enabled

Table 1.1 Outline of Specifications (5/9)

Classification	Module/Function	Description
Timers	General PWM timer (GPTA)	• 16 bits × 4 channels • Counting up or down (saw-wave), counting up and down (triangle-wave) selectable for all channels • Four clock sources independently selectable for all channels (PCLKA/1, PCLKA/4, PCLKA/8, PCLKA/16) • 2 input/output pins per channel • 2 output compare/input capture registers per channel • For the 2 output compare/input capture registers of each channel, 4 registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use. • In output compare operation, buffer switching can be at peaks or troughs, enabling the generation of laterally asymmetrically PWM waveforms. • Registers for setting up frame intervals on each channel (with capability for generating interrupts on overflow or underflow) • Synchronizable operation of the several counters • Modes of synchronized operation (synchronized, or displaced by desired times for phase shifting) • Generation of dead times in PWM operation • Through combination of three counters, generation of automatic three-phase PWM waveforms incorporating dead times • Starting, clearing, and stopping counters in response to external or internal triggers • Internal trigger sources: output of the internal comparator detection, software, and compare-match • Digital filter function for signals on the input capture and external trigger pins • Event linking by the ELC
	Programmable pulse generator (PPG)	• (4 bits × 4 groups) × 2 units • Pulse output with the MTU or TPU output as a trigger • Maximum of 32 pulse-output possible
	8-bit timers (TMRb)	• (8 bits × 2 channels) × 2 units • Select from among seven internal clock signals (PCLKB/1, PCLKB/2, PCLKB/8, PCLKB/32, PCLKB/64, PCLKB/1024, PCLKB/8192) and one external clock signal • Capable of output of pulse trains with desired duty cycles or of PWM signals • The 2 channels of each unit can be cascaded to create a 16-bit timer • Generation of triggers for A/D converter conversion • Capable of generating baud-rate clocks for SCI5, SCI6, and SCI12 • Event linking by the ELC
	Compare match timer (CMT)	• (16 bits × 2 channels) × 2 units • Select from among four internal clock signals (PCLKB/8, PCLKB/32, PCLKB/128, PCLKB/512) • Event linking by the ELC
	Compare match timer W (CMTW)	• (32 bits × 1 channel) × 2 units • Compare-match, input-capture input, and output-comparison output are available. • Select from among four internal clock signals (PCLKB/8, PCLKB/32, PCLKB/128, PCLKB/512) • Interrupt requests can be output in response to compare-match, input-capture, and output-comparison events. • Event linking by the ELC
	Realtime clock (RTCd)	• Clock sources: Main clock, sub clock • Selection of the 32-bit binary count in time count/second unit possible • Clock and calendar functions • Interrupt sources: Alarm interrupt, periodic interrupt, and carry interrupt • Battery backup operation • Time-capture facility for three values • Event linking by the ELC
	Watchdog timer (WDTA)	• 14 bits × 1 channel • Select from among 6 counter-input clock signals (PCLKB/4, PCLKB/64, PCLKB/128, PCLKB/512, PCLKB/2048, PCLKB/8192)
	Independent watchdog timer (IWDTa)	• 14 bits × 1 channel • Counter-input clock: IWDT-dedicated on-chip oscillator • Dedicated clock/1, dedicated clock/16, dedicated clock/32, dedicated clock/64, dedicated clock/128, dedicated clock/256 • Window function: The positions where the window starts and ends are specifiable (the window defines the timing with which refreshing is enabled and disabled). • Event linking by the ELC

Table 1.1 Outline of Specifications (6/9)

Classification	Module/Function	Description
Communication function	Ethernet controller (ETHERC)	• 2 channels • Input and output of Ethernet/IEEE 802.3 frames • Transfer at 10 or 100 Mbps • Full- and half-duplex modes • MII (Media Independent Interface) or RMII (Reduced Media Independent Interface) as defined in IEEE 802.3u • Detection of Magic Packets™*1 or output of a "wake-on-LAN" signal (WOL) • Compliance with flow control as defined in IEEE 802.3x standards • Filtering of multicast frames • Direct transfer of frames between two channels by cut-through
	PTP controller for Ethernet controller (EPTPC)	• A block compatible with the IEEE 1588 standard is connected to the Ethernet controller (ETHERC). • Matching with a time stamp can start counting by MTU3 and the GPT.
	DMA controller for Ethernet controller (EDMACa)	• 3 channels (the round-robin method determines the priority of the channels) 2 channels for ETHERC; 1 channel for EPTPC • Alleviation of CPU load by the descriptor control method • Transmission FIFO: 2 Kbytes; Reception FIFO: 4 Kbytes
	USB 2.0 FS host/ function module (USBb)	• Includes a UDC (USB Device Controller) and transceiver for USB 2.0 FS • One port • Compliance with the USB 2.0 specification • Transfer rate: Full speed (12 Mbps), low speed (1.5 Mbps) (host only) • Both self-power mode and bus power are supported • OTG (On the Go) operation is possible (low-speed is not supported) • Incorporates 2 Kbytes of RAM as a transfer buffer • External pull-up and pull-down resistors are not required
	USB 2.0 FS host/ function module with battery charging (USBA)	• Includes a UDC (USB Device Controller) and transceiver for USB 2.0 FS • One port (only in 176-pin devices) • Compliance with the USB 2.0 specification • Transfer rate: Full speed (12 Mbps), low speed (1.5 Mbps) (host only) • Both self-power mode and bus power are supported • OTG (On the Go) operation is possible (low-speed is not supported) • Incorporates 8.5 Kbytes of RAM as a transfer buffer • External pull-up and pull-down resistors are not required
	Serial communications interfaces (SCIg, SCIh)	• 9 channels (SCIg: 8 channels + SCIh: 1 channel) • SCIg Serial communications modes: Asynchronous, clock synchronous, and smart-card interface Multi-processor function On-chip baud rate generator allows selection of the desired bit rate Choice of LSB-first or MSB-first transfer Average transfer rate clock can be input from TMR timers for SCI5, SCI6, and SCI12 Start-bit detection: Level or edge detection is selectable. Simple I^2C Simple SPI 9-bit transfer mode Bit rate modulation Double-speed mode Event linking by the ELC (only on channel 5) • SCIh (The following functions are added to SCIg) Supports the serial communications protocol, which contains the start frame and information frame Supports the LIN format
	Serial communications interface with FIFO (SCIFA)	• 4 channels • Methods of transfer: Asynchronous and clock synchronous • Desired bit rates can be selected from the internal baud rate generators. • LSB or MSB first is selectable. • Both the transmission and reception sections are equipped with 16-byte FIFO buffers, allowing continuous transmission and reception. • Bit rate modulation • Double-speed mode

Table 1.1 Outline of Specifications (7/9)

Classification	Module/Function	Description
Communication function	I²C bus interface (RIICa)	• 2 channels (only channel 0 can be used in fast-mode plus) Communication formats I²C bus format/SMBus format Supports the multi-master Max. transfer rate: 1 Mbps (channel 0) • Event linking by the ELC
	CAN module (CAN)	• 3 channels • Compliance with the ISO11898-1 specification (standard frame and extended frame) • 32 mailboxes per channel
	Serial peripheral interface (RSPIa)	• 1 channel • RSPI transfer facility Using the MOSI (master out, slave in), MISO (master in, slave out), SSL (slave select), and RSPCK (RSPI clock) signals enables serial transfer through SPI operation (four lines) or clock-synchronous operation (three lines) Capable of handling serial transfer as a master or slave • Data formats Switching between MSB first and LSB first The number of bits in each transfer can be changed to any number of bits from 8 to 16, or to 20, 24, or 32 bits. 128-bit buffers for transmission and reception Up to four frames can be transmitted or received in a single transfer operation (with each frame having up to 32 bits) • Buffered structure Double buffers for both transmission and reception • RSPCK can be stopped with the receive buffer full for master reception. • Event linking by the ELC
	Quad serial peripheral interface (QSPI)	• 1 channel • Connectable with serial flash memory equipped with multiple input and output lines (i.e. for single, dual, or quad operation) • Programmable bit length and selectable active sense and phase of the clock signal • Sequential execution of transfer • LSB or MSB first is selectable.
Serial sound interface (SSI)		• 2 channels • Full-duplex transfer is possible (only on channel 0). • Support for multiple audio formats • Support for master or slave operation • Bit clock frequency is selectable from four different types (16 fs, 32 fs, 48 fs, and 64 fs). • Support for 8-/16-/18-/20-/22-/24 bit data formats • Internal 8-stage FIFO for transmission and reception • Stopping SSIWS when data transfer is stopped is selectable.
Sampling rate converter (SRC)		• 1 channel • Data formats: 32-bit stereo (16 bits for the left, 16 bits for the right) and 16-bit monaural. • Input sampling rates: 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48 kHz • Output sampling rates: 32, 44.1, 48, 8*2 or 16 kHz*2
SD host interface (SDHI)*4		• 1 channel • Transfer speed: Supports high-speed mode (15 MB/s) and default speed mode (10 MB/s) • One interface for SD memory and I/O cards (supporting 1- and 4-bit SD buses) • SD specifications Part 1: Physical Layer Specification Ver. 3.01 compliant (DDR not supported) Part E1: SDIO Specification Ver. 3.00 • Error checking: CRC7 for commands and CRC16 for data • Interrupt requests: Card access interrupt, SDIO access interrupt, card detection interrupt • DMA transfer requests: SD_BUF write and SD_BUF read • Support for card detection and write protection
MMC host interface (MMCIF)		• 1 channel • Transfer speed: Supports high-speed mode (30 MB/s) and Backward-compatible mode (25 MB/s) • Compliant with JEDEC STANDARD JESD84-A441 (DDR is not supported) • Interface for Multimedia Cards (MMCs) • Device buses: Support for 1-, 4-, and 8-bit MMC buses • Interrupt requests: Card detection interrupt, error/timeout interrupt, normal operation interrupt • DMA transfer requests: CE_DATA write and CE_DATA read • Support for card detection, boot operation, high priority interrupt (HPI)

Table 1.1 Outline of Specifications (8/9)

Classification	Module/Function	Description
Parallel data capture unit (PDC)		• 1 channel • Acquisition of synchronization through external 8-bit horizontal and vertical synchronization signals • Setting of the image size when clipping of the output for a one-frame image is required
12-bit A/D converter (S12ADC)		• 12 bits × 2 units (unit 0: 8 channels; unit 1: 21 channels) • 12-bit resolution (switchable between 8, 10, and 12 bits) • Conversion time 0.48 µs per channel (for 12-bit conversion) 0.45 µs per channel (for 10-bit conversion) 0.42 µs per channel (for 8-bit conversion) • Operating mode Scan mode (single scan mode, continuous scan mode, or group scan mode) Group A priority control (only for group scan mode) • Sample-and-hold function Common sample-and-hold circuit included In addition, channel-dedicated sample-and-hold function (3ch: in unit 0 only) included • Sampling variable Sampling time can be set up for each channel. • Digital comparison Method: Comparison to detect voltages above or below thresholds and window comparison Measurement: Comparison of two results of conversion or comparison of a value in the comparison register and a result of conversion • Self-diagnostic function The self-diagnostic function internally generates three analog input voltages (unit 0: VREFL0, VREFH0 × 1/2, VREFH0; unit 1: AVSS1, AVCC1 × 1/2, AVCC1) • Double trigger mode (A/D conversion data duplicated) • Detection of analog input disconnection • Three ways to start A/D conversion Software trigger, timer (MTU3, GPT, TMR, TPU) trigger, external trigger • Event linking by the ELC
12-bit D/A converter (R12DA)		• 2 channels • 12-bit resolution • Output voltage: 0.2 V to AVCC1 – 0.2 V (amplifier output), 0 V to AVCC1 (direct output) • Output via an amplifier or direct output can be selected. • Event linking by the ELC
Temperature sensor		• 1 channel • Relative precision: ±1°C • The voltage of the temperature is converted into a digital value by the 12-bit A/D converter (unit 1).
Safety	Memory protection unit (MPU)	• Protection area: Eight areas (max.) can be specified in the range from 0000 0000h to FFFF FFFFh. • Minimum protection unit: 16 bytes • Reading from, writing to, and enabling the execution access can be specified for each area. • An address exception occurs when the detected access is not in the permitted area.
	Trusted Memory (TM) Function	• Protects against the reading of programs from blocks 8 and 9 of the code flash memory • Instruction fetching by the CPU is the only form of access to these areas when the TM function is enabled.
	Register write protection function	• Protects important registers from being overwritten for in case a program runs out of control.
	CRC calculator (CRC)	• CRC code generation for arbitrary amounts of data in 8-bit units • Select any of three generating polynomials: $X^8 + X^2 + X + 1$, $X^{16} + X^{15} + X^2 + 1$, or $X^{16} + X^{12} + X^5 + 1$ • Generation of CRC codes for use with LSB-first or MSB-first communications is selectable
	Main clock oscillation stop function	• Main clock oscillation stop detection: Available
	Clock frequency accuracy measurement circuit (CAC)	• Monitors the clock output from the main clock oscillator, sub-clock oscillator, low- and high-speed on-chip oscillators, the PLL frequency synthesizer, IWDT-dedicated on-chip oscillator, and PCLKB, and generates interrupts when the setting range is exceeded.
	Data operation circuit (DOC)	• The function to compare, add, or subtract 16-bit data

Table 1.1 Outline of Specifications (9/9)

Classification	Module/Function	Description
Encryption function	AES*3	• Key lengths: 128, 192, and 256 bits • Support for CBC, ECB, CFB, OFB, CTR, and CMAC operating modes • Speed of calculations: 128-bit key length in 22 cycles 192-bit key length in 26 cycles 256-bit key length in 30 cycles • Compliant with FIPS PUB 197
	DES*3	• Key lengths: 56 bits (DES)/3 × 56 bits (T-DES) • Support for DES and triple DES • Support for ECB and CBC operating modes • Speed of calculations: 6 clock cycles in single DES mode 14 clock cycles in triple DES mode • Compliant with FIPS PUB 46-3 • Compliant with FIPS PUB 81
	SHA*3	• Support for SHA-1 (128), SHA-2 (224 or 256), and HMAC (160, 224, or 256) • Speed of calculations: 50 clock cycles in SHA-1 mode 42 clock cycles in SHA-224 mode 42 clock cycles in SHA-256 mode • Compliant with SHA as defined in FIPS PUB 180-1 and -2 • Compliant with HMAC as defined in FIPS PUB 198
	True random number generator (RNG)*3	• Length of random numbers: 16 bits • Generation of random-number-generated interrupts after a number is generated • Random number generation time: 3.6 ms (typ)
Operating frequency		Up to 120 MHz
Power supply voltage		VCC = AVCC0 = AVCC1 = VCC_USB = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0, VCC_USBA = AVCC_USBA = 2.7 to 3.6 V, V_{BATT} = 2.0 to 3.6 V
Operating temperature		D-version: –40 to +85°C G-version: –40 to +105°C (in planning)
Package		177-pin TFLGA (PTLG0177KA-A) 176-pin LFBGA (PLBG0176GA-A) 176-pin LFQFP (PLQP0176KB-A) 145-pin TFLGA (PTLG0145KA-A) 144-pin LFQFP (PLQP0144KA-A) 100-pin TFLGA (PTLG0100JA-A) 100-pin LFQFP (PLQP0100KB-A)
On-chip debugging system		• E1 emulator (JTAG and FINE interfaces) • E20 emulator (JTAG interface)

Note 1. Magic Packet™ is a registered trademark of Advanced Micro Devices, Inc.
Note 2. Setting is only possible when the input sampling rate 44.1 kHz is selected.
Note 3. The product part number differs according to whether or not it supports encryption.
Note 4. The product part number differs according to whether or not it includes an SDHI (SD host interface).

Table 1.2 Comparison of Functions for Different Packages (1/2)

Functions		RX64M Group		
Package		**177 Pins, 176 Pins**	**145 Pins, 144 Pins**	**100 Pins**
External bus	External bus width	32 bits	16 bits	
	SDRAM area controller	Available		Not supported
DMA	DMA controller	Ch. 0 to 7		
	Data transfer controller	Available		
	EXDMA controller	Ch. 0 and 1		
Timers	16-bit timer pulse unit	Ch. 0 to 5		
	Multi-function timer pulse unit 3	Ch. 0 to 8		
	General-purpose PWM timer	Ch. 0 to 3		
	Port output enable 3	Available		
	Programmable pulse generator	Ch. 0 and 1		
	8-bit timers	Ch. 0 to 3		
	Compare match timer	Ch. 0 to 3		
	Compare match timer W	Ch. 0 and 1		
	Realtime clock	Available		
	Watchdog timer	Available		
	Independent watchdog timer	Available		
Communication function	Ethernet controller	Ch. 0 and 1	Ch. 0	
	PTP controller for ethernet controller	Available		
	DMAC controller for ethernet	Ch. 0 and 1 (ETHERC) Ch. 2 (EPTPC)	Ch. 0 (ETHERC) and 2 (EPTPC)	
	USB 2.0 FS host/function module	Ch. 0		
	USB 2.0 FS host/function module with battery charging	Available	Not supported	
	Serial communications interfaces (SCIg)	Ch. 0 to 7		Ch. 0 to 3, 5 and 6
	Serial communications interfaces (SCIh)	Ch. 12		
	Serial communications interfaces with FIFO	Ch. 8 to 11		Ch. 8 and 9
	I^2C bus interfaces	Ch. 0 and 2		
	Serial peripheral interface	Ch. 0		
	CAN module	Ch. 0 to 2		Ch. 0 and 1
	Quad serial peripheral interface	Ch. 0		
	Serial sound interfaces	Ch. 0 and 1		
	Sampling rate converter	Available		
	SD host interface	Ch. 0		
	MMC host interface	Ch. 0		
	Parallel data capture unit	Available		Not supported
12-bit A/D converter		AN000 to 007 (unit 0: 8 channels) AN100 to 120 (unit 1: 21 channels)		AN000 to 007 (unit 0: 8 channels) AN100 to 113 (unit 1: 14 channels)
12-bit D/A converter		Ch. 0 and 1		Ch. 1
Temperature sensor		Available		
CRC calculator		Available		
Data operation circuit		Available		
Clock frequency accuracy measurement circuit		Available		
AES		Available		

Table 1.2 Comparison of Functions for Different Packages (2/2)

Functions	RX64M Group		
Package	**177 Pins, 176 Pins**	**145 Pins, 144 Pins**	**100 Pins**
DES	Available		
SHA	Available		
RNG	Available		
Event link controller	Available		

1.2 List of Products

Table 1.3 is a list of products, and Figure 1.1 shows how to read the product part no.

Table 1.3 List of Products (1/3)

Group	Part No.	Package	Code Flash Memory Capacity	RAM Capacity	Data Flash Memory Capacity	Operating Frequency (Max.)	Encryption Module	SDHI
RX64M	R5F564MLCDFC	PLQP0176KB-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MLDDFC	PLQP0176KB-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MLGDFC	PLQP0176KB-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MLHDFC	PLQP0176KB-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MJCDFC	PLQP0176KB-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MJDDFC	PLQP0176KB-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MJGDFC	PLQP0176KB-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MJHDFC	PLQP0176KB-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MGCDFC	PLQP0176KB-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MGDDFC	PLQP0176KB-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MGGDFC	PLQP0176KB-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MGHDFC	PLQP0176KB-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MFCDFC	PLQP0176KB-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MFDDFC	PLQP0176KB-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MFGDFC	PLQP0176KB-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MFHDFC	PLQP0176KB-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MLCDFB	PLQP0144KA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MLDDFB	PLQP0144KA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MLGDFB	PLQP0144KA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MLHDFB	PLQP0144KA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MJCDFB	PLQP0144KA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MJDDFB	PLQP0144KA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MJGDFB	PLQP0144KA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MJHDFB	PLQP0144KA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MGCDFB	PLQP0144KA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MGDDFB	PLQP0144KA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MGGDFB	PLQP0144KA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MGHDFB	PLQP0144KA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MFCDFB	PLQP0144KA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MFDDFB	PLQP0144KA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MFGDFB	PLQP0144KA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MFHDFB	PLQP0144KA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MLCDFP	PLQP0100KB-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MLDDFP	PLQP0100KB-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MLGDFP	PLQP0100KB-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MLHDFP	PLQP0100KB-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MJCDFP	PLQP0100KB-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MJDDFP	PLQP0100KB-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MJGDFP	PLQP0100KB-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MJHDFP	PLQP0100KB-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MGCDFP	PLQP0100KB-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MGDDFP	PLQP0100KB-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MGGDFP	PLQP0100KB-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MGHDFP	PLQP0100KB-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available

Table 1.3 List of Products (2/3)

Group	Part No.	Package	Code Flash Memory Capacity	RAM Capacity	Data Flash Memory Capacity	Operating Frequency (Max.)	Encryption Module	SDHI
RX64M	R5F564MFCDFP	PLQP0100KB-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MFDDFP	PLQP0100KB-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MFGDFP	PLQP0100KB-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MFHDFP	PLQP0100KB-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MLCDBG	PLBG0176GA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MLDDBG	PLBG0176GA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MLGDBG	PLBG0176GA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MLHDBG	PLBG0176GA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MJCDBG	PLBG0176GA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MJDDBG	PLBG0176GA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MJGDBG	PLBG0176GA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MJHDBG	PLBG0176GA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MGCDBG	PLBG0176GA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MGDDBG	PLBG0176GA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MGGDBG	PLBG0176GA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MGHDBG	PLBG0176GA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MFCDBG	PLBG0176GA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MFDDBG	PLBG0176GA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MFGDBG	PLBG0176GA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MFHDBG	PLBG0176GA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MLCDLC	PTLG0177KA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MLDDLC	PTLG0177KA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MLGDLC	PTLG0177KA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MLHDLC	PTLG0177KA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MJCDLC	PTLG0177KA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MJDDLC	PTLG0177KA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MJGDLC	PTLG0177KA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MJHDLC	PTLG0177KA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MGCDLC	PTLG0177KA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MGDDLC	PTLG0177KA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MGGDLC	PTLG0177KA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MGHDLC	PTLG0177KA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MFCDLC	PTLG0177KA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MFDDLC	PTLG0177KA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MFGDLC	PTLG0177KA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MFHDLC	PTLG0177KA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MLCDLK	PTLG0145KA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MLDDLK	PTLG0145KA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MLGDLK	PTLG0145KA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MLHDLK	PTLG0145KA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MJCDLK	PTLG0145KA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MJDDLK	PTLG0145KA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MJGDLK	PTLG0145KA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MJHDLK	PTLG0145KA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MGCDLK	PTLG0145KA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MGDDLK	PTLG0145KA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MGGDLK	PTLG0145KA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MGHDLK	PTLG0145KA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available

Table 1.3 List of Products (3/3)

Group	Part No.	Package	Code Flash Memory Capacity	RAM Capacity	Data Flash Memory Capacity	Operating Frequency (Max.)	Encryption Module	SDHI
RX64M	R5F564MFCDLK	PTLG0145KA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MFDDLK	PTLG0145KA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MFGDLK	PTLG0145KA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MFHDLK	PTLG0145KA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MLCDLJ	PTLG0100JA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MLDDLJ	PTLG0100JA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MLGDLJ	PTLG0100JA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MLHDLJ	PTLG0100JA-A	4 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MJCDLJ	PTLG0100JA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MJDDLJ	PTLG0100JA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MJGDLJ	PTLG0100JA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MJHDLJ	PTLG0100JA-A	3 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MGCDLJ	PTLG0100JA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MGDDLJ	PTLG0100JA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MGGDLJ	PTLG0100JA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MGHDLJ	PTLG0100JA-A	2.5 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available
	R5F564MFCDLJ	PTLG0100JA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Not supported
	R5F564MFDDLJ	PTLG0100JA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Not supported	Available
	R5F564MFGDLJ	PTLG0100JA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Not supported
	R5F564MFHDLJ	PTLG0100JA-A	2 Mbytes	512 Kbytes	64 Kbytes	120 MHz	Available	Available

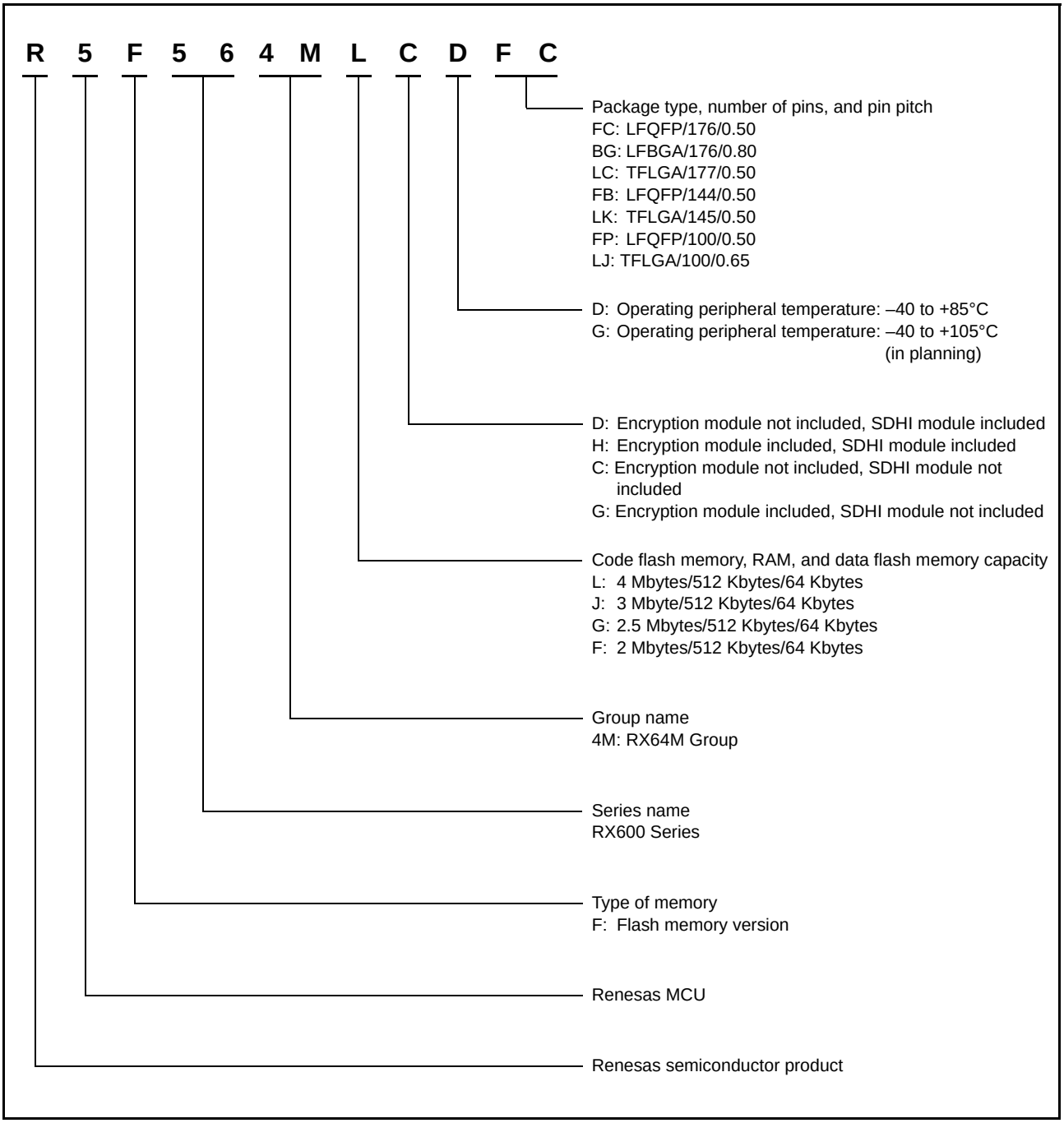
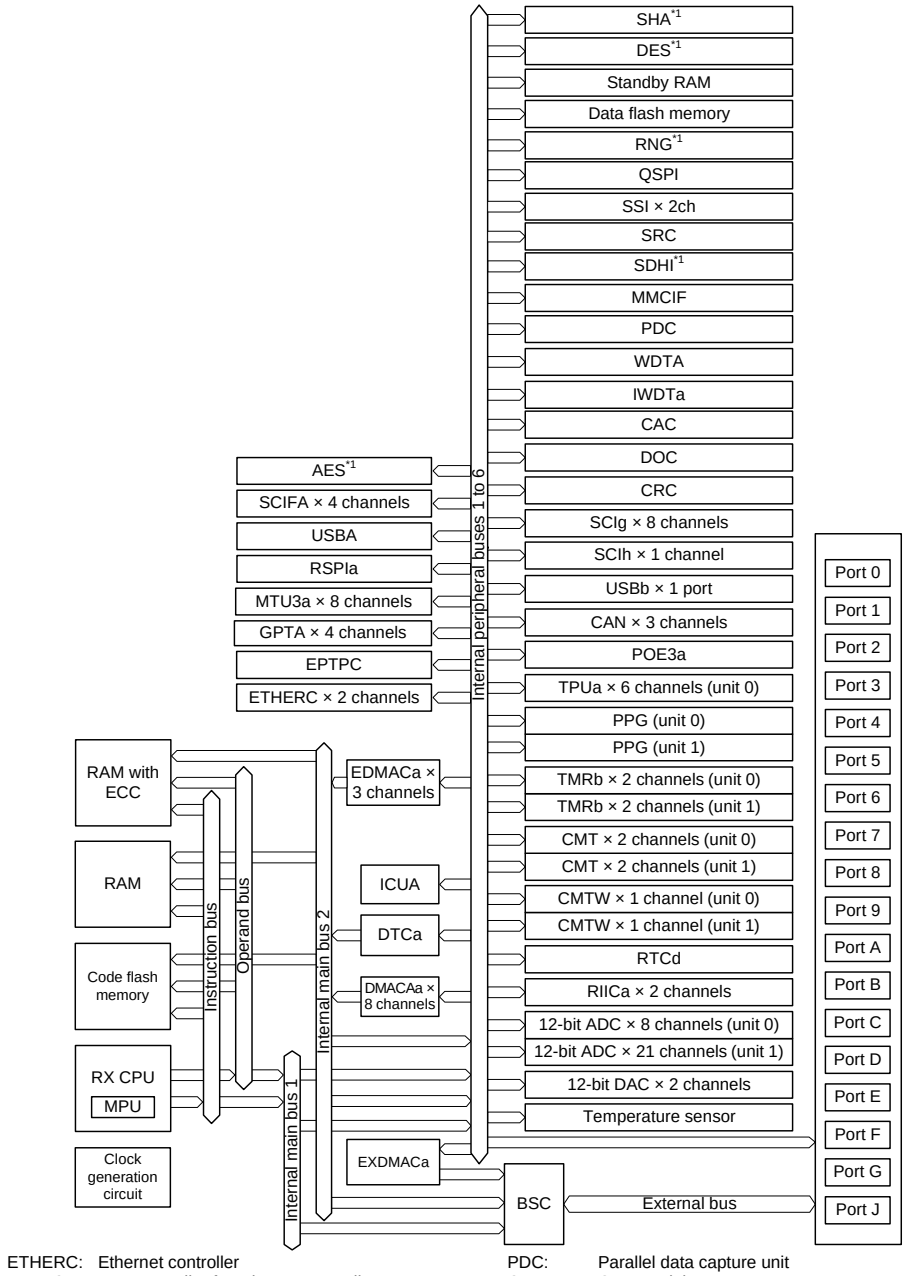


Figure 1.1 How to Read the Product Part Number

1.3 Block Diagram

Figure 1.2 shows a block diagram.



ETHERC: Ethernet controller
EPTPC: PTP controller for ethernet controller
EDMAC: DMA controller for ethernet controller
ICUA: Interrupt controller
DTCa: Data transfer controller
DMACAa: DMA controller
EXDMACa: EXDMA controller
BSC: Bus controller
WDTA: Watchdog timer
IWDTa: Independent watchdog timer
CRC: CRC (cyclic redundancy check) calculator
SCI: Serial communications interface
SCIFA: Serial communications interface with FIFO
USBb: USB2.0 FS host/function module
USBA: USB2.0 FS host/function module with battery charging
RSPIa: Serial peripheral interface
MPU: Memory protection unit
QSPI: Quad serial peripheral interface
SDHI: SD host interface*1
MMCIF: MMC host interface

PDC: Parallel data capture unit
CAN: CAN module
MTU3a: Multi-function timer pulse unit 3
POE3a: Port output enable 3
GPTA: General-purpose PWM timer
TPUa: 16-bit timer pulse unit
PPG: Programmable pulse generator
TMRb: 8-bit timer
CMT: Compare match timer
CMTW: Compare match timer W
RTCd: Realtime clock
RIICa: I^2C bus interface
DOC: Data operation circuit
CAC: Clock frequency accuracy measurement circuit
AES: AES*1
DES: DES*1
SHA: SHA-256*1
RNG: True random number generator*1
SSI: Serial sound interface
SRC: Sampling rate converter

Note 1. Optional

Figure 1.2 Block Diagram

1.4 Pin Functions

Table 1.4 lists the pin functions.

Table 1.4 Pin Functions (1/8)

Classifications	Pin Name	I/O	Description
Digital power supply	VCC	Input	Power supply pin. Connect this pin to the system power supply. Connect the pin to VSS via a 0.1-μF multilayer ceramic capacitor. The capacitor should be placed close to the pin.
	VCL	Input	Connect this pin to VSS via a 0.1-μF multilayer ceramic capacitor. The capacitor should be placed close to the pin.
	VSS	Input	Ground pin. Connect it to the system power supply (0 V).
	VBATT	Input	Backup power pin
Clock	XTAL	Output	Pins for a crystal resonator. An external clock signal can be input through the EXTAL pin.
	EXTAL	Input	
	BCLK	Output	Outputs the external bus clock for external devices.
	SDCLK	Output	Outputs the SDRAM-dedicated clock.
	XCOUT	Output	Input/output pins for the sub clock oscillator. Connect a crystal resonator between XCOUT and XCIN.
	XCIN	Input	
Clock frequency accuracy measurement	CACREF	Input	Reference clock input pin for the clock frequency accuracy measurement circuit
Operating mode control	MD	Input	Pins for setting the operating mode. The signal levels on these pins must not be changed during operation.
	UB	Input	USB boot mode or user boot mode enable pin
	UPSEL	Input	Selects the power supply method in USB boot mode. The low level selects self-power mode and the high level selects bus power mode.
System control	RES#	Input	Reset signal input pin. This LSI enters the reset state when this signal goes low.
	EMLE	Input	Input pin for the on-chip emulator enable signal. When the on-chip emulator is used, this pin should be driven high. When not used, it should be driven low.
	BSCANP	Input	Boundary scan enable pin. Boundary scan is enabled when this pin goes high. When not used, it should be driven low.
On-chip emulator	FINED	I/O	Fine interface pin
	TRST#	Input	On-chip emulator or boundary scan pins. When the EMLE pin is driven high, these pins are dedicated for the on-chip emulator.
	TMS	Input	
	TDI	Input	
	TCK	Input	
	TDO	Output	
	TRCLK	Output	This pin outputs the clock for synchronization with the trace data.
	TRSYNC	Output	This pin indicates that output from the TRDATA0 to TRDATA3 pins is valid.
	TRDATA0 to TRDATA3	Output	These pins output the trace information.
Address bus	A0 to A23	Output	Output pins for the address
Data bus	D0 to D31	I/O	Input and output pins for the bidirectional data bus
Multiplexed bus	A0/D0 to A15/D15	I/O	Address/data multiplexed bus

Table 1.4 Pin Functions (2/8)

Classifications	Pin Name	I/O	Description
Bus control	RD#	Output	Strobe signal which indicates that reading from the external bus interface space is in progress
	WR#	Output	Strobe signal which indicates that writing to the external bus interface space is in progress, in 1-write strobe mode
	WR0# to WR3#	Output	Strobe signals which indicate that either group of data bus pins (D7 to D0, D15 to D8, D23 to D16 and D31 to D24) is valid in writing to the external bus interface space, in byte strobe mode
	BC0# to BC3#	Output	Strobe signals which indicate that either group of data bus pins (D7 to D0, D15 to D8, D23 to D16 and D31 to D24) is valid in access to the external bus interface space, in 1-write strobe mode
	ALE	Output	Address latch signal when address/data multiplexed bus is selected
	WAIT#	Input	Input pin for wait request signals in access to the external space
	CS0# to CS7#	Output	Select signals for CS areas
	CKE	Output	SDRAM clock enable signal
	SDCS#	Output	SDRAM chip select signal
	RAS#	Output	SDRAM row address strobe signal
	CAS#	Output	SDRAM column address strove signal
	WE#	Output	SDRAM write enable pin
	DQM0 to DQM3	Output	SDRAM I/O data mask enable signals
EXDMA controller	EDREQ0, EDREQ1	Input	External DMA transfer request pins
	EDACK0, EDACK1	Output	Single address transfer acknowledge signals
Interrupt	NMI	Input	Non-maskable interrupt request pin
	IRQ0 to IRQ15	Input	Maskable interrupt request pins
Multi-function timer pulse unit 3	MTIOC0A, MTIOC0B MTIOC0C, MTIOC0D	I/O	The TGRA0 to TGRD0 input capture input/output compare output/PWM output pins
	MTIOC1A, MTIOC1B	I/O	The TGRA1 and TGRB1 input capture input/output compare output/PWM output pins
	MTIOC2A, MTIOC2B	I/O	The TGRA2 and TGRB2 input capture input/output compare output/PWM output pins
	MTIOC3A, MTIOC3B MTIOC3C, MTIOC3D	I/O	The TGRA3 to TGRD3 input capture input/output compare output/PWM output pins
	MTIOC4A, MTIOC4B MTIOC4C, MTIOC4D	I/O	The TGRA4 to TGRD4 input capture input/output compare output/PWM output pins
	MTIC5U, MTIC5V MTIC5W	Input	The TGRU5, TGRV5, and TGRW5 input capture input/dead time compensation input pins
	MTIOC6A, MTIOC6B MTIOC6C, MTIOC6D	I/O	The TGRA6 to TGRD6 input capture input/output compare output/PWM output pins
	MTIOC7A, MTIOC7B MTIOC7C, MTIOC7D	I/O	The TGRA7 to TGRD7 input capture input/output compare output/PWM output pins
	MTIOC8A, MTIOC8B MTIOC8C, MTIOC8D	I/O	The TGRA8 to TGRD8 input capture input/output compare output/PWM output pins
	MTCLKA, MTCLKB MTCLKC, MTCLKD	Input	Input pins for external clock signals or for phase counting mode clock signals
Port output enable 3	POE0#, POE4#, POE8#, POE10#, POE11#	Input	Input pins for request signals to place the MTU or GPT in the high impedance state

Table 1.4 Pin Functions (3/8)

Classifications	Pin Name	I/O	Description
General-purpose PWM timer	GTIOC0A-A/GTIOC0A-B/ GTIOC0A-C/GTIOC0A-D/ GTIOC0A-E, GTIOC0B-A/GTIOC0B-B/ GTIOC0B-C/GTIOC0B-D/ GTIOC0B-E	I/O	GPT0.GTGRA and GPT0.GTGRB input capture input/output compare output/PWM output pins
	GTIOC1A-A/GTIOC1A-B/ GTIOC1A-C/GTIOC1A-D/ GTIOC1A-E, GTIOC1B-A/GTIOC1B-B/ GTIOC1B-C/GTIOC1B-D/ GTIOC1B-E	I/O	GPT1.GTGRA and GPT1.GTGRB input capture input/output compare output/PWM output pins
	GTIOC2A-A/GTIOC2A-B/ GTIOC2A-C/GTIOC2A-D/ GTIOC2A-E, GTIOC2B-A/GTIOC2B-B/ GTIOC2B-C/GTIOC2B-D/ GTIOC2B-E	I/O	GPT2.GTGRA and GPT2.GTGRB input capture input/output compare output/PWM output pins
	GTIOC3A-D/GTIOC3A-E, GTIOC3B-D/GTIOC3B-E	I/O	GPT3.GTGRA and GPT3.GTGRB input capture input/output compare output/PWM output pins
	GTETRG-B/GTETRG-C/ GTETRG-D	Input	External trigger input pin for GPT0 to GPT3
16-bit timer pulse unit	TIOCA0, TIOCB0 TIOCC0, TIOCD0	I/O	The TGRA0 to TGRD0 input capture input/output compare output/PWM output pins
	TIOCA1, TIOCB1	I/O	The TGRA1 and TGRB1 input capture input/output compare output/PWM output pins
	TIOCA2, TIOCB2	I/O	The TGRA2 and TGRB2 input capture input/output compare output/PWM output pins
	TIOCA3, TIOCB3 TIOCC3, TIOCD3	I/O	The TGRA3 to TGRD3 input capture input/output compare output/PWM output pins
	TIOCA4, TIOCB4	I/O	The TGRA4 and TGRB4 input capture input/output compare output/PWM output pins
	TIOCA5, TIOCB5	I/O	The TGRA5 and TGRB5 input capture input/output compare output/PWM output pins
	TCLKA, TCLKB TCLKC, TCLKD	Input	Input pins for external clock signals or for phase counting mode clock signals
Programmable pulse generator	PO0 to PO31	Output	Output pins for the pulse signals
8-bit timer	TMO0 to TMO3	Output	Compare match output pins
	TMCI0 to TMCI3	Input	Input pins for external clocks to be input to the counter
	TMRI0 to TMRI3	Input	Input pins for the counter reset
Compare match timer W	TIC0 to TIC3	Input	Input pins for CMTW
	TOC0 to TOC3	Output	Output pins for CMTW

Table 1.4 Pin Functions (4/8)

Classifications	Pin Name	I/O	Description
Serial communications interface (SCIg)	• Asynchronous mode/clock synchronous mode		
	SCK0 to SCK7	I/O	Input/output pins for the clock
	RXD0 to RXD7	Input	Input pins for received data
	TXD0 to TXD7	Output	Output pins for transmitted data
	CTS0# to CTS7#	Input	Input pins for controlling the start of transmission and reception
	RTS0# to RTS7#	Output	Output pins for controlling the start of transmission and reception
	• Simple I^2C mode		
	SSCL0 to SSCL7	I/O	Input/output pins for the I^2C clock
	SSDA0 to SSDA7	I/O	Input/output pins for the I^2C data
	• Simple SPI mode		
	SCK0 to SCK7	I/O	Input/output pins for the clock
	SMISO0 to SMISO7	I/O	Input/output pins for slave transmission of data
	SMOSI0 to SMOSI7	I/O	Input/output pins for master transmission of data
	SS0# to SS7#	Input	Chip-select input pins
Serial communications interface (SCIh)	• Asynchronous mode/clock synchronous mode		
	SCK12	I/O	Input/output pin for the clock
	RXD12	Input	Input pin for received data
	TXD12	Output	Output pin for transmitted data
	CTS12#	Input	Input pin for controlling the start of transmission and reception
	RTS12#	Output	Output pin for controlling the start of transmission and reception
	• Simple I^2C mode		
	SSCL12	I/O	Input/output pin for the I^2C clock
	SSDA12	I/O	Input/output pin for the I^2C data
	• Simple SPI mode		
	SCK12	I/O	Input/output pin for the clock
	SMISO12	I/O	Input/output pin for slave transmission of data
	SMOSI12	I/O	Input/output pin for master transmission of data
	SS12#	Input	Chip-select input pin
	• Extended serial mode		
	RXDX12	Input	Input pin for received data
	TXDX12	Output	Output pin for transmitted data
	SIOX12	I/O	Input/output pin for received or transmitted data
Serial communications interface with FIFO (SCIFA)	SCK8 to SCK11	I/O	Input/output pins for the clock
	RXD8 to RXD11	Input	Input pins for received data
	TXD8 to TXD11	Output	Output pins for transmitted data
	CTS8# to CTS11#	Input	Input pins for controlling the start of transmission and reception
	RTS8# to RTS11#	Output	Output pins for controlling the start of transmission and reception
I^2C bus interface	SCL0[FM+], SCL2	I/O	Input/output pins for clocks. Bus can be directly driven by the N-channel open drain
	SDA0[FM+], SDA2	I/O	Input/output pins for data. Bus can be directly driven by the N-channel open drain

Table 1.4 Pin Functions (5/8)

Classifications	Pin Name	I/O	Description
Ethernet controller	REF50CK0, REF50CK1	Input	50-MHz reference clocks. These pins input reference signals for transmission/reception timings in RMII mode.
	RMII0_CRS_DV, RMII1_CRS_DV	Input	Indicate that there are carrier detection signals and valid receive data on RMII_RXD1 and RMII_RXD0 in RMII mode.
	RMII0_TXD0, RMII0_TXD1, RMII1_TXD0, RMII1_TXD1	Output	2-bit transmit data in RMII mode
	RMII0_RXD0, RMII0_RXD1, RMII1_RXD0, RMII1_RXD1	Input	2-bit receive data in RMII mode
	RMII0_TXD_EN, RMII1_TXD_EN	Output	Output pins for data transmit enable signals in RMII mode
	RMII0_RX_ER, RMII1_RX_ER	Input	Indicate an error has occurred during reception of data in RMII mode.
	ET0_CRS, ET1_CRS	Input	Carrier detection/data reception enable pins
	ET0_RX_DV, ET1_RX_DV	Input	Indicate that there are valid receive data on ET_ERXD3 to ET_ERXD0.
	ET0_EXOUT, ET1_EXOUT	Output	General-purpose external output pins
	ET0_LINKSTA ET1_LINKSTA	Input	Input link status from the PHY-LSI.
	ET0_ETXD0 to ET0_ETXD3, ET1_ETXD0 to ET1_ETXD3	Output	4 bits of MII transmit data
	ET0_ERXD0 to ET0_ERXD3, ET1_ERXD0 to ET1_ERXD3	Input	4 bits of MII receive data
	ET0_TX_EN, ET1_TX_EN	Output	Transmit enable pins. Function as signals indicating that transmit data is ready on ET_ETXD3 to ET_ETXD0.
	ET0_TX_ER, ET1_TX_ER	Output	Transmit error pins. Function as signals notifying the PHY-LSI of an error during transmission.
	ET0_RX_ER, ET1_RX_ER	Input	Receive error pins. Function as signals to recognize an error during reception.
	ET0_TX_CLK, ET1_TX_CLK	Input	Transmit clock pins. These pins input reference signals for output timings from ET_TX_EN, ET_ETXD3 to ET_ETXD0, and ET_TX_ER.
	ET0_RX_CLK, ET1_RX_CLK	Input	Receive clock pins. These pins input reference signals for input timings to ET_RX_DV, ET_ERXD3 to ET_ERXD0, and ET_RX_ER.
	ET0_COL, ET1_COL	Input	Input collision detection signals.
	ET0_WOL, ET1_WOL	Output	Receive Magic packets.
	ET0_MDC, ET1_MDC	Output	Output reference clock signals for information transfer via ET_MDIO.
	ET0_MDIO, ET1_MDIO	I/O	Input or output bidirectional signals for exchange of management information between this MCU and the PHY-LSI.

Table 1.4 Pin Functions (6/8)

Classifications	Pin Name	I/O	Description
USB 2.0 host/function module	VCC_USB, VCC_USBA	Input	Power supply pins
	VSS_USB, VSS1_USBA, VSS2_USBA	Input	Ground pins
	AVCC_USBA	Input	USBA analog power supply pin
	AVSS_USBA	Input	USBA analog ground pin. Short this pin with the PVSS_USBA pin.
	PVSS_USBA	Input	USBA PLL circuit ground pin. Short this pin with the AVSS_USBA pin.
	USBA_RREF	I/O	USBA reference current supply pin. Connect 2.2 kΩ (±1%) to the AVSS_USBA pin.
	USB0_DP, USBA_DP	I/O	Input or output USB transceiver D+ data.
	USB0_DM, USBA_DM	I/O	Input or output USB transceiver D- data.
	USB0_EXICEN, USBA_EXICEN	Output	Connect to the OTG power IC.
	USB0_ID, USBA_ID	Input	Connect to the OTG power IC.
	USB0_VBUSEN USBA_VBUSEN	Output	USB VBUS power enable pins
	USB0_OVRCURA/ USB0_OVRCURB, USBA_OVRCURA/ USBA_OVRCURB	Input	USB overcurrent pins
	USB0_VBUS, USBA_VBUS	Input	USB cable connection/disconnection detection input pins
CAN module	CRX0, CRX1-DS, CRX2	Input	Input pins
	CTX0 to CTX2	Output	Output pins
Serial peripheral interface	RSPCKA-A/RSPCKA-B	I/O	Clock input/output pin
	MOSIA-A/MOSIA-B	I/O	Inputs or outputs data output from the master
	MISOA-A/MISOA-B	I/O	Inputs or outputs data output from the slave
	SSLA0-A/SSLA0-B	I/O	Input or output pin for slave selection
	SSLA1-A/SSLA1-B to SSLA3-A/SSLA3-B	Output	Output pin for slave selection
Quad serial peripheral interface	QSPCLK-A/-B	Output	QSPI clock output pin
	QSSL-A/-B	Output	QSPI slave output pin
	QMO-A/-B, QIO0-A/-B	I/O	Master transmit data/data 0
	QMI-A/-B, QIO1-A/-B	I/O	Master input data/data 1
	QIO2-A/-B, QIO3-A/-B	I/O	Data 2, data 3
Serial sound interface	SSISCK0, SSISCK1	I/O	SSI serial bit clock pins
	SSIWS0, SSIWS1	I/O	Word select pins
	SSITXD0, SSITXD1	Output	Serial data output pins
	SSIRXD0, SSIRXD1	Input	Serial data input pins
	SSIDATA0, SSIDATA1	I/O	Serial data input/output pins
	AUDIO_MCLK	Input	Master clock pin for audio

Table 1.4 Pin Functions (7/8)

Classifications	Pin Name	I/O	Description
MMC host interface	MMC_CLK-A/ MMC_CLK-B	Output	MMC clock pin
	MMC_CMD-A/ MMC_CMD-B	I/O	Command/response pin
	MMC_D7-A/MMC_D7-B to MMC_D0-A/MMC_D0-B	I/O	Transmit data/receive data
	MMC_CD-A/MMC_CD-B	Input	Card detection pin
	MMC_RES#-A/MMC_RES#-B	Output	MMC reset output pin
SD host interface	SDHI_CLK-A/SDHI_CLK-B	Output	SD clock output pin
	SDHI_CMD-A/SDHI_CMD-B	I/O	SD command output, response input signal pin
	SDHI_D3-A/SDHI_D3-B to SDHI_D0-A/SDHI_D0-B	I/O	SD data bus pins
	SDHI_CD-A/SDHI_CD-B	Input	SD card detection pin
	SDHI_WP-A/SDHI_WP-B	Input	SD write-protect signal
Parallel data capture unit	PIXCLK	Input	Image transfer clock pin
	VSYNC	Input	Vertical synchronization signal pin
	HSYNC	Input	Horizontal synchronization signal pin
	PIXD0 to PIXD7	Input	8-bit image data pins
	PCKO	Output	Output pin for dot clock
Realtime clock	RTCOUT	Output	Output pin for 1-Hz/64-Hz clock
	RTCIC0 to RTCIC2	Input	Time capture event input pins
12-bit A/D converter	AN000 to AN007, AN100 to AN120	Input	Input pins for the analog signals to be processed by the A/D converter
	ADTRG0#, ADTRG1#	Input	Input pins for the external trigger signals that start the A/D conversion
	ANEX0	Output	Extended analog output pin
	ANEX1	Input	Extended analog input pin
12-bit D/A converter	DA0, DA1	Output	Output pins for the analog signals to be processed by the D/A converter
Analog power supply	AVCC0	Input	Analog voltage supply pin for the 12-bit A/D converter (unit 0). Connect this pin to a branch from the VCC power supply.
	AVSS0	Input	Analog ground pin for the 12-bit A/D converter (unit 0). Connect this pin to a branch from the VSS ground power supply.
	VREFH0	Input	Analog reference voltage supply pin for the 12-bit A/D converter (unit 0). Connect this pin to VCC if the 12-bit A/D converter is not to be used.
	VREFL0	Input	Analog reference ground pin for the 12-bit A/D converter (unit 0). Connect this pin to VSS if the 12-bit A/D converter is not to be used.
	AVCC1	Input	Analog voltage supply and reference voltage supply pin for the 12-bit A/D converter (unit 1) and D/A converter. This pin also supplies the analog voltage to the temperature sensor. Connect this pin to a branch from the VCC power supply.
	AVSS1	Input	Analog voltage supply and reference voltage supply pin for the 12-bit A/D converter (unit 1) and D/A converter. This pin also supplies the analog ground voltage to the temperature sensor. Connect this pin to a branch from the VSS ground power supply.

Table 1.4 Pin Functions (8/8)

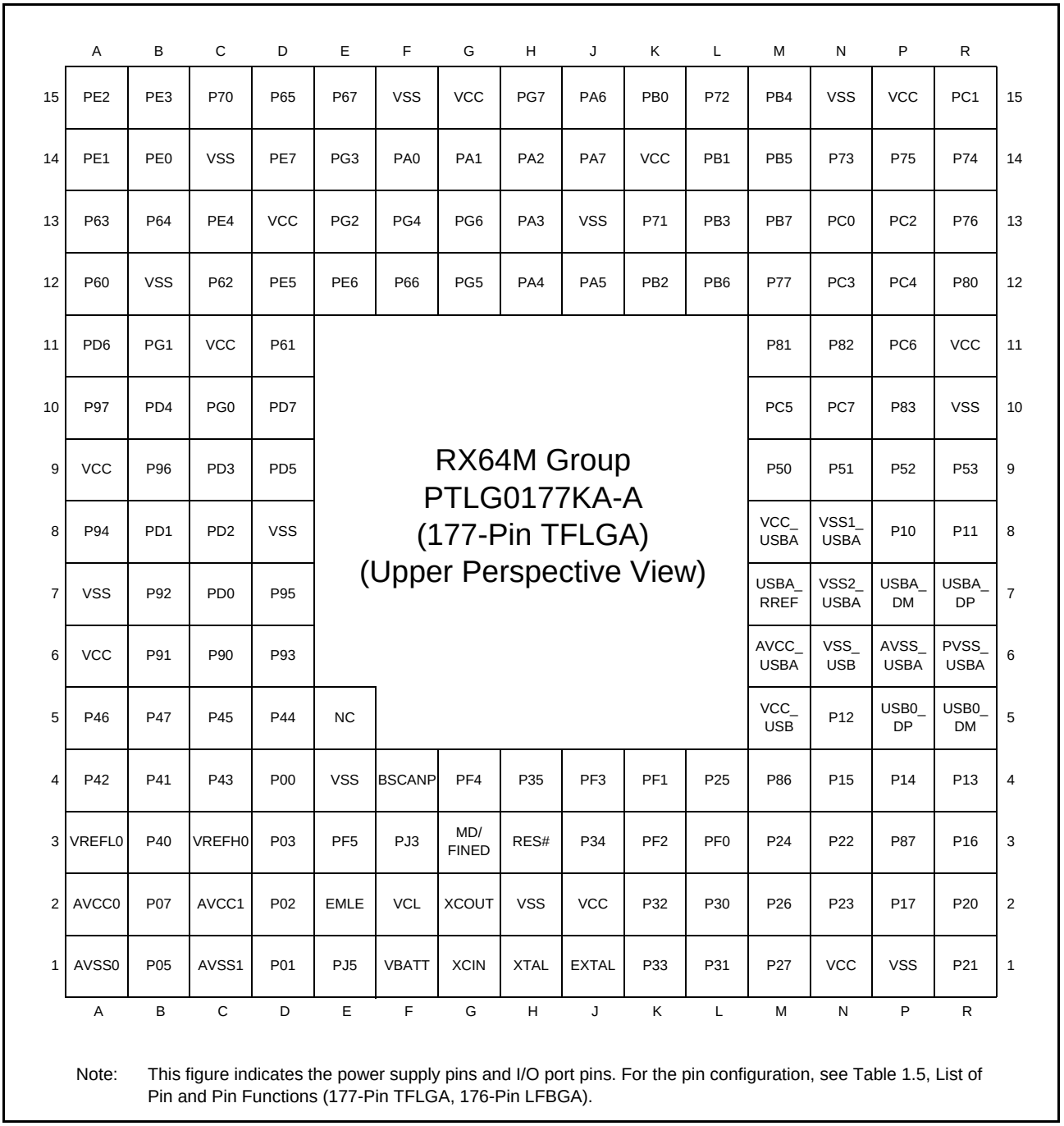
Classifications	Pin Name	I/O	Description
I/O ports	P00 to P03, P05, P07	I/O	6-bit input/output pins
	P10 to P17	I/O	8-bit input/output pins
	P20 to P27	I/O	8-bit input/output pins
	P30 to P37	I/O	8-bit input/output pins (P35: input pin)
	P40 to P47	I/O	8-bit input/output pins
	P50 to P56	I/O	7-bit input/output pins (176-pin devices have only P50 to P53)
	P60 to P67	I/O	8-bit input/output pins
	P70 to P77	I/O	8-bit input/output pins
	P80 to P83, P86, P87	I/O	6-bit input/output pins
	P90 to P97	I/O	8-bit input/output pins
	PA0 to PA7	I/O	8-bit input/output pins
	PB0 to PB7	I/O	8-bit input/output pins
	PC0 to PC7	I/O	8-bit input/output pins
	PD0 to PD7	I/O	8-bit input/output pins
	PE0 to PE7	I/O	8-bit input/output pins
	PF0 to PF5	I/O	6-bit input/output pins
	PG0 to PG7	I/O	8-bit input/output pins
	PJ3, PJ5	I/O	2-bit input/output pins

Note: Note the following regarding pin names. For details, see section 1.5, Pin Assignments.

- We recommend using pins that have a letter (“-A”, “-B”, etc.) to indicate group membership appended to their names as groups. For the RSPI, QSPI, SDHI, and MMC interfaces, the AC portion of the electrical characteristics is measured for each group.
- Pins that have "-DS" appended to their names can be used as triggers for release from deep software standby.
- RIIC pin functions that have [FM+] appended to their names support fast-mode plus.

1.5 Pin Assignments

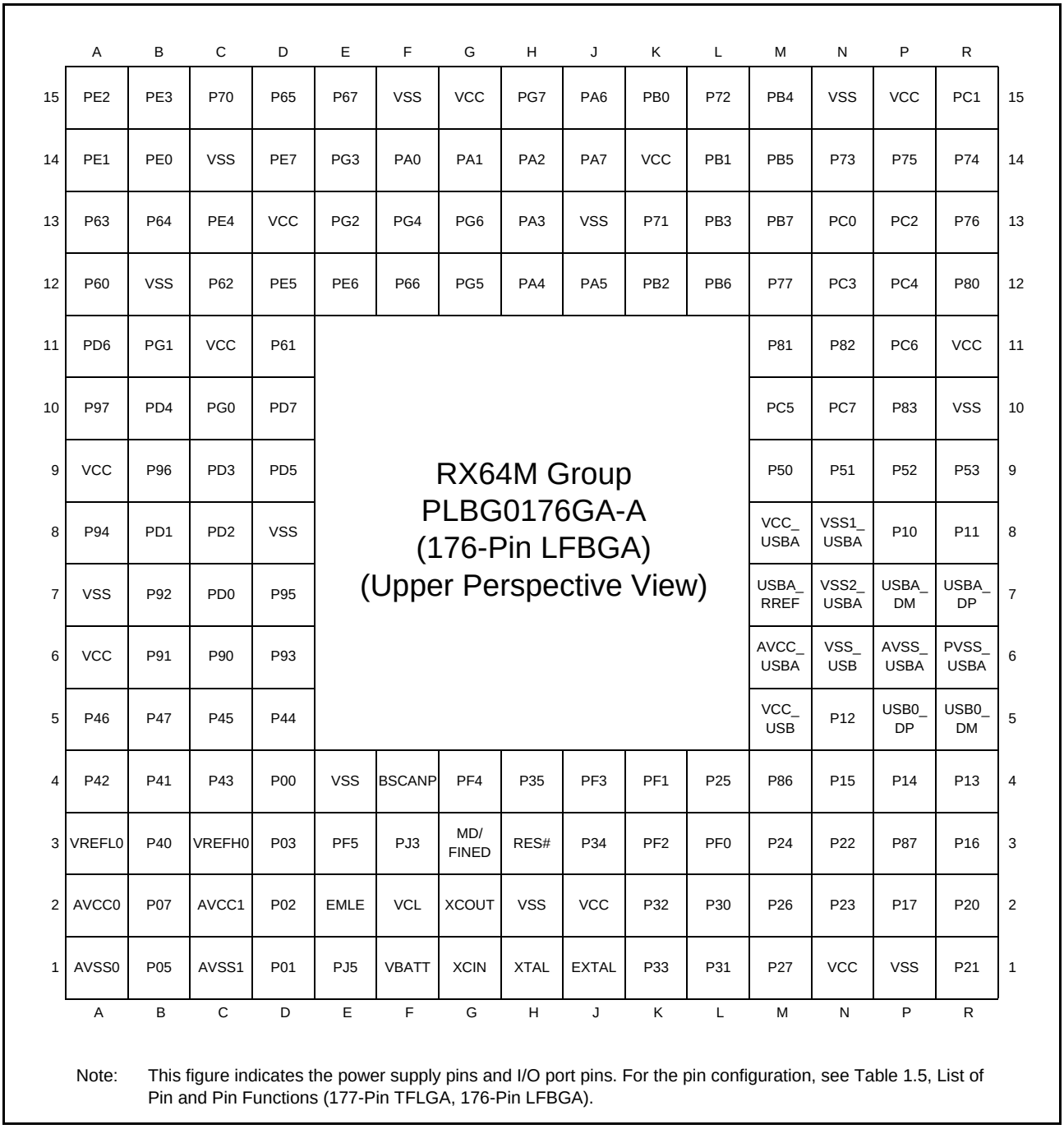
Figure 1.3 to Figure 1.9 show the pin assignments. Table 1.5 to Table 1.10 show the lists of pins and pin functions.



	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	
15	PE2	PE3	P70	P65	P67	VSS	VCC	PG7	PA6	PB0	P72	PB4	VSS	VCC	PC1	15
14	PE1	PE0	VSS	PE7	PG3	PA0	PA1	PA2	PA7	VCC	PB1	PB5	P73	P75	P74	14
13	P63	P64	PE4	VCC	PG2	PG4	PG6	PA3	VSS	P71	PB3	PB7	PC0	PC2	P76	13
12	P60	VSS	P62	PE5	PE6	P66	PG5	PA4	PA5	PB2	PB6	P77	PC3	PC4	P80	12
11	PD6	PG1	VCC	P61								P81	P82	PC6	VCC	11
10	P97	PD4	PG0	PD7								PC5	PC7	P83	VSS	10
9	VCC	P96	PD3	PD5								P50	P51	P52	P53	9
8	P94	PD1	PD2	VSS								VCC_ USBA	VSS1_ USBA	P10	P11	8
7	VSS	P92	PD0	P95								USBA_ RREF	VSS2_ USBA	USBA_ DM	USBA_ DP	7
6	VCC	P91	P90	P93								AVCC_ USBA	VSS_ USB	AVSS_ USBA	PVSS_ USBA	6
5	P46	P47	P45	P44	NC							VCC_ USB	P12	USB0_ DP	USB0_ DM	5
4	P42	P41	P43	P00	VSS	BSCANP	PF4	P35	PF3	PF1	P25	P86	P15	P14	P13	4
3	VREFL0	P40	VREFH0	P03	PF5	PJ3	MD/ FINED	RES#	P34	PF2	PF0	P24	P22	P87	P16	3
2	AVCC0	P07	AVCC1	P02	EMLE	VCL	XCOUT	VSS	VCC	P32	P30	P26	P23	P17	P20	2
1	AVSS0	P05	AVSS1	P01	PJ5	VBATT	XCIN	XTAL	EXTAL	P33	P31	P27	VCC	VSS	P21	1
	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	

Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.5, List of Pin and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA).

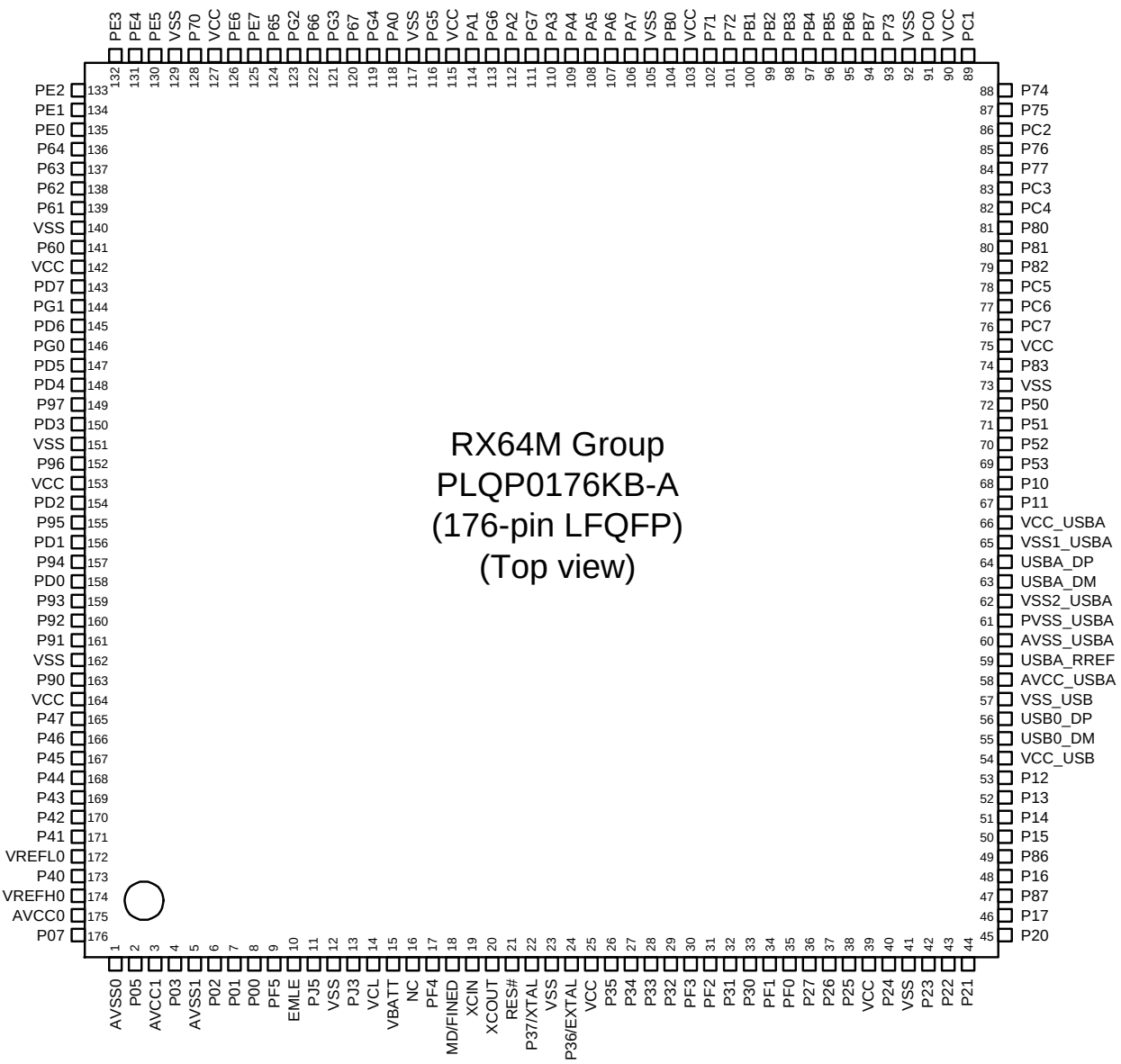
Figure 1.3 Pin Assignment (177-Pin TFLGA)



	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	
15	PE2	PE3	P70	P65	P67	VSS	VCC	PG7	PA6	PB0	P72	PB4	VSS	VCC	PC1	15
14	PE1	PE0	VSS	PE7	PG3	PA0	PA1	PA2	PA7	VCC	PB1	PB5	P73	P75	P74	14
13	P63	P64	PE4	VCC	PG2	PG4	PG6	PA3	VSS	P71	PB3	PB7	PC0	PC2	P76	13
12	P60	VSS	P62	PE5	PE6	P66	PG5	PA4	PA5	PB2	PB6	P77	PC3	PC4	P80	12
11	PD6	PG1	VCC	P61								P81	P82	PC6	VCC	11
10	P97	PD4	PG0	PD7								PC5	PC7	P83	VSS	10
9	VCC	P96	PD3	PD5								P50	P51	P52	P53	9
8	P94	PD1	PD2	VSS								VCC_ USBA	VSS1_ USBA	P10	P11	8
7	VSS	P92	PD0	P95								USBA_ RREF	VSS2_ USBA	USBA_ DM	USBA_ DP	7
6	VCC	P91	P90	P93								AVCC_ USBA	VSS_ USB	AVSS_ USBA	PVSS_ USBA	6
5	P46	P47	P45	P44								VCC_ USB	P12	USB0_ DP	USB0_ DM	5
4	P42	P41	P43	P00	VSS	BSCANP	PF4	P35	PF3	PF1	P25	P86	P15	P14	P13	4
3	VREFL0	P40	VREFH0	P03	PF5	PJ3	MD/ FINED	RES#	P34	PF2	PF0	P24	P22	P87	P16	3
2	AVCC0	P07	AVCC1	P02	EMLE	VCL	XCOUT	VSS	VCC	P32	P30	P26	P23	P17	P20	2
1	AVSS0	P05	AVSS1	P01	PJ5	VBATT	XCIN	XTAL	EXTAL	P33	P31	P27	VCC	VSS	P21	1
	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	

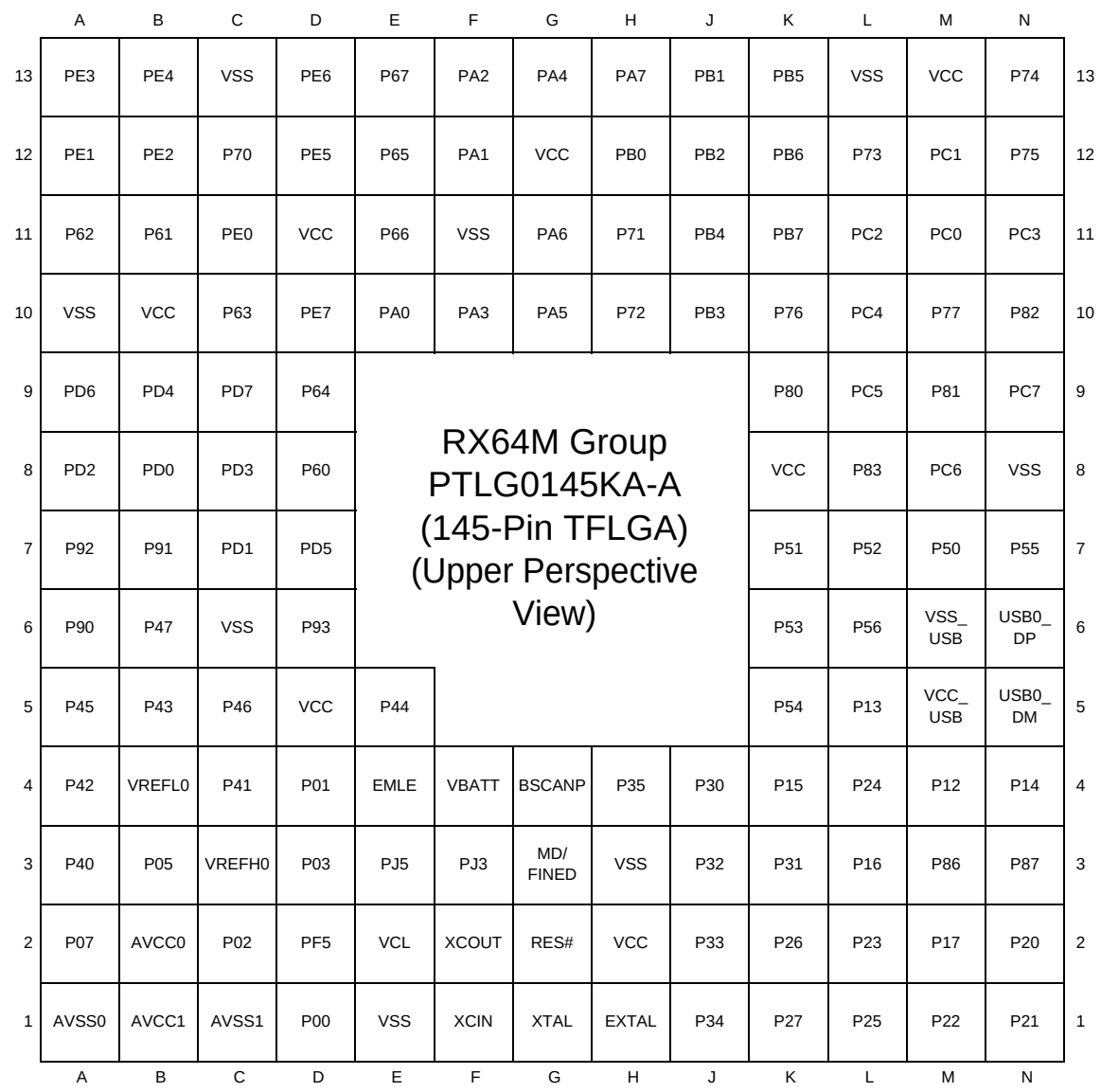
Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.5, List of Pin and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA).

Figure 1.4 Pin Assignment (176-Pin LFBGA)



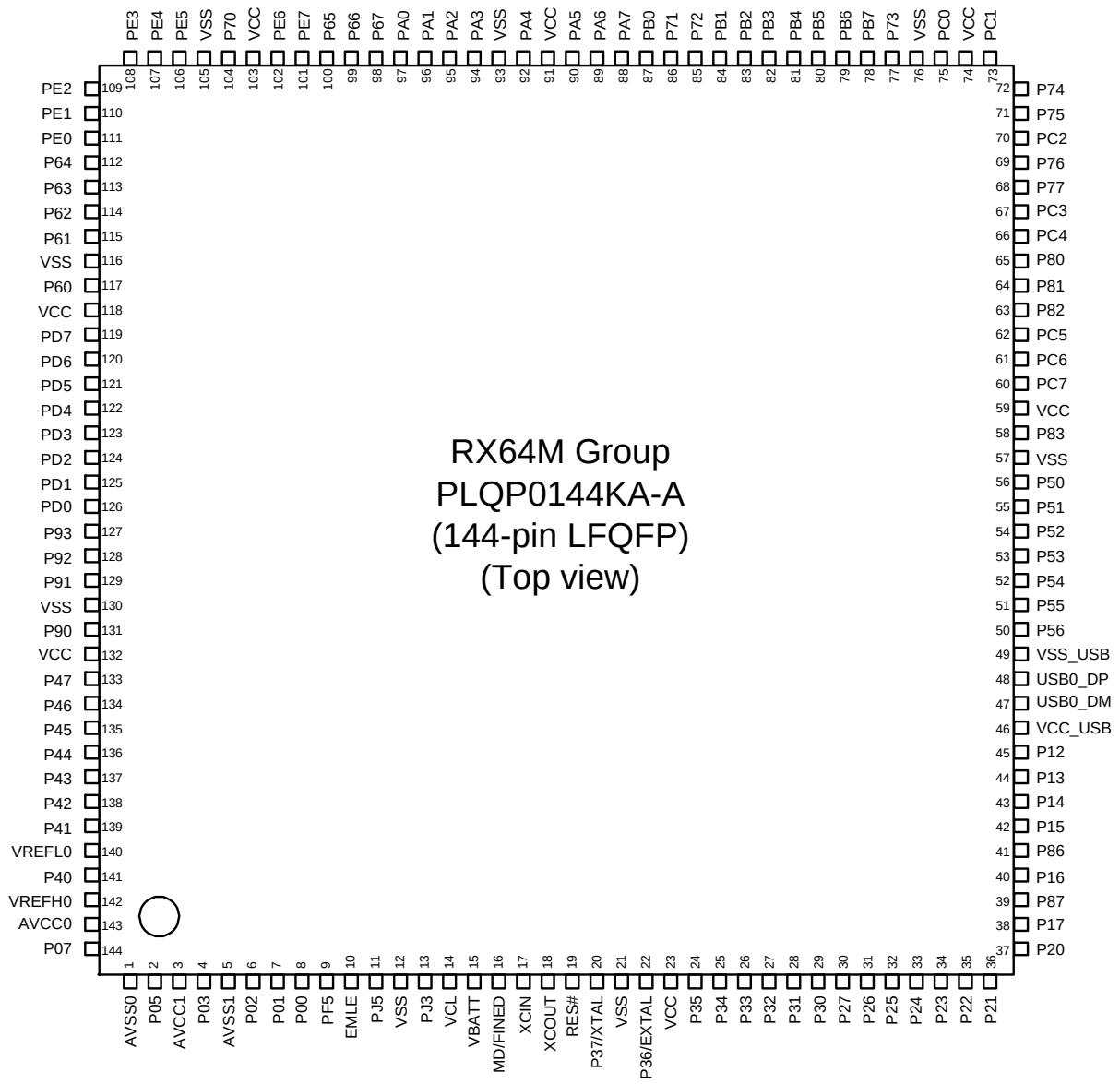
Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.6, List of Pin and Pin Functions (176-Pin LFQFP).

Figure 1.5 Pin Assignment (176-Pin LFQFP)



Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.7, List of Pin and Pin Functions (145-Pin TFLGA).

Figure 1.6 Pin Assignment (145-Pin TFLGA)



Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.8, List of Pin and Pin Functions (144-Pin LFQFP).

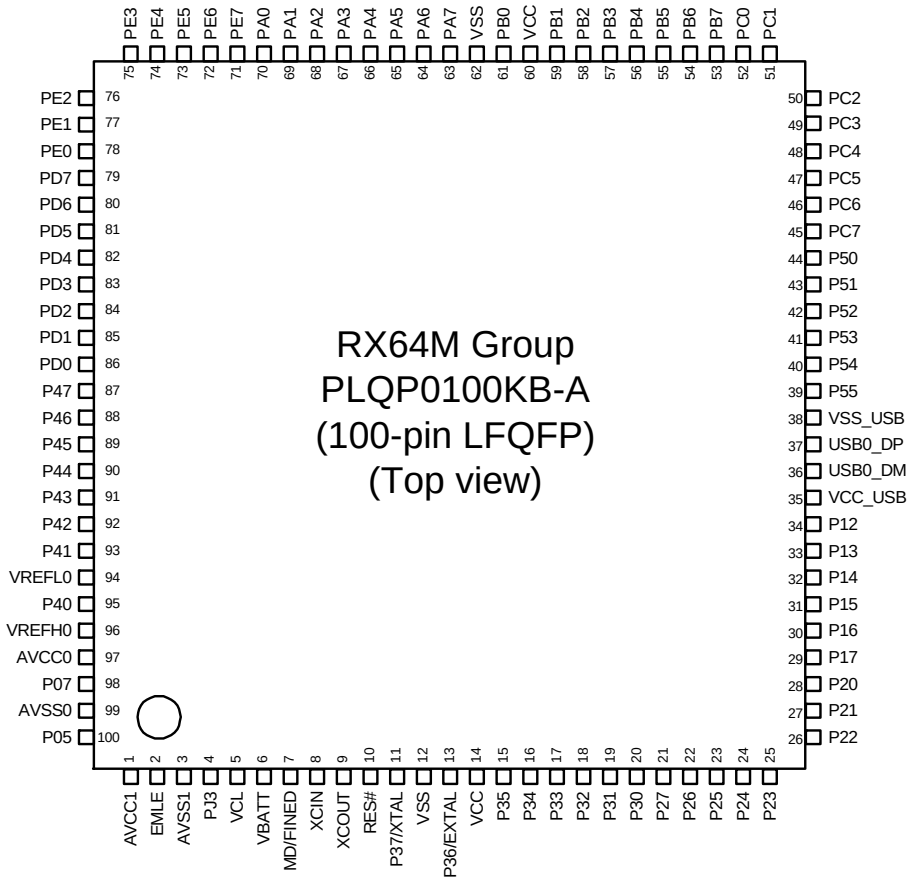
Figure 1.7 Pin Assignment (144-Pin LFQFP)

RX64M Group
PTLG0100JA-A (100-Pin TFLGA)
(Upper Perspective View)

	A	B	C	D	E	F	G	H	J	K	
10	PE2	PE3	PE4	PA0	PA3	VSS	VCC	PB7	PC1	PC2	10
9	PE1	PD7	PE5	PA1	PA5	PA7	PB1	PB6	PC0	PC3	9
8	PE0	PD6	PD5	PE7	PA4	PB0	PB4	PC6	PC4	PC5	8
7	PD4	PD3	PD2	PE6	PA6	PB2	PB5	PC7	P50	P51	7
6	PD0	PD1	P47	P46	PA2	PB3	P52	P54	VCC_ USB	USB0_ DP	6
5	P43	P44	P42	P45	P41	P12	P53	P55	VSS_ USB	USB0_ DM	5
4	VREFL0	P40	VREFH0	VBATT	P34	P32	P27	P15	P13	P14	4
3	P07	AVCC0	PJ3	MD/ FINED	RES#	P35	P30	P16	P17	P20	3
2	AVCC1	AVSS0	AVSS1	XCOUT	VSS	VCC	P31	P25	P21	P22	2
1	P05	EMLE	VCL	XCIN	XTAL	EXTAL	P33	P26	P24	P23	1
	A	B	C	D	E	F	G	H	J	K	

Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.9, List of Pin and Pin Functions (100-Pin TFLGA).

Figure 1.8 Pin Assignment (100-Pin TFLGA)



Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.10, List of Pin and Pin Functions (100-Pin LFQFP).

Figure 1.9 Pin Assignment (100-Pin LFQFP)

Table 1.5 List of Pin and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA) (1/7)

Pin Number 177-Pin TFLGA 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
A1	AVSS0							
A2	AVCC0							
A3	VREFL0							
A4		P42					IRQ10-DS	AN002
A5		P46					IRQ14-DS	AN006
A6	VCC							
A7	VSS							
A8		P94	A20/D20		ET1_ERXD0/ RMII1_RXD0			
A9	VCC							
A10		P97	A23/D23		ET1_ERXD3			
A11		PD6	D6[A6/D6]	MTIC5V/MTIOC8A/ POE4#		MMC_D0-B/ SDHI_D0-B/ QIO0-B/ QMO-B	IRQ6	AN106
A12		P60	CS0#		ET1_TX_EN/ RMII1_TXD_EN			
A13		P63	CS3#/CAS#					
A14		PE1	D9[A9/D9]	MTIOC4C/MTIOC3B/ GTIOC1B-A/PO18	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12	MMC_D5-B		ANEX1
A15		PE2	D10[A10/D10]	MTIOC4A/ GTIOC0B-A/PO23/ TIC3	RXD12/SMISO12/ SSCL12/ RXDX12	MMC_D6-B	IRQ7-DS	AN100
B1		P05					IRQ13	DA1
B2		P07					IRQ15	ADTRG0#
B3		P40					IRQ8-DS	AN000
B4		P41					IRQ9-DS	AN001
B5		P47					IRQ15-DS	AN007
B6		P91	A17/D17		ET1_COL/SCK7			AN115
B7		P92	A18/D18	POE4#	ET1_CRS/ RMII1_CRS_DV/ RXD7/SMISO7/SSCL7			AN116
B8		PD1	D1[A1/D1]	MTIOC4B/ GTIOC1A-E/POE0#	CTX0		IRQ1	AN109
B9		P96	A22/D22		ET1_ERXD2			
B10		PD4	D4[A4/D4]	MTIOC8B/POE11#		MMC_CMD-B/ SDHI_CMD-B/ QSSL-B	IRQ4	AN112
B11		PG1	D25		ET1_RX_ER/ RMII1_RX_ER			
B12	VSS							
B13		P64	CS4#/WE#					
B14		PE0	D8[A8/D8]	MTIOC3D/ GTIOC2B-A	SCK12	MMC_D4-B		ANEX0
B15		PE3	D11[A11/D11]	MTIOC4B/ GTIOC2A-A/PO26/ POE8#/TOC3	CTS12#/RTS12#/ SS12#/ ET0_ERXD3	MMC_D7-B		AN101
C1	AVSS1							
C2	AVCC1							
C3	VREFH0							

Table 1.5 List of Pin and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA) (2/7)

Pin Number 177-Pin TFLGA 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
C4		P43					IRQ11-DS	AN003
C5		P45					IRQ13-DS	AN005
C6		P90	A16/D16		ET1_RX_DV/ TXD7/SMOSI7/SSDA7			AN114
C7		PD0	D0[A0/D0]	GTIOC1B-E/POE4#			IRQ0	AN108
C8		PD2	D2[A2/D2]	MTIOC4D/ GTIOC0B-E/TIC2	CRX0	MMC_D2-B/ SDHI_D2-B/ QIO2_B	IRQ2	AN110
C9		PD3	D3[A3/D3]	MTIOC8D/ GTIOC0A-E/POE8#/ TOC2		MMC_D3-B/ SDHI_D3-B/ QIO3-B	IRQ3	AN111
C10		PG0	D24		ET1_RX_CLK/ REF50CK1			
C11	VCC							
C12		P62	CS2#/RAS#					
C13		PE4	D12[A12/D12]	MTIOC4D/MTIOC1A/ GTIOC1A-A/PO28	ET0_ERXD2			AN102
C14	VSS							
C15		P70	SDCLK					
D1		P01		TMCI0	RXD6/SMISO6/ SSCL6		IRQ9	AN119
D2		P02		TMCI1	SCK6		IRQ10	AN120
D3		P03					IRQ11	DA0
D4		P00		TMRI0	TXD6/SMOSI6/ SSDA6		IRQ8	AN118
D5		P44					IRQ12-DS	AN004
D6		P93	A19/D19	POE0#	ET1_LINKSTA/CTS7#/ RTS7#/SS7#			AN117
D7		P95	A21/D21		ET1_ERXD1/ RMII1_RXD1			
D8	VSS							
D9		PD5	D5[A5/D5]	MTIC5W/MTIOC8C/ POE10#		MMC_CLK-B/ SDHI_CLK-B/ QSPCLK-B	IRQ5	AN113
D10		PD7	D7[A7/D7]	MTIC5U/POE0#		MMC_D1-B/ SDHI_D1-B/ QIO1-B/QMI-B	IRQ7	AN107
D11		P61	CS1#/SDCS#					
D12		PE5	D13[A13/D13]	MTIOC4C/MTIOC2B/ GTIOC0A-A	ET0_RX_CLK/ REF50CK0		IRQ5	AN103
D13	VCC							
D14		PE7	D15[A15/D15]	MTIOC6A/ GTIOC3A-E/TOC1		MMC_RES#-B/ SDHI_WP-B	IRQ7	AN105
D15		P65	CS5#/CKE					
E1		PJ5		POE8#	CTS2#/RTS2#/SS2#			
E2	EMLE							
E3		PF5					IRQ4	
E4	VSS							
E5*1								
E12		PE6	D14[A14/D14]	MTIOC6C/ GTIOC3B-E/TIC1		MMC_CD-B/ SDHI_CD-B	IRQ6	AN104
E13	TRDATA0	PG2	D26		ET1_TX_CLK			

Table 1.5 List of Pin and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA) (3/7)

Pin Number 177-Pin TFLGA 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
E14	TRDATA1	PG3	D27		ET1_ETXD0/ RMII1_TXD0			
E15		P67	CS7#/DQM1	MTIOC7C/ GTIOC1B-C	CRX2		IRQ15	
F1	VBATT							
F2	VCL							
F3		PJ3	EDACK1	MTIOC3C	ET0_EXOUT/ CTS6#/RTS6#/ CTS0#/RTS0#/ SS6#/SS0#			
F4	BSCANP							
F12		P66	CS6#/DQM0	MTIOC7D/ GTIOC2B-C	CTX2			
F13	TRSYNC	PG4	D28		ET1_ETXD1/ RMII1_TXD1			
F14		PA0	A0/BC0#/ DQM2	MTIOC4A/MTIOC6D/ GTIOC0B-C/TIOCA0/ CACREF/PO16	SSLA1-B/ ET0_TX_EN/ RMII0_TXD_EN			
F15	VSS							
G1	XCIN							
G2	XCOUT							
G3	MD/FINED							
G4	TRST#	PF4						
G12	TRCLK	PG5	D29		ET1_ETXD2			
G13	TRDATA2	PG6	D30		ET1_ETXD3			
G14		PA1	A1/DQM3	MTIOC0B/MTCLKC/ MTIOC7B/ GTIOC2A-C/TIOCB0/ PO17	SCK5/SSLA2-B/ ET0_WOL		IRQ11	
G15	VCC							
H1	XTAL	P37						
H2	VSS							
H3	RES#							
H4	UPSEL	P35					NMI	
H12		PA4	A4	MTIC5U/MTCLKA/ TIOCA1/TMRI0/PO20	TXD5/SMOSI5/ SSDA5/SSLA0-B/ ET0_MDC		IRQ5-DS	
H13		PA3	A3	MTIOC0D/MTCLKD/ TIOCD0/TCLKB/PO19	RXD5/SMISO5/ SSCL5/ ET0_MDIO		IRQ6-DS	
H14		PA2	A2	MTIOC7A/ GTIOC1A-C/PO18	RXD5/SMISO5/ SSCL5/SSLA3-B			
H15	TRDATA3	PG7	D31		ET1_TX_ER			
J1	EXTAL	P36						
J2	VCC							
J3		P34		MTIOC0A/TMCI3/ PO12/POE10#	SCK6/SCK0/ ET0_LINKSTA		IRQ4	
J4	TMS	PF3						
J12		PA5	A5	MTIOC6B/ GTIOC0A-C/TIOCB1/ PO21	RSPCKA-B/ ET0_LINKSTA			
J13	VSS							
J14		PA7	A7	TIOCB2/PO23	MISOA-B/ ET0_WOL			

Table 1.5 List of Pin and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA) (4/7)

Pin Number 177-Pin TFLGA 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
J15		PA6	A6	MTIC5V/MTCLKB/ GTETRG-C/TIOCA2/ TMCI3/PO22/POE10#	CTS5#/RTS5#/SS5#/ MOSIA-B/ ET0_EXOUT			
K1		P33	EDREQ1	MTIOC0D/TIOCD0/ TMRI3/PO11/POE4#/ POE11#	RXD6/RXD0/ SMISO6/ SMISO0/SSCL6/ SSCL0/CRX0	PCKO	IRQ3-DS	
K2		P32		MTIOC0C/TIOCC0/ TMO3/PO10/ RTCOUT/RTCIC2/ POE0#/POE10#	TXD6/TXD0/ SMOSI6/SMOSI0/ SSDA6/SSDA0/ CTX0/ USB0_VBUSEN	VSYNC	IRQ2-DS	
K3	TDI	PF2			RXD1/SMISO1/ SSCL1			
K4	TCK	PF1			SCK1			
K12		PB2	A10	TIOCC3/TCLKC/ PO26	CTS4#/RTS4#/CTS6#/ RTS6#/SS4#/SS6#/ ET0_RX_CLK/ REF50CK0			
K13		P71	A18/CS1#		ET0_MDIO			
K14	VCC							
K15		PB0	A8	MTIC5W/TIOCA3/ PO24	RXD4/RXD6/SMISO4/ SMISO6/SSCL4/ SSCL6/ET0_ERXD1/ RMII0_RXD1		IRQ12	
L1		P31		MTIOC4D/TMCI2/ PO9/RTCIC1	CTS1#/RTS1#/ SS1#/ET1_MDC		IRQ1-DS	
L2		P30		MTIOC4B/TMRI3/ PO8/RTCIC0/POE8#	RXD1/SMISO1/ SSCL1/ ET1_MDIO		IRQ0-DS	
L3	TDO	PF0			TXD1/SMOSI1/ SSDA1			
L4		P25	CS5#/ EDACK1	MTIOC4C/MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/ SSCL3/ SSIDATA1	HSYNC		ADTRG0#
L12		PB6	A14	MTIOC3D/TIOCA5/ PO30	RXD9/ET0_ETXD1/ RMII0_TXD1			
L13		PB3	A11	MTIOC0A/MTIOC4A/ TIOCD3/TCLKD/ TMO0/PO27/POE11#	SCK4/SCK6/ ET0_RX_ER/ RMII0_RX_ER			
L14		PB1	A9	MTIOC0C/MTIOC4C/ TIOCB3/TMCI0/PO25	TXD4/TXD6/SMOSI4/ SMOSI6/SSDA4/ SSDA6/ET0_ERXD0/ RMII0_RXD0		IRQ4-DS	
L15		P72	A19/CS2#		ET0_MDC			
M1		P27	CS7#	MTIOC2B/TMCI3/PO7	SCK1/ET1_WOL			
M2		P26	CS6#	MTIOC2A/TMO1/PO6	TXD1/CTS3#/ RTS3#/SMOSI1/ SS3#/SSDA1/ ET1_EXOUT			
M3		P24	CS4#/ EDREQ1	MTIOC4A/MTCLKA/ TIOCB4/TMRI1/PO4	SCK3/ USB0_VBUSEN/ SSISCK1	PIXCLK		
M4		P86		MTIOC4D/ GTIOC2B-B/TIOCA0	RXD10	PIXD1		
M5	VCC_USB	P12	WR3#/BC3#	MTIC5U/TMCI1	RXD2/SMISO2/ SSCL2/ SCL0[FM+]		IRQ2	
M6	AVCC_ USBA							

Table 1.5 List of Pin and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA) (5/7)

Pin Number 177-Pin TFLGA 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
M7	USBA_ RREF	P11		MTIC5V/TMCI3	SCK2/USBA_VBUS/ USBA_VBUSEN		IRQ1	
M8	VCC_ USBA	P10	ALE	MTIC5W/TMRI3	USBA_OVRCURA		IRQ0	
M9		P50	WR0#/WR#		TXD2/SMOSI2/SSDA2			
M10		PC5	A21/CS2#/ WAIT#	MTIOC3B/MTCLKD/ GTIOC1A-D/TMRI2/ PO29	SCK8/RSPCKA-A/ RTS8#/ET0_ETXD2	MMC_D5-A		
M11		P81	EDACK0	MTIOC3D/ GTIOC0B-D/PO27	RXD10/ET0_ETXD0/ RMII0_TXD0	MMC_D3-A/ SDHI_CD-A/ QIO3-A		
M12		P77	CS7#	PO23	TXD11/ET0_RX_ER/ RMII0_RX_ER	MMC_CLK-A/ SDHI_CLK-A/ QSPCLK-A		
M13		PB7	A15	MTIOC3B/TIOCB5/ PO31	TXD9/ET0_CRS/ RMII0_CRS_DV			
M14		PB5	A13	MTIOC2A/MTIOC1B/ TIOCB4/TMRI1/PO29/ POE4#	SCK9/RTS9#/ ET0_ETXD0/ RMII0_TXD0			
M15		PB4	A12	TIOCA4/PO28	CTS9#/ET0_TX_EN/ RMII0_TXD_EN			
N1	VCC							
N2		P23	EDACK0	MTIOC3D/MTCLKD/ GTIOC0A-B/TIOCD3/ PO3	TXD3/CTS0#/ RTS0#/SMOSI3/ SS0#/SSDA3/ SSISCK0	PIXD7		
N3		P22	EDREQ0	MTIOC3B/MTCLKC/ GTIOC1A-B/TIOCC3/ TMO0/PO2	SCK0/ USB0_OVRCURB/ USBA_OVRCURB/ AUDIO_MCLK	PIXD6		
N4		P15		MTIOC0B/MTCLKB/ GTETRG-B/TIOCB2/ TCLKB/TMCI2/PO13	RXD1/SCK3/ SMISO1/SSCL1/ CRX1-DS/ USBA_VBUSEN/ SSIWS1	PIXD0	IRQ5	
N5		P12	WR3#/BC3#	MTIC5U/TMCI1	RXD2/SMISO2/ SSCL2/ SCL0[FM+]		IRQ2	
N6	VSS_USB							
N7	VSS2_ USBA							
N8	VSS1_ USBA							
N9		P51	WR1#/BC1#/ WAIT#		SCK2			
N10	UB	PC7	A23/CS0#	MTIOC3A/MTCLKB/ GTIOC3A-D/TMO2/ TOC0/PO31/CACREF	TXD8/MISOA-A/ ET0_COL	MMC_D7-A	IRQ14	
N11		P82	EDREQ1	MTIOC4A/ GTIOC2A-D/PO28	TXD10/ET0_ETXD1/ RMII0_TXD1	MMC_D4-A		
N12		PC3	A19	MTIOC4D/ GTIOC1B-D/TCLKB/ PO24	TXD5/SMOSI5/ SSDA5/ ET0_TX_ER	MMC_D0-A/ SDHI_D0-A/ QIO0-A/ QMO-A		
N13		PC0	A16	MTIOC3C/TCLKC/ PO17	CTS5#/RTS5#/SS5#/ SSLA1-A/ET0_ERXD3		IRQ14	
N14		P73	CS3#	PO16	ET0_WOL			
N15	VSS							
P1	VSS							

Table 1.5 List of Pin and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA) (6/7)

Pin Number 177-Pin TFLGA 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
P2		P17		MTIOC3A/MTIOC3B/ MTIOC4B/ GTIOC0B-B/TIOCB0/ TCLKD/TMO1/PO15/ POE8#	SCK1/TXD3/ SMOSI3/SSDA3/ SDA2-DS/ SSITXD0	PIXD3	IRQ7	ADTRG1#
P3		P87		MTIOC4C/ GTIOC1B-B/TIOCA2	TXD10	PIXD2		
P4		P14		MTIOC3A/MTCLKA/ TIOCB5/TCLKA/ TMRI2/PO15	CTS1#/RTS1#/ SS1#/CTX1/ USB0_OVRCURA		IRQ4	
P5					USB0_DP			
P6	AVSS_ USBA							
P7					USBA_DM			
P8		P10	ALE	MTIC5W/TMRI3	USBA_OVRCURA		IRQ0	
P9		P52	RD#		RXD2/SMISO2/ SSCL2			
P10		P83	EDACK1	MTIOC4C/ GTIOC0A-D	CTS10#/ET0_CRS/ RMII0_CRS_DV/ SCK10			
P11		PC6	A22/CS1#	MTIOC3C/MTCLKA/ GTIOC3B-D/TMCI2/ TIC0/PO30	RXD8/MOSIA-A/ ET0_ETXD3	MMC_D6-A	IRQ13	
P12		PC4	A20/CS3#	MTIOC3D/MTCLKC/ GTETRG-D/TMCI1/ PO25/POE0#	SCK5/CTS8#/SSLA0-A/ET0_TX_CLK	MMC_D1-A/ SDHI_D1-A/ QIO1-A/QMI-A		
P13		PC2	A18	MTIOC4B/ GTIOC2B-D/TCLKA/ PO21	RXD5/SMISO5/ SSCL5/SSLA3-A/ ET0_RX_DV/	MMC_CD-A/ SDHI_D3-A		
P14		P75	CS5#	PO20	SCK11/RTS11/ ET0_ERXD0/ RMII0_RXD0/	MMC_RES#-A/ SDHI_D2-A		
P15	VCC							
R1		P21		MTIOC1B/MTIOC4A/ GTIOC2A-B/TIOCA3/ TMCI0/PO1	RXD0/SMISO0/ SSCL0/ USB0_EXICEN/ USBA_EXICEN/ SSIWS0	PIXD5	IRQ9	
R2		P20		MTIOC1A/TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/ SSDA0/USB0_ID/ USBA_ID/ SSIRXD0	PIXD4	IRQ8	
R3		P16		MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/ TMO2/PO14/ RTCOUT	TXD1/RXD3/ SMOSI1/SMISO3/ SSDA1/SSCL3/ SCL2-DS/ USB0_VBUS/ USB0_VBUSEN/ USB0_OVRCURB		IRQ6	ADTRG0#
R4		P13	WR2#/BC2#	MTIOC0B/TIOCA5/ TMO3/PO13	TXD2/SMOSI2/ SSDA2/ SDA0[FM+]		IRQ3	ADTRG1#
R5					USB0_DM			
R6	PVSS_ USBA							
R7					USBA_DP			
R8		P11		MTIC5V/TMCI3	SCK2/ USBA_VBUS/ USBA_VBUSEN			
R9		P53*2	BCLK					
R10	VSS							

Table 1.5 List of Pin and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA) (7/7)

Pin Number 177-Pin TFLGA 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
R11	VCC							
R12		P80	EDREQ0	MTIOC3B/PO26	SCK10/RTS10#/ ET0_TX_EN/ RMII0_TXD_EN	MMC_D2-A/ SDHI_WP-A/ QIO2-A		
R13		P76	CS6#	PO22	RXD11/ET0_RX_CLK/ REF50CK0	MMC_CMD-A/ SDHI_CMD-A/ QSSL-A		
R14		P74	A20/CS4#	PO19	CTS11#/ET0_ERXD1/ RMII0_RXD1			
R15		PC1	A17	MTIOC3A/TCLKD/ PO18	SCK5/SSLA2-A/ ET0_ERXD2		IRQ12	

Note 1. The 176-pin LFBGA does not include the E5 pin.
Note 2. The BCLK function is multiplexed with the I/O port function for pin P53, so the port function is not available if the external bus is enabled.

Table 1.6 List of Pin and Pin Functions (176-Pin LFQFP) (1/7)

Pin Number 176-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
1	AVSS0							
2		P05					IRQ13	DA1
3	AVCC1							
4		P03					IRQ11	DA0
5	AVSS1							
6		P02		TMCI1	SCK6		IRQ10	AN120
7		P01		TMCI0	RXD6/SMISO6/ SSCL6		IRQ9	AN119
8		P00		TMRI0	TXD6/SMOSI6/ SSDA6		IRQ8	AN118
9		PF5					IRQ4	
10	EMLE							
11		PJ5		POE8#	CTS2#/RTS2#/SS2#			
12	VSS							
13		PJ3	EDACK1	MTIOC3C	ET0_EXOUT/ CTS6#/RTS6#/ CTS0#/RTS0#/ SS6#/SS0#			
14	VCL							
15	VBATT							
16	NC							
17	TRST#	PF4						
18	MD/FINED							
19	XCIN							
20	XCOUT							
21	RES#							
22	XTAL	P37						
23	VSS							
24	EXTAL	P36						
25	VCC							
26	UPSEL	P35					NMI	
27		P34		MTIOC0A/TMCI3/ PO12/POE10#	SCK6/SCK0/ ET0_LINKSTA		IRQ4	
28		P33	EDREQ1	MTIOC0D/TIOCD0/ TMRI3/PO11/POE4#/ POE11#	RXD6/RXD0/ SMISO6/ SMISO0/SSCL6/ SSCL0/CRX0	PCKO	IRQ3-DS	
29		P32		MTIOC0C/TIOCC0/ TMO3/PO10/ RTCOUT/RTCIC2/ POE0#/POE10#	TXD6/TXD0/ SMOSI6/SMOSI0/ SSDA6/SSDA0/ CTX0/ USB0_VBUSEN	VSYNC	IRQ2-DS	
30	TMS	PF3						
31	TDI	PF2			RXD1/SMISO1/ SSCL1			
32		P31		MTIOC4D/TMCI2/ PO9/RTCIC1	CTS1#/RTS1#/ SS1#/ET1_MDC		IRQ1-DS	
33		P30		MTIOC4B/TMRI3/ PO8/RTCIC0/POE8#	RXD1/SMISO1/ SSCL1/ ET1_MDIO		IRQ0-DS	
34	TCK	PF1			SCK1			
35	TDO	PF0			TXD1/SMOSI1/ SSDA1			

Table 1.6 List of Pin and Pin Functions (176-Pin LFQFP) (2/7)

Pin Number 176-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
36		P27	CS7#	MTIOC2B/TMCI3/PO7	SCK1/ET1_WOL			
37		P26	CS6#	MTIOC2A/TMO1/PO6	TXD1/CTS3#/ RTS3#/SMOSI1/ SS3#/SSDA1/ ET1_EXOUT			
38		P25	CS5#/ EDACK1	MTIOC4C/MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/ SSCL3/ SSIDATA1	HSYNC		ADTRG0#
39	VCC							
40		P24	CS4#/ EDREQ1	MTIOC4A/MTCLKA/ TIOCB4/TMRI1/PO4	SCK3/ USB0_VBUSEN/ SSISCK1	PIXCLK		
41	VSS							
42		P23	EDACK0	MTIOC3D/MTCLKD/ GTIOC0A-B/TIOCD3/ PO3	TXD3/CTS0#/ RTS0#/SMOSI3/ SS0#/SSDA3/ SSISCK0	PIXD7		
43		P22	EDREQ0	MTIOC3B/MTCLKC/ GTIOC1A-B/TIOCC3/ TMO0/PO2	SCK0/ USB0_OVRCURB/ USBA_OVRCURB/ AUDIO_MCLK	PIXD6		
44		P21		MTIOC1B/MTIOC4A/ GTIOC2A-B/TIOCA3/ TMCI0/PO1	RXD0/SMISO0/ SSCL0/ USB0_EXICEN/ USBA_EXICEN/ SSIWS0	PIXD5	IRQ9	
45		P20		MTIOC1A/TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/ SSDA0/USB0_ID/ USBA_ID/ SSIRXD0	PIXD4	IRQ8	
46		P17		MTIOC3A/MTIOC3B/ MTIOC4B/ GTIOC0B-B/TIOCB0/ TCLKD/TMO1/PO15/ POE8#	SCK1/TXD3/ SMOSI3/SSDA3/ SDA2-DS/ SSITXD0	PIXD3	IRQ7	ADTRG1#
47		P87		MTIOC4C/ GTIOC1B-B/TIOCA2	TXD10	PIXD2		
48		P16		MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/ TMO2/PO14/ RTCOUT	TXD1/RXD3/ SMOSI1/SMISO3/ SSDA1/SSCL3/ SCL2-DS/ USB0_VBUS/ USB0_VBUSEN/ USB0_OVRCURB		IRQ6	ADTRG0#
49		P86		MTIOC4D/ GTIOC2B-B/TIOCA0	RXD10	PIXD1		
50		P15		MTIOC0B/MTCLKB/ GTETRG-B/TIOCB2/ TCLKB/TMCI2/PO13	RXD1/SCK3/ SMISO1/SSCL1/ CRX1-DS/ USBA_VBUSEN/ SSIWS1	PIXD0	IRQ5	
51		P14		MTIOC3A/MTCLKA/ TIOCB5/TCLKA/ TMRI2/PO15	CTS1#/RTS1#/ SS1#/CTX1/ USB0_OVRCURA		IRQ4	
52		P13	WR2#/BC2#	MTIOC0B/TIOCA5/ TMO3/PO13	TXD2/SMOSI2/ SSDA2/ SDA0[FM+]		IRQ3	ADTRG1#
53		P12	WR3#/BC3#	MTIC5U/TMCI1	RXD2/SMISO2/ SSCL2/ SCL0[FM+]		IRQ2	
54	VCC_USB							
55					USB0_DM			
56					USB0_DP			

Table 1.6 List of Pin and Pin Functions (176-Pin LFQFP) (3/7)

Pin Number 176-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
57	VSS_USB							
58	AVCC_ USBA							
59	USBA_ RREF							
60	AVSS_ USBA							
61	PVSS_ USBA							
62	VSS2_ USBA							
63					USBA_DM			
64					USBA_DP			
65	VSS1_ USBA							
66	VCC_ USBA							
67		P11		MTIC5V/TMCI3	SCK2/USBA_VBUS/ USBA_VBUSEN		IRQ1	
68		P10	ALE	MTIC5W/TMRI3	USBA_OVRCURA		IRQ0	
69		P53*1	BCLK					
70		P52	RD#		RXD2/SMISO2/SSCL2			
71		P51	WR1#/BC1#/ WAIT#		SCK2			
72		P50	WR0#/WR#		TXD2/SMOSI2/SSDA2			
73	VSS							
74		P83	EDACK1	MTIOC4C/ GTIOC0A-D	CTS10#/ET0_CRS/ RMII0_CRS_DV/ SCK10			
75	VCC							
76	UB	PC7	A23/CS0#	MTIOC3A/MTCLKB/ GTIOC3A-D/TMO2/ TOC0/PO31/CACREF	TXD8/MISOA-A/ ET0_COL	MMC_D7-A	IRQ14	
77		PC6	A22/CS1#	MTIOC3C/MTCLKA/ GTIOC3B-D/TMCI2/ TIC0/PO30	RXD8/MOSIA-A/ ET0_ETXD3	MMC_D6-A	IRQ13	
78		PC5	A21/CS2#/ WAIT#	MTIOC3B/MTCLKD/ GTIOC1A-D/TMRI2/ PO29	SCK8/RSPCKA-A/ RTS8#/ET0_ETXD2	MMC_D5-A		
79		P82	EDREQ1	MTIOC4A/ GTIOC2A-D/PO28	TXD10/ET0_ETXD1/ RMII0_TXD1	MMC_D4-A		
80		P81	EDACK0	MTIOC3D/ GTIOC0B-D/PO27	RXD10/ET0_ETXD0/ RMII0_TXD0	MMC_D3-A/ SDHI_CD-A/ QIO3-A		
81		P80	EDREQ0	MTIOC3B/PO26	SCK10/RTS10#/ ET0_TX_EN/ RMII0_TXD_EN	MMC_D2-A/ SDHI_WP-A/ QIO2-A		
82		PC4	A20/CS3#	MTIOC3D/MTCLKC/ GTETRG-D/TMCI1/ PO25/POE0#	SCK5/CTS8#/SSLA0-A/ET0_TX_CLK	MMC_D1-A/ SDHI_D1-A/ QIO1-A/QMI-A		
83		PC3	A19	MTIOC4D/ GTIOC1B-D/TCLKB/ PO24	TXD5/SMOSI5/ SSDA5/ ET0_TX_ER	MMC_D0-A/ SDHI_D0-A/ QIO0-A/ QMO-A		
84		P77	CS7#	PO23	TXD11/ET0_RX_ER/ RMII0_RX_ER	MMC_CLK-A/ SDHI_CLK-A/ QSPCLK-A		

Table 1.6 List of Pin and Pin Functions (176-Pin LFQFP) (4/7)

Pin Number 176-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
85		P76	CS6#	PO22	RXD11/ET0_RX_CLK/ REF50CK0	MMC_CMD-A/ SDHI_CMD-A/ QSSL-A		
86		PC2	A18	MTIOC4B/ GTIOC2B-D/TCLKA/ PO21	RXD5/SMISO5/ SSCL5/SSLA3-A/ ET0_RX_DV	MMC_CD-A/ SDHI_D3-A		
87		P75	CS5#	PO20	SCK11/RTS11#/ ET0_ERXD0/ RMII0_RXD0	MMC_RES#-A/ SDHI_D2-A		
88		P74	A20/CS4#	PO19	CTS11#/ET0_ERXD1/ RMII0_RXD1			
89		PC1	A17	MTIOC3A/TCLKD/ PO18	SCK5/SSLA2-A/ ET0_ERXD2		IRQ12	
90	VCC							
91		PC0	A16	MTIOC3C/TCLKC/ PO17	CTS5#/RTS5#/SS5#/ SSLA1-A/ET0_ERXD3		IRQ14	
92	VSS							
93		P73	CS3#	PO16	ET0_WOL			
94		PB7	A15	MTIOC3B/TIOCB5/ PO31	TXD9/ET0_CRS/ RMII0_CRS_DV			
95		PB6	A14	MTIOC3D/TIOCA5/ PO30	RXD9/ET0_ETXD1/ RMII0_TXD1			
96		PB5	A13	MTIOC2A/MTIOC1B/ TIOCB4/TMRI1/PO29/ POE4#	SCK9/RTS9#/ ET0_ETXD0/ RMII0_TXD0			
97		PB4	A12	TIOCA4/PO28	CTS9#/ET0_TX_EN/ RMII0_TXD_EN			
98		PB3	A11	MTIOC0A/MTIOC4A/ TIOCD3/TCLKD/ TMO0/PO27/POE11#	SCK4/SCK6/ ET0_RX_ER/ RMII0_RX_ER			
99		PB2	A10	TIOCC3/TCLKC/ PO26	CTS4#/RTS4#/CTS6#/ RTS6#/SS4#/SS6#/ ET0_RX_CLK/ REF50CK0			
100		PB1	A9	MTIOC0C/MTIOC4C/ TIOCB3/TMCI0/PO25	TXD4/TXD6/SMOSI4/ SMOSI6/SSDA4/ SSDA6/ET0_ERXD0/ RMII0_RXD0		IRQ4-DS	
101		P72	A19/CS2#		ET0_MDC			
102		P71	A18/CS1#		ET0_MDIO			
103	VCC							
104		PB0	A8	MTIC5W/TIOCA3/ PO24	RXD4/RXD6/SMISO4/ SMISO6/SSCL4/ SSCL6/ET0_ERXD1/ RMII0_RXD1		IRQ12	
105	VSS							
106		PA7	A7	TIOCB2/PO23	MISOA-B/ ET0_WOL			
107		PA6	A6	MTIC5V/MTCLKB/ GTETRG-C/TIOCA2/ TMCI3/PO22/POE10#	CTS5#/RTS5#/SS5#/ MOSIA-B/ ET0_EXOUT			
108		PA5	A5	MTIOC6B/ GTIOC0A-C/TIOCB1/ PO21	RSPCKA-B/ ET0_LINKSTA			
109		PA4	A4	MTIC5U/MTCLKA/ TIOCA1/TMRI0/PO20	TXD5/SMOSI5/ SSDA5/SSLA0-B/ ET0_MDC		IRQ5-DS	
110		PA3	A3	MTIOC0D/MTCLKD/ TIOCD0/TCLKB/PO19	RXD5/SMISO5/ SSCL5/ ET0_MDIO		IRQ6-DS	

Table 1.6 List of Pin and Pin Functions (176-Pin LFQFP) (5/7)

Pin Number 176-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
111	TRDATA3	PG7	D31		ET1_TX_ER			
112		PA2	A2	MTIOC7A/ GTIOC1A-C/PO18	RXD5/SMISO5/ SSCL5/SSLA3-B			
113	TRDATA2	PG6	D30		ET1_ETXD3			
114		PA1	A1/DQM3	MTIOC0B/MTCLKC/ MTIOC7B/ GTIOC2A-C/TIOCB0/ PO17	SCK5/SSLA2-B/ ET0_WOL		IRQ11	
115	VCC							
116	TRCLK	PG5	D29		ET1_ETXD2			
117	VSS							
118		PA0	A0/BC0#/ DQM2	MTIOC4A/MTIOC6D/ GTIOC0B-C/TIOCA0/ CACREF/PO16	SSLA1-B/ ET0_TX_EN/ RMII0_TXD_EN			
119	TRSYNC	PG4	D28		ET1_ETXD1/ RMII1_TXD1			
120		P67	CS7#/DQM1	MTIOC7C/ GTIOC1B-C	CRX2		IRQ15	
121	TRDATA1	PG3	D27		ET1_ETXD0/ RMII1_TXD0			
122		P66	CS6#/DQM0	MTIOC7D/ GTIOC2B-C	CTX2			
123	TRDATA0	PG2	D26		ET1_TX_CLK			
124		P65	CS5#/CKE					
125		PE7	D15[A15/D15]	MTIOC6A/ GTIOC3A-E/TOC1		MMC_RES#-B/ SDHI_WP-B	IRQ7	AN105
126		PE6	D14[A14/D14]	MTIOC6C/ GTIOC3B-E/TIC1		MMC_CD-B/ SDHI_CD-B	IRQ6	AN104
127	VCC							
128		P70	SDCLK					
129	VSS							
130		PE5	D13[A13/D13]	MTIOC4C/MTIOC2B/ GTIOC0A-A	ET0_RX_CLK/ REF50CK0		IRQ5	AN103
131		PE4	D12[A12/D12]	MTIOC4D/MTIOC1A/ GTIOC1A-A/PO28	ET0_ERXD2			AN102
132		PE3	D11[A11/D11]	MTIOC4B/ GTIOC2A-A/PO26/ POE8#/TOC3	CTS12#/RTS12#/ SS12#/ ET0_ERXD3	MMC_D7-B		AN101
133		PE2	D10[A10/D10]	MTIOC4A/ GTIOC0B-A/PO23/ TIC3	RXD12/SMISO12/ SSCL12/ RXDX12	MMC_D6-B	IRQ7-DS	AN100
134		PE1	D9[A9/D9]	MTIOC4C/MTIOC3B/ GTIOC1B-A/PO18	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12	MMC_D5-B		ANEX1
135		PE0	D8[A8/D8]	MTIOC3D/ GTIOC2B-A	SCK12	MMC_D4-B		ANEX0
136		P64	CS4#/WE#					
137		P63	CS3#/CAS#					
138		P62	CS2#/RAS#					
139		P61	CS1#/SDCS#					
140	VSS							
141		P60	CS0#		ET1_TX_EN/ RMII1_TXD_EN			
142	VCC							

Table 1.6 List of Pin and Pin Functions (176-Pin LFQFP) (6/7)

Pin Number 176-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
143		PD7	D7[A7/D7]	MTIC5U/POE0#		MMC_D1-B/ SDHI_D1-B/ QIO1-B/QMI-B	IRQ7	AN107
144		PG1	D25		ET1_RX_ER/ RMII1_RX_ER			
145		PD6	D6[A6/D6]	MTIC5V/MTIOC8A/ POE4#		MMC_D0-B/ SDHI_D0-B/ QIO0-B/ QMO-B	IRQ6	AN106
146		PG0	D24		ET1_RX_CLK/ REF50CK1			
147		PD5	D5[A5/D5]	MTIC5W/MTIOC8C/ POE10#		MMC_CLK-B/ SDHI_CLK-B/ QSPCLK-B	IRQ5	AN113
148		PD4	D4[A4/D4]	MTIOC8B/POE11#		MMC_CMD-B/ SDHI_CMD-B/ QSSL-B	IRQ4	AN112
149		P97	A23/D23		ET1_ERXD3			
150		PD3	D3[A3/D3]	MTIOC8D/ GTIOC0A-E/POE8#/ TOC2		MMC_D3-B/ SDHI_D3-B/ QIO3-B	IRQ3	AN111
151	VSS							
152		P96	A22/D22		ET1_ERXD2			
153	VCC							
154		PD2	D2[A2/D2]	MTIOC4D/ GTIOC0B-E/TIC2	CRX0	MMC_D2-B/ SDHI_D2-B/ QIO2_B	IRQ2	AN110
155		P95	A21/D21		ET1_ERXD1/ RMII1_RXD1			
156		PD1	D1[A1/D1]	MTIOC4B/ GTIOC1A-E/POE0#	CTX0		IRQ1	AN109
157		P94	A20/D20		ET1_ERXD0/ RMII1_RXD0			
158		PD0	D0[A0/D0]	GTIOC1B-E/POE4#			IRQ0	AN108
159		P93	A19/D19	POE0#	ET1_LINKSTA/CTS7#/ RTS7#/SS7#			AN117
160		P92	A18/D18	POE4#	ET1_CRS/ RMII1_CRS_DV/ RXD7/SMISO7/SSCL7			AN116
161		P91	A17/D17		ET1_COL/SCK7			AN115
162	VSS							
163		P90	A16/D16		ET1_RX_DV/ TXD7/SMOSI7/SSDA7			AN114
164	VCC							
165		P47					IRQ15-DS	AN007
166		P46					IRQ14-DS	AN006
167		P45					IRQ13-DS	AN005
168		P44					IRQ12-DS	AN004
169		P43					IRQ11-DS	AN003
170		P42					IRQ10-DS	AN002
171		P41					IRQ9-DS	AN001
172	VREFL0							

Table 1.6 List of Pin and Pin Functions (176-Pin LFQFP) (7/7)

Pin Number 176-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
173		P40					IRQ8-DS	AN000
174	VREFH0							
175	AVCC0							
176		P07					IRQ15	ADTRG0#

Note 1. The BCLK function is multiplexed with the I/O port function for pin P53, so the port function is not available if the external bus is enabled.

Table 1.7 List of Pin and Pin Functions (145-Pin TFLGA) (1/5)

Pin Number 145-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
A1	AVSS0							
A2		P07					IRQ15	ADTRG0#
A3		P40					IRQ8-DS	AN000
A4		P42					IRQ10-DS	AN002
A5		P45					IRQ13-DS	AN005
A6		P90	A16		TXD7/SMOSI7/SSDA7			AN114
A7		P92	A18	POE4#	RXD7/SMISO7/SSCL7			AN116
A8		PD2	D2[A2/D2]	MTIOC4D/ GTIOC0B-E/TIC2	CRX0	MMC_D2-B/ SDHI_D2-B/ QIO2-B	IRQ2	AN110
A9		PD6	D6[A6/D6]	MTIC5V/MTIOC8A/ POE4#		MMC_D0-B/ SDHI_D0-B/ QIO0-B/ QMO-B	IRQ6	AN106
A10	VSS							
A11		P62	CS2#/RAS#					
A12		PE1	D9[A9/D9]	MTIOC4C/MTIOC3B/ GTIOC1B-A/PO18	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12	MMC_D5-B		ANEX1
A13		PE3	D11[A11/D11]	MTIOC4B/ GTIOC2A-A/PO26/ POE8#/TOC3	CTS12#/RTS12#/ SS12#/ET0_ERXD3/	MMC_D7-B		AN101
B1	AVCC1							
B2	AVCC0							
B3		P05					IRQ13	DA1
B4	VREFL0							
B5		P43					IRQ11-DS	AN003
B6		P47					IRQ15-DS	AN007
B7		P91	A17		SCK7			AN115
B8		PD0	D0[A0/D0]	GTIOC1B-E/POE4#			IRQ0	AN108
B9		PD4	D4[A4/D4]	MTIOC8B/POE11#		MMC_CMD-B/ SDHI_CMD-B/ QSSL-B	IRQ4	AN112
B10	VCC							
B11		P61	CS1#/SDCS#					
B12		PE2	D10[A10/D10]	MTIOC4A/ GTIOC0B-A/PO23/ TIC3	RXD12/SMISO12/ SSCL12/RXDX12/	MMC_D6-B	IRQ7-DS	AN100
B13		PE4	D12[A12/D12]	MTIOC4D/MTIOC1A/ GTIOC1A-A/PO28	ET0_ERXD2			AN102
C1	AVSS1							
C2		P02		TMCI1	SCK6		IRQ10	AN120
C3	VREFH0							
C4		P41					IRQ9-DS	AN001
C5		P46					IRQ14-DS	AN006
C6	VSS							
C7		PD1	D1[A1/D1]	MTIOC4B/ GTIOC1A-E/POE0#	CTX0		IRQ1	AN109
C8		PD3	D3[A3/D3]	MTIOC8D/ GTIOC0A-E/POE8#/ TOC2		MMC_D3-B/ SDHI_D3-B/ QIO3-B	IRQ3	AN111

Table 1.7 List of Pin and Pin Functions (145-Pin TFLGA) (2/5)

Pin Number 145-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
C9		PD7	D7[A7/D7]	MTIC5U/POE0#		MMC_D1-B/ SDHI_D1-B/ QIO1-B/QMI-B	IRQ7	AN107
C10		P63	CS3#/CAS#					
C11		PE0	D8[A8/D8]	MTIOC3D/ GTIOC2B-A	SCK12	MMC_D4-B		ANEX0
C12		P70	SDCLK					
C13	VSS							
D1		P00		TMRI0	TXD6/SMOSI6/SSDA6		IRQ8	AN118
D2		PF5					IRQ4	
D3		P03					IRQ11	DA0
D4		P01		TMCI0	RXD6/SMISO6/SSCL6		IRQ9	AN119
D5	VCC							
D6		P93	A19	POE0#	CTS7#/RTS7#/SS7#			AN117
D7		PD5	D5[A5/D5]	MTIC5W/MTIOC8C/ POE10#		MMC_CLK-B/ SDHI_CLK-B/ QSPCLK-B	IRQ5	AN113
D8		P60	CS0#					
D9		P64	CS4#/WE#					
D10		PE7	D15[A15/D15]	MTIOC6A/ GTIOC3A-E/TOC1		MMC_RES#-B/ SDHI_WP-B	IRQ7	AN105
D11	VCC							
D12		PE5	D13[A13/D13]	MTIOC4C/MTIOC2B/ GTIOC0A-A	ET0_RX_CLK/ REF50CK0		IRQ5	AN103
D13		PE6	D14[A14/D14]	MTIOC6C/GTIOC3B-E/TIC1		MMC_CD-B/ SDHI_CD-B	IRQ6	AN104
E1	VSS							
E2	VCL							
E3		PJ5		POE8#	CTS2#/RTS2#/SS2#			
E4	EMLE							
E5		P44					IRQ12-DS	AN004
E10		PA0	A0/BC0#	MTIOC4A/MTIOC6D/ GTIOC0B-C/TIOCA0/ CACREF/PO16	SSLA1-B/ ET0_TX_EN/ RMII0_TXD_EN			
E11		P66	CS6#/DQM0	MTIOC7D/ GTIOC2B-C	CTX2			
E12		P65	CS5#/CKE					
E13		P67	CS7#/DQM1	MTIOC7C/ GTIOC1B-C	CRX2		IRQ15	
F1	XCIN							
F2	XCOUT							
F3		PJ3	EDACK1	MTIOC3C	ET0_EXOUT/CTS6#/ RTS6#/CTS0#/RTS0#/ SS6#/SS0#			
F4	VBATT							
F10		PA3	A3	MTIOC0D/MTCLKD/ TIOCD0/TCLKB/PO19	RXD5/SMISO5/ SSCL5/ET0_MDIO		IRQ6-DS	
F11	VSS							
F12		PA1	A1	MTIOC0B/MTCLKC/ MTIOC7B/ GTIOC2A-C/TIOCB0/ PO17	SCK5/SSLA2-B/ ET0_WOL		IRQ11	

Table 1.7 List of Pin and Pin Functions (145-Pin TFLGA) (3/5)

Pin Number 145-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
F13		PA2	A2	MTIOC7A/ GTIOC1A-C/PO18	RXD5/SMISO5/ SSCL5/SSLA3-B			
G1	XTAL	P37						
G2	RES							
G3	MD/FINED							
G4	BSCANP							
G10		PA5	A5	MTIOC6B/TIOCB1/ GTIOC0A-C/PO21	RSPCKA-B/ ET0_LINKSTA			
G11		PA6	A6	MTIC5V/MTCLKB/ GTETRG-C/TIOCA2/ TMCI3/PO22/POE10#	CTS5#/RTS5#/SS5#/ MOSIA-B/ ET0_EXOUT			
G12	VCC							
G13		PA4	A4	MTIC5U/MTCLKA/ TIOCA1/TMRI0/PO20	TXD5/SMOSI5/ SSDA5/SSLA0-B/ ET0_MDC		IRQ5-DS	
H1	EXTAL	P36						
H2	VCC							
H3	VSS							
H4	UPSEL	P35					NMI	
H10		P72	A19/CS2#		ET0_MDC			
H11		P71	A18/CS1#		ET0_MDIO			
H12		PB0	A8	MTIC5W/TIOCA3/ PO24	RXD4/RXD6/SMISO4/ SMISO6/SSCL4/ SSCL6/ET0_ERXD1/ RMII0_RXD1		IRQ12	
H13		PA7	A7	TIOCB2/PO23	MISOA-B/ET0_WOL			
J1	TRST#	P34		MTIOC0A/TMCI3/ PO12/POE10#	SCK6/SCK0/ ET0_LINKSTA		IRQ4	
J2		P33	EDREQ1	MTIOC0D/TIOCD0/ TMRI3/PO11/POE4#/ POE11#	RXD6/RXD0/SMISO6/ SMISO0/SSCL6/ SSCL0/CRX0	PCKO	IRQ3-DS	
J3		P32		MTIOC0C/TIOCC0/ TMO3/PO10/ RTCOUT/RTCIC2/ POE0#/POE10#	TXD6/TXD0/SMOSI6/ SMOSI0/SSDA6/ SSDA0/CTX0/ USB0_VBUSEN	VSYNC	IRQ2-DS	
J4	TDI	P30		MTIOC4B/TMRI3/ PO8/RTCIC0/POE8#	RXD1/SMISO1/SSCL1		IRQ0-DS	
J10		PB3	A11	MTIOC0A/MTIOC4A/ TIOCD3/TCLKD/ TMO0/PO27/POE11#	SCK4/SCK6/ ET0_RX_ER/ RMII0_RX_ER			
J11		PB4	A12	TIOCA4/PO28	CTS9#/ET0_TX_EN/ RMII0_TXD_EN			
J12		PB2	A10	TIOCC3/TCLKC/ PO26	CTS4#/RTS4#/CTS6#/ RTS6#/SS4#/SS6#/ ET0_RX_CLK/ REF50CK0			
J13		PB1	A9	MTIOC0C/MTIOC4C/ TIOCB3/TMCI0/PO25	TXD4/TXD6/SMOSI4/ SMOSI6/SSDA4/ SSDA6/ET0_ERXD0/ RMII0_RXD0		IRQ4-DS	
K1	TCK	P27	CS7#	MTIOC2B/TMCI3/PO7	SCK1			
K2	TDO	P26	CS6#	MTIOC2A/TMO1/PO6	TXD1/CTS3#/RTS3#/ SMOSI1/SS3#/SSDA1			
K3	TMS	P31		MTIOC4D/TMCI2/ PO9/RTCIC1	CTS1#/RTS1#/SS1#		IRQ1-DS	
K4		P15		MTIOC0B/MTCLKB/ GTETRG-B/TIOCB2/ TCLKB/TMCI2/PO13	RXD1/SCK3/SMISO1/ SSCL1/CRX1-DS/ SSIWS1	PIXD0	IRQ5	

Table 1.7 List of Pin and Pin Functions (145-Pin TFLGA) (4/5)

Pin Number 145-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
K5	TRDATA2	P54	ALE/EDACK0	MTIOC4B/TMCI1	CTS2#/RTS2#/SS2#/ CTX1/ET0_LINKSTA			
K6		P53*1	BCLK					
K7		P51	WR1#/BC1#/ WAIT#		SCK2			
K8	VCC							
K9	TRDATA0	P80	EDREQ0	MTIOC3B/PO26	SCK10/RTS10#/ ET0_TX_EN/ RMII0_TXD_EN	MMC_D2-A/ SDHI_WP-A/ QIO2-A		
K10		P76	CS6#	PO22	RXD11/ET0_RX_CLK/ REF50CK0	MMC_CMD-A/ SDHI_CMD-A/ QSSL-A		
K11		PB7	A15	MTIOC3B/TIOCB5/ PO31	TXD9/ET0_CRS/ RMII0_CRS_DV			
K12		PB6	A14	MTIOC3D/TIOCA5/ PO30	RXD9/ET0_ETXD1/ RMII0_TXD1			
K13		PB5	A13	MTIOC2A/MTIOC1B/ TIOCB4/TMRI1/PO29/ POE4#	SCK9/RTS9#/ ET0_ETXD0/ RMII0_TXD0			
L1		P25	CS5#/ EDACK1	MTIOC4C/MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/ SSCL3/SSIDATA1	HSYNC		ADTRG0#
L2		P23	EDACK0	MTIOC3D/MTCLKD/ GTIOC0A-B/TIOCD3/ PO3	TXD3/CTS0#/RTS0#/ SMOSI3/SS0#/ SSDA3/SSISCK0	PIXD7		
L3		P16		MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/ TMO2/PO14/ RTCOUT	TXD1/RXD3/SMOSI1/ SMISO3/SSDA1/ SSCL3/SCL2-DS/ USB0_VBUS/ USB0_VBUSEN/ USB0_OVRCURB		IRQ6	ADTRG0#
L4		P24	CS4#/ EDREQ1	MTIOC4A/MTCLKA/ TIOCB4/TMRI1/PO4	SCK3/ USB0_VBUSEN/ SSISCK1	PIXCLK		
L5		P13		MTIOC0B/TIOCA5/ TMO3/PO13	TXD2/SMOSI2/ SSDA2/SDA0[FM+]		IRQ3	ADTRG1#
L6		P56	EDACK1	MTIOC3C/TIOCA1				
L7		P52	RD#		RXD2/SMISO2/SSCL2			
L8	TRCLK	P83	EDACK1	MTIOC4C/ GTIOC0A-D	CTS10#/ET0_CRS/ RMII0_CRS_DV/ SCK10			
L9		PC5	A21/CS2#/ WAIT#	MTIOC3B/MTCLKD/ GTIOC1A-D/TMRI2/ PO29	SCK8/RSPCKA-A/ RTS8#/ET0_ETXD2	MMC_D5-A		
L10		PC4	A20/CS3#	MTIOC3D/MTCLKC/ GTETRG-D/TMCI1/ PO25/POE0#	SCK5/CTS8#/ SSLA0-A/ ET0_TX_CLK	MMC_D1-A/ SDHI_D1-A/ QIO1-A/QMI-A		
L11		PC2	A18	MTIOC4B/ GTIOC2B-D/TCLKA/ PO21	RXD5/SMISO5/ SSCL5/SSLA3-A/ ET0_RX_DV	MMC_CD-A/ SDHI_D3-A		
L12		P73	CS3#	PO16	ET0_WOL			
L13	VSS							
M1		P22	EDREQ0	MTIOC3B/MTCLKC/ GTIOC1A-B/TIOCC3/ TMO0/PO2	SCK0/ USB0_OVRCURB/ AUDIO_MCLK	PIXD6		
M2		P17		MTIOC3A/MTIOC3B/ MTIOC4B/ GTIOC0B-B/TIOCB0/ TCLKD/TMO1/PO15/ POE8#	SCK1/TXD3/SMOSI3/ SSDA3/SDA2-DS/ SSITXD0	PIXD3	IRQ7	ADTRG1#
M3		P86		MTIOC4D/ GTIOC2B-B/TIOCA0	RXD10	PIXD1		

Table 1.7 List of Pin and Pin Functions (145-Pin TFLGA) (5/5)

Pin Number 145-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
M4		P12		TMCI1	RXD2/SMISO2/ SSCL2/SCL0[FM+]		IRQ2	
M5	VCC_USB							
M6	VSS_USB							
M7		P50	WR0#/WR#		TXD2/SMOSI2/SSDA2			
M8		PC6	A22/CS1#	MTIOC3C/MTCLKA/ GTIOC3B-D/TMCI2/ TIC0/PO30	RXD8/MOSIA-A/ ET0_ETXD3	MMC_D6-A	IRQ13	
M9	TRDATA1	P81	EDACK0	MTIOC3D/ GTIOC0B-D/PO27	RXD10/ET0_ETXD0/ RMII0_TXD0	MMC_D3-A/ SDHI_CD-A/ QIO3-A		
M10		P77	CS7#	PO23	TXD11/ET0_RX_ER/ RMII0_RX_ER	MMC_CLK-A/ SDHI_CLK-A/ QSPCLK-A		
M11		PC0	A16	MTIOC3C/TCLKC/ PO17	CTS5#/RTS5#/SS5#/ SSLA1-A/ET0_ERXD3		IRQ14	
M12		PC1	A17	MTIOC3A/TCLKD/ PO18	SCK5/SSLA2-A/ ET0_ERXD2		IRQ12	
M13	VCC							
N1		P21		MTIOC1B/MTIOC4A/ GTIOC2A-B/TIOCA3/ TMCI0/PO1	RXD0/SMISO0/ SSCL0/ USB0_EXICEN/ SSIWS0	PIXD5	IRQ9	
N2		P20		MTIOC1A/TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/ SSDA0/USB0_ID/ SSIRXD0	PIXD4	IRQ8	
N3		P87		MTIOC4C/ GTIOC1B-B/TIOCA2	TXD10	PIXD2		
N4		P14		MTIOC3A/MTCLKA/ TIOCB5/TCLKA/ TMRI2/PO15	CTS1#/RTS1#/SS1#/ CTX1/ USB0_OVRCURA		IRQ4	
N5					USB0_DM			
N6					USB0_DP			
N7	TRDATA3	P55	WAIT#/ EDREQ0	MTIOC4D/TMO3	CRX1/ET0_EXOUT		IRQ10	
N8	VSS							
N9	UB	PC7	A23/CS0#	MTIOC3A/MTCLKB/ GTIOC3A-D/TMO2/ TOC0/PO31/CACREF	TXD8/MISOA-A/ ET0_COL	MMC_D7-A	IRQ14	
N10	TRSYNC	P82	EDREQ1	MTIOC4A/ GTIOC2A-D/PO28	TXD10/ET0_ETXD1/ RMII0_TXD1	MMC_D4-A		
N11		PC3	A19	MTIOC4D/ GTIOC1B-D/TCLKB/ PO24	TXD5/SMOSI5/ SSDA5/ET0_TX_ER	MMC_D0-A/ SDHI_D0-A/ QIO0-A/ QMO-A		
N12		P75	CS5#	PO20	SCK11/RTS11#/ ET0_ERXD0/ RMII0_RXD0	MMC_RES#-A/ SDHI_D2-A		
N13		P74	A20/CS4#	PO19	CTS11#/ET0_ERXD1/ RMII0_RXD1			

Note 1. The BCLK function is multiplexed with the I/O port function for pin P53, so the port function is not available if the external bus is enabled.

Table 1.8 List of Pin and Pin Functions (144-Pin LFQFP) (1/5)

Pin Number 144-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
1	AVSS0							
2		P05					IRQ13	DA1
3	AVCC1							
4		P03					IRQ11	DA0
5	AVSS1							
6		P02		TMCI1	SCK6		IRQ10	AN120
7		P01		TMCI0	RXD6/SMISO6/SSCL6		IRQ9	AN119
8		P00		TMRI0	TXD6/SMOSI6/SSDA6		IRQ8	AN118
9		PF5					IRQ4	
10	EMLE							
11		PJ5		POE8#	CTS2#/RTS2#/SS2#			
12	VSS							
13		PJ3	EDACK1	MTIOC3C	ET0_EXOUT/CTS6#/ RTS6#/CTS0#/RTS0#/ SS6#/SS0#			
14	VCL							
15	VBATT							
16	MD/FINED							
17	XCIN							
18	XCOUT							
19	RES							
20	XTAL	P37						
21	VSS							
22	EXTAL	P36						
23	VCC							
24		P35					NMI	
25	TRST#	P34		MTIOC0A/TMCI3/ PO12/POE10#	SCK6/SCK0/ ET0_LINKSTA		IRQ4	
26		P33	EDREQ1	MTIOC0D/TIOCD0/ TMRI3/PO11/POE4#/ POE11#	RXD6/RXD0/SMISO6/ SMISO0/SSCL6/ SSCL0/CRX0	PCKO	IRQ3-DS	
27		P32		MTIOC0C/TIOCC0/ TMO3/PO10/ RTCOUT/RTCIC2/ POE0#/POE10#	TXD6/TXD0/SMOSI6/ SMOSI0/SSDA6/ SSDA0/CTX0/ USB0_VBUSEN	VSYNC	IRQ2-DS	
28	TMS	P31		MTIOC4D/TMCI2/ PO9/RTCIC1	CTS1#/RTS1#/SS1#		IRQ1-DS	
29	TDI	P30		MTIOC4B/TMRI3/ PO8/RTCIC0/POE8#	RXD1/SMISO1/SSCL1		IRQ0-DS	
30	TCK	P27	CS7#	MTIOC2B/TMCI3/PO7	SCK1			
31	TDO	P26	CS6#	MTIOC2A/TMO1/PO6	TXD1/CTS3#/RTS3#/ SMOSI1/SS3#/SSDA1			
32		P25	CS5#/ EDACK1	MTIOC4C/MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/ SSCL3/SSIDATA1	HSYNC		ADTRG0#
33		P24	CS4#/ EDREQ1	MTIOC4A/MTCLKA/ TIOCB4/TMRI1/PO4	SCK3/ USB0_VBUSEN/ SSISCK1	PIXCLK		
34		P23	EDACK0	MTIOC3D/MTCLKD/ GTIOC0A-B/TIOCD3/ PO3	TXD3/CTS0#/RTS0#/ SMOSI3/SS0#/ SSDA3/SSISCK0	PIXD7		
35		P22	EDREQ0	MTIOC3B/MTCLKC/ GTIOC1A-B/TIOCC3/ TMO0/PO2	SCK0/ USB0_OVRCURB/ AUDIO_MCLK	PIXD6		

Table 1.8 List of Pin and Pin Functions (144-Pin LFQFP) (2/5)

Pin Number 144-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
36		P21		MTIOC1B/MTIOC4A/ GTIOC2A-B/TIOCA3/ TMCI0/PO1	RXD0/SMISO0/ SSCL0/ USB0_EXICEN/ SSIWS0	PIXD5	IRQ9	
37		P20		MTIOC1A/TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/ SSDA0/USB0_ID/ SSIRXD0	PIXD4	IRQ8	
38		P17		MTIOC3A/MTIOC3B/ MTIOC4B/ GTIOC0B-B/TIOCB0/ TCLKD/TMO1/PO15/ POE8#	SCK1/TXD3/SMOSI3/ SSDA3/SDA2-DS/ SSITXD0	PIXD3	IRQ7	ADTRG1#
39		P87		MTIOC4C/ GTIOC1B-B/TIOCA2	TXD10	PIXD2		
40		P16		MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/ TMO2/PO14/ RTCOUT	TXD1/RXD3/SMOSI1/ SMISO3/SSDA1/ SSCL3/SCL2-DS/ USB0_VBUS/ USB0_VBUSEN/ USB0_OVRCURB		IRQ6	ADTRG0#
41		P86		MTIOC4D/ GTIOC2B-B/TIOCA0	RXD10	PIXD1		
42		P15		MTIOC0B/MTCLKB/ GTETRG-B/TIOCB2/ TCLKB/TMCI2/PO13	RXD1/SCK3/SMISO1/ SSCL1/CRX1-DS/ SSIWS1	PIXD0	IRQ5	
43		P14		MTIOC3A/MTCLKA/ TIOCB5/TCLKA/ TMRI2/PO15	CTS1#/RTS1#/SS1#/ CTX1/ USB0_OVRCURA		IRQ4	
44		P13		MTIOC0B/TIOCA5/ TMO3/PO13	TXD2/SMOSI2/ SSDA2/SDA0[FM+]		IRQ3	ADTRG1#
45		P12		TMCI1	RXD2/SMISO2/ SSCL2/SCL0[FM+]		IRQ2	
46	VCC_USB							
47					USB0_DM			
48					USB0_DP			
49	VSS_USB							
50		P56	EDACK1	MTIOC3C/TIOCA1				
51	TRDATA3	P55	WAIT#/ EDREQ0	MTIOC4D/TMO3	CRX1/ET0_EXOUT		IRQ10	
52	TRDATA2	P54	ALE/EDACK0	MTIOC4B/TMCI1	CTS2#/RTS2#/SS2#/ CTX1/ET0_LINKSTA			
53		P53*1	BCLK					
54		P52	RD#		RXD2/SMISO2/SSCL2			
55		P51	WR1#/BC1#/ WAIT#		SCK2			
56		P50	WR0#/WR#		TXD2/SMOSI2/SSDA2			
57	VSS							
58	TRCLK	P83	EDACK1	MTIOC4C/ GTIOC0A-D	CTS10#/ET0_CRS/ RMII0_CRS_DV/ SCK10			
59	VCC							
60	UB	PC7	A23/CS0#	MTIOC3A/MTCLKB/ GTIOC3A-D/TMO2/ TOC0/PO31/CACREF	TXD8/MISOA-A/ ET0_COL	MMC_D7-A	IRQ14	
61		PC6	A22/CS1#	MTIOC3C/MTCLKA/ GTIOC3B-D/TMCI2/ TIC0/PO30	RXD8/MOSIA-A/ ET0_ETXD3	MMC_D6-A	IRQ13	
62		PC5	A21/CS2#/ WAIT#	MTIOC3B/MTCLKD/ GTIOC1A-D/TMRI2/ PO29	SCK8/RSPCKA-A/ RTS8#/ET0_ETXD2	MMC_D5-A		

Table 1.8 List of Pin and Pin Functions (144-Pin LFQFP) (3/5)

Pin Number 144-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
63	TRSYNC	P82	EDREQ1	MTIOC4A/ GTIOC2A-D/PO28	TXD10/ET0_ETXD1/ RMII0_TXD1	MMC_D4-A		
64	TRDATA1	P81	EDACK0	MTIOC3D/ GTIOC0B-D/PO27	RXD10/ET0_ETXD0/ RMII0_TXD0	MMC_D3-A/ SDHI_CD-A/ QIO3-A		
65	TRDATA0	P80	EDREQ0	MTIOC3B/PO26	SCK10/RTS10#/ ET0_TX_EN/ RMII0_TXD_EN	MMC_D2-A/ SDHI_WP-A/ QIO2-A		
66		PC4	A20/CS3#	MTIOC3D/MTCLKC/ GTETRG-D/TMCI1/ PO25/POE0#	SCK5/CTS8#/ SSLA0-A/ ET0_TX_CLK/	MMC_D1-A/ SDHI_D1-A/ QIO1-A/QMI-A		
67		PC3	A19	MTIOC4D/ GTIOC1B-D/TCLKB/ PO24	TXD5/SMOSI5/ SSDA5/ET0_TX_ER	MMC_D0-A/ SDHI_D0-A/ QIO0-A/ QMO-A		
68		P77	CS7#	PO23	TXD11/ET0_RX_ER/ RMII0_RX_ER	MMC_CLK-A/ SDHI_CLK-A/ QSPCLK-A		
69		P76	CS6#	PO22	RXD11/ET0_RX_CLK/ REF50CK0	MMC_CMD-A/ SDHI_CMD-A/ QSSL-A		
70		PC2	A18	MTIOC4B/ GTIOC2B-D/TCLKA/ PO21	RXD5/SMISO5/ SSCL5/SSLA3-A/ ET0_RX_DV	MMC_CD-A/ SDHI_D3-A		
71		P75	CS5#	PO20	SCK11/RTS11/ ET0_ERXD0/ RMII0_RXD0	MMC_RES#-A/ SDHI_D2-A		
72		P74	A20/CS4#	PO19	CTS11#/ET0_ERXD1/ RMII0_RXD1			
73		PC1	A17	MTIOC3A/TCLKD/ PO18	SCK5/SSLA2-A/ ET0_ERXD2		IRQ12	
74	VCC							
75		PC0	A16	MTIOC3C/TCLKC/ PO17	CTS5#/RTS5#/SS5#/ SSLA1-A/ET0_ERXD3		IRQ14	
76	VSS							
77		P73	CS3#	PO16	ET0_WOL			
78		PB7	A15	MTIOC3B/TIOCB5/ PO31	TXD9/ET0_CRS/ RMII0_CRS_DV			
79		PB6	A14	MTIOC3D/TIOCA5/ PO30	RXD9/ET0_ETXD1/ RMII0_TXD1			
80		PB5	A13	MTIOC2A/MTIOC1B/ TIOCB4/TMRI1/PO29/ POE4#	SCK9/RTS9#/ ET0_ETXD0/ RMII0_TXD0			
81		PB4	A12	TIOCA4/PO28	CTS9#/ET0_TX_EN/ RMII0_TXD_EN			
82		PB3	A11	MTIOC0A/MTIOC4A/ TIOCD3/TCLKD/ TMO0/PO27/POE11#	SCK4/SCK6/ ET0_RX_ER/ RMII0_RX_ER			
83		PB2	A10	TIOCC3/TCLKC/ PO26	CTS4#/RTS4#/CTS6#/ RTS6#/SS4#/SS6#/ ET0_RX_CLK/ REF50CK0			
84		PB1	A9	MTIOC0C/MTIOC4C/ TIOCB3/TMCI0/PO25	TXD4/TXD6/SMOSI4/ SMOSI6/SSDA4/ SSDA6/ET0_ERXD0/ RMII0_RXD0		IRQ4-DS	
85		P72	A19/CS2#		ET0_MDC			
86		P71	A18/CS1#		ET0_MDIO			

Table 1.8 List of Pin and Pin Functions (144-Pin LFQFP) (4/5)

Pin Number 144-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
87		PB0	A8	MTIC5W/TIOCA3/ PO24	RXD4/RXD6/SMISO4/ SMISO6/SSCL4/ SSCL6/ET0_ERXD1/ RMII0_RXD1		IRQ12	
88		PA7	A7	TIOCB2/PO23	MISOA-B/ET0_WOL			
89		PA6	A6	MTIC5V/MTCLKB/ GTETRG-C/TIOCA2/ TMCI3/PO22/POE10#	CTS5#/RTS5#/SS5#/ MOSIA-B/ ET0_EXOUT			
90		PA5	A5	MTIOC6B/TIOCB1/ GTIOC0A-C/PO21	RSPCKA-B/ ET0_LINKSTA			
91	VCC							
92		PA4	A4	MTIC5U/MTCLKA/ TIOCA1/TMRI0/PO20	TXD5/SMOSI5/ SSDA5/SSLA0-B/ ET0_MDC		IRQ5-DS	
93	VSS							
94		PA3	A3	MTIOC0D/MTCLKD/ TIOCD0/TCLKB/PO19	RXD5/SMISO5/ SSCL5/ET0_MDIO		IRQ6-DS	
95		PA2	A2	MTIOC7A/ GTIOC1A-C/PO18	RXD5/SMISO5/ SSCL5/SSLA3-B			
96		PA1	A1	MTIOC0B/MTCLKC/ MTIOC7B/ GTIOC2A-C/TIOCB0/ PO17	SCK5/SSLA2-B/ ET0_WOL		IRQ11	
97		PA0	A0/BC0#	MTIOC4A/MTIOC6D/ GTIOC0B-C/TIOCA0/ CACREF/PO16	SSLA1-B/ ET0_TX_EN/ RMII0_TXD_EN			
98		P67	CS7#/DQM1	MTIOC7C/ GTIOC1B-C	CRX2		IRQ15	
99		P66	CS6#/DQM0	MTIOC7D/ GTIOC2B-C	CTX2			
100		P65	CS5#/CKE					
101		PE7	D15[A15/D15]	MTIOC6A/ GTIOC3A-E/TOC1		MMC_RES#-B/ SDHI_WP-B	IRQ7	AN105
102		PE6	D14[A14/D14]	MTIOC6C/GTIOC3B-E/TIC1		MMC_CD-B/ SDHI_CD-B	IRQ6	AN104
103	VCC							
104		P70	SDCLK					
105	VSS							
106		PE5	D13[A13/D13]	MTIOC4C/MTIOC2B/ GTIOC0A-A	ET0_RX_CLK/ REF50CK0		IRQ5	AN103
107		PE4	D12[A12/D12]	MTIOC4D/MTIOC1A/ GTIOC1A-A/PO28	ET0_ERXD2			AN102
108		PE3	D11[A11/D11]	MTIOC4B/ GTIOC2A-A/PO26/ POE8#/TOC3	CTS12#/RTS12#/ SS12#/ET0_ERXD3/	MMC_D7-B		AN101
109		PE2	D10[A10/D10]	MTIOC4A/ GTIOC0B-A/PO23/ TIC3	RXD12/SMISO12/ SSCL12/RXDX12/	MMC_D6-B	IRQ7-DS	AN100
110		PE1	D9[A9/D9]	MTIOC4C/MTIOC3B/ GTIOC1B-A/PO18	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12	MMC_D5-B		ANEX1
111		PE0	D8[A8/D8]	MTIOC3D/ GTIOC2B-A	SCK12	MMC_D4-B		ANEX0
112		P64	CS4#/WE#					
113		P63	CS3#/CAS#					
114		P62	CS2#/RAS#					
115		P61	CS1#/SDCS#					
116	VSS							

Table 1.8 List of Pin and Pin Functions (144-Pin LFQFP) (5/5)

Pin Number 144-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
117		P60	CS0#					
118	VCC							
119		PD7	D7[A7/D7]	MTIC5U/POE0#		MMC_D1-B/ SDHI_D1-B/ QIO1-B/QMI-B	IRQ7	AN107
120		PD6	D6[A6/D6]	MTIC5V/MTIOC8A/ POE4#		MMC_D0-B/ SDHI_D0-B/ QIO0-B/QMO-B	IRQ6	AN106
121		PD5	D5[A5/D5]	MTIC5W/MTIOC8C/ POE10#		MMC_CLK-B/ SDHI_CLK-B/ QSPCLK-B	IRQ5	AN113
122		PD4	D4[A4/D4]	MTIOC8B/POE11#		MMC_CMD-B/ SDHI_CMD-B/ QSSL-B	IRQ4	AN112
123		PD3	D3[A3/D3]	MTIOC8D/ GTIOC0A-E/POE8#/ TOC2		MMC_D3-B/ SDHI_D3-B/ QIO3-B	IRQ3	AN111
124		PD2	D2[A2/D2]	MTIOC4D/ GTIOC0B-E/TIC2	CRX0	MMC_D2-B/ SDHI_D2-B/ QIO2-B	IRQ2	AN110
125		PD1	D1[A1/D1]	MTIOC4B/ GTIOC1A-E/POE0#	CTX0		IRQ1	AN109
126		PD0	D0[A0/D0]	GTIOC1B-E/POE4#			IRQ0	AN108
127		P93	A19	POE0#	CTS7#/RTS7#/SS7#			AN117
128		P92	A18	POE4#	RXD7/SMISO7/SSCL7			AN116
129		P91	A17		SCK7			AN115
130	VSS							
131		P90	A16		TXD7/SMOSI7/SSDA7			AN114
132	VCC							
133		P47					IRQ15-DS	AN007
134		P46					IRQ14-DS	AN006
135		P45					IRQ13-DS	AN005
136		P44					IRQ12-DS	AN004
137		P43					IRQ11-DS	AN003
138		P42					IRQ10-DS	AN002
139		P41					IRQ9-DS	AN001
140	VREFL0							
141		P40					IRQ8-DS	AN000
142	VREFH0							
143	AVCC0							
144		P07					IRQ15	ADTRG0#

Note 1. The BCLK function is multiplexed with the I/O port function for pin P53, so the port function is not available if the external bus is enabled.

Table 1.9 List of Pin and Pin Functions (100-Pin TFLGA) (1/4)

Pin Number 100-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
A1	P05						IRQ13	DA1
A2	AVCC1							
A3		P07					IRQ15	ADTRG0#
A4	VREFL0							
A5		P43					IRQ11-DS	AN003
A6		PD0	D0[A0/D0]	GTIOC1B-E/POE4#			IRQ0	AN108
A7		PD4	D4[A4/D4]	MTIOC8B/POE11#		MMC_CMD-B/ SDHI_CMD-B/ QSSL-B	IRQ4	AN112
A8		PE0	D8[A8/D8]	MTIOC3D/GTIOC2B-A	SCK12	MMC_D4-B		ANEX0
A9		PE1	D9[A9/D9]	MTIOC4C/MTIOC3B/ GTIOC1B-A/PO18	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12	MMC_D5-B		ANEX1
A10		PE2	D10[A10/ D10]	MTIOC4A/GTIOC0B-A/PO23/TIC3	RXD12/SMISO12/ SSCL12/RXDX12	MMC_D6-B	IRQ7-DS	AN100
B1	EMLE							
B2	AVSS0							
B3	AVCC0							
B4		P40					IRQ8-DS	AN000
B5		P44					IRQ12-DS	AN004
B6		PD1	D1[A1/D1]	MTIOC4B/GTIOC1A-E/POE0#	CTX0		IRQ1	AN109
B7		PD3	D3[A3/D3]	MTIOC8D/GTIOC0A-E/POE8#/TOC2		MMC_D3-B/ SDHI_D3-B/ QIO3-B	IRQ3	AN111
B8		PD6	D6[A6/D6]	MTIC5V/MTIOC8A/ POE4#		MMC_D0-B/ SDHI_D0-B/ QIO0-B/QMO-B	IRQ6	AN106
B9		PD7	D7[A7/D7]	MTIC5U/POE0#		MMC_D1-B/ SDHI_D1-B/ QIO1/QMI-B	IRQ7	AN107
B10		PE3	D11[A11/ D11]	MTIOC4B/GTIOC2A-A/PO26/POE8#/TOC3	CTS12#/RTS12#/ SS12#/ET0_ERXD3	MMC_D7-B		AN101
C1	VCL							
C2	AVSS1							
C3		PJ3	EDACK1	MTIOC3C	ET0_EXOUT/CTS6#/ RTS6#/CTS0#/RTS0#/ SS6#/SS0#			
C4	VREFH0							
C5		P42					IRQ10-DS	AN002
C6		P47					IRQ15-DS	AN007
C7		PD2	D2[A2/D2]	MTIOC4D/GTIOC0B-E/TIC2	CRX0	MMC_D2-B/ SDHI_D2-B/ QIO2-B	IRQ2	AN110

Table 1.9 List of Pin and Pin Functions (100-Pin TFLGA) (2/4)

Pin Number 100-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
C8		PD5	D5[A5/D5]	MTIC5W/MTIOC8C/ POE10#		MMC_CLK-B/ SDHI_CLK-B/ QSPCLK-B	IRQ5	AN113
C9		PE5	D13[A13/ D13]	MTIOC4C/MTIOC2B/ GTIOC0A-A	ET0_RX_CLK/ REF50CK0		IRQ5	AN103
C10		PE4	D12[A12/ D12]	MTIOC4D/MTIOC1A/ GTIOC1A-A/PO28	ET0_ERXD2			AN102
D1	XCIN							
D2	XCOUT							
D3	MD/FINED							
D4	VBATT							
D5		P45					IRQ13-DS	AN005
D6		P46					IRQ14-DS	AN006
D7		PE6	D14[A14/ D14]	MTIOC6C/GTIOC3B-E/TIC1		MMC_CD-B/ SDHI_CD-B	IRQ6	AN104
D8		PE7	D15[A15/ D15]	MTIOC6A/GTIOC3A-E/TOC1		MMC_RES#-B/ SDHI_WP-B	IRQ7	AN105
D9		PA1	A1	MTIOC0B/MTCLKC/ MTIOC7B/GTIOC2A-C/TIOCB0/PO17	SCK5/SSLA2-B/ ET0_WOL		IRQ11	
D10		PA0	A0/BC0#	MTIOC4A/MTIOC6D/ GTIOC0B-C/TIOCA0/ CACREF/PO16	SSLA1-B/ ET0_TX_EN/ RMII0_TXD_EN			
E1	XTAL	P37						
E2	VSS							
E3	RES#							
E4	TRST#	P34		MTIOC0A/TMCI3/ PO12/POE10#	SCK6/SCK0/ ET0_LINKSTA		IRQ4	
E5		P41					IRQ9-DS	AN001
E6		PA2	A2	MTIOC7A/GTIOC1A-C/PO18	RXD5/SMISO5/ SSCL5/SSLA3-B			
E7		PA6	A6	MTIC5V/MTCLKB/ GTETRG-C/TIOCA2/ TMCI3/PO22/POE10#	CTS5#/RTS5#/SS5#/ MOSIA-B/ ET0_EXOUT			
E8		PA4	A4	MTIC5U/MTCLKA/ TIOCA1/TMRI0/PO20	TXD5/SMOSI5/ SSDA5/SSLA0-B/ ET0_MDC		IRQ5-DS	
E9		PA5	A5	MTIOC6B/TIOCB1/ GTIOC0A-C/PO21	RSPCKA-B/ ET0_LINKSTA			
E10		PA3	A3	MTIOC0D/MTCLKD/ TIOCD0/TCLKB/PO19	RXD5/SMISO5/ SSCL5/ET0_MDIO		IRQ6-DS	
F1	EXTAL	P36						
F2	VCC							
F3	UPSEL	P35					NMI	
F4		P32		MTIOC0C/TIOCC0/ TMO3/PO10/ RTCOUT/RTCIC2/ POE0#/POE10#	TXD6/TXD0/SMOSI6/ SMOSI0/SSDA6/ SSDA0/CTX0/ USB0_VBUSEN		IRQ2-DS	
F5		P12		TMCI1	RXD2/SMISO2/ SSCL2/SCL0[FM+]		IRQ2	

Table 1.9 List of Pin and Pin Functions (100-Pin TFLGA) (3/4)

Pin Number 100-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
F6		PB3	A11	MTIOC0A/MTIOC4A/ TIOCD3/TCLKD/ TMO0/PO27/POE11#	SCK6/ET0_RX_ER/ RMII0_RX_ER			
F7		PB2	A10	TIOCC3/TCLKC/ PO26	CTS6#/RTS6#SS6#/ ET0_RX_CLK/ REF50CK0			
F8		PB0	A8	MTIC5W/TIOCA3/ PO24	RXD6/SMISO6/ SSCL6/ET0_ERXD1/ RMII0_RXD1		IRQ12	
F9		PA7	A7	TIOCB2/PO23	MISOA-B/ET0_WOL			
F10	VSS							
G1		P33	EDREQ1	MTIOC0D/TIOCD0/ TMRI3/PO11/POE4#/ POE11#	RXD6/RXD0/SMISO6/ SMISO0/SSCL6/ SSCL0/CRX0		IRQ3-DS	
G2	TMS	P31		MTIOC4D/TMCI2/ PO9/RTCIC1	CTS1#/RTS1#/SS1#		IRQ1-DS	
G3	TDI	P30		MTIOC4B/TMRI3/ PO8/RTCIC0/POE8#	RXD1/SMISO1/SSCL1		IRQ0-DS	
G4	TCK	P27	CS7#	MTIOC2B/TMCI3/PO7	SCK1			
G5		P53*1	BCLK					
G6		P52	RD#		RXD2/SMISO2/SSCL2			
G7		PB5	A13	MTIOC2A/MTIOC1B/ TIOCB4/TMRI1/PO29/ POE4#	SCK9/RTS9#/ ET0_ETXD0/ RMII0_TXD0			
G8		PB4	A12	TIOCA4/PO28	CTS9#/ET0_TX_EN/ RMII0_TXD_EN			
G9		PB1	A9	MTIOC0C/MTIOC4C/ TIOCB3/TMCI0/PO25	TXD6/SMOSI6/ SSDA6/ET0_ERXD0/ RMII0_RXD0		IRQ4-DS	
G10	VCC							
H1	TDO	P26	CS6#	MTIOC2A/TMO1/PO6	TXD1/CTS3#/RTS3#/ SMOSI1/SS3#/SSDA1			
H2		P25	CS5#/ EDACK1	MTIOC4C/MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/ SSCL3/SSIDATA1			ADTRG0#
H3		P16		MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/ TMO2/PO14/ RTCOUT	TXD1/RXD3/SMOSI1/ SMISO3/SSDA1/ SSCL3/SCL2-DS/ USB0_VBUS/ USB0_VBUSEN/ USB0_OVRCURB		IRQ6	ADTRG0#
H4		P15		MTIOC0B/MTCLKB/ GTETRG-B/TIOCB2/ TCLKB/TMCI2/PO13	RXD1/SCK3/SMISO1/ SSCL1/CRX1-DS/ SSIWS1		IRQ5	
H5		P55	WAIT#/ EDREQ0	MTIOC4D/TMO3	CRX1/ET0_EXOUT		IRQ10	
H6		P54	ALE/ EDACK0	MTIOC4B/TMCI1	CTS2#/RTS2#/SS2#/ CTX1/ET0_LINKSTA			
H7	UB	PC7	A23/CS0#	MTIOC3A/MTCLKB/ GTIOC3A-D/TMO2/ TOC0/PO31/CACREF	TXD8/MISOA-A/ ET0_COL		IRQ14	
H8		PC6	A22/CS1#	MTIOC3C/MTCLKA/ GTIOC3B-D/TMCI2/ TIC0/PO30	RXD8/MOSIA-A/ ET0_ETXD3		IRQ13	

Table 1.9 List of Pin and Pin Functions (100-Pin TFLGA) (4/4)

Pin Number 100-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
H9		PB6	A14	MTIOC3D/TIOCA5/ PO30	RXD9/ET0_ETXD1/ RMII0_TXD1			
H10		PB7	A15	MTIOC3B/TIOCB5/ PO31	TXD9/ET0_CRS/ RMII0_CRS_DV			
J1		P24	CS4#/ EDREQ1	MTIOC4A/MTCLKA/ TIOCB4/TMRI1/PO4	SCK3/ USB0_VBUSEN/ SSISCK1			
J2		P21		MTIOC1B/MTIOC4A/ GTIOC2A-B/TIOCA3/ TMCI0/PO1	RXD0/SMISO0/ SSCL0/ USB0_EXICEN/ SSIWS0		IRQ9	
J3		P17		MTIOC3A/MTIOC3B/ MTIOC4B/GTIOC0B-B/TIOCB0/TCLKD/ TMO1/PO15/POE8#	SCK1/TXD3/SMOSI3/ SSDA3/SDA2-DS/ SSITXD0		IRQ7	ADTRG1#
J4		P13		MTIOC0B/TIOCA5/ TMO3/PO13	TXD2/SMOSI2/ SSDA2/SDA0[FM+]		IRQ3	ADTRG1#
J5	VSS_USB							
J6	VCC_USB							
J7		P50	WR0#/WR#		TXD2/SMOSI2/SSDA2			
J8		PC4	A20/CS3#	MTIOC3D/MTCLKC/ GTETRG-D/TMCI1/ PO25/POE0#	SCK5/CTS8#/SSLA0-A/ET0_TX_CLK			
J9		PC0	A16	MTIOC3C/TCLKC/ PO17	CTS5#/RTS5#/SS5#/ SSLA1-A/ET0_ERXD3		IRQ14	
J10		PC1	A17	MTIOC3A/TCLKD/ PO18	SCK5/SSLA2-A/ ET0_ERXD2		IRQ12	
K1		P23	EDACK0	MTIOC3D/MTCLKD/ GTIOC0A-B/TIOCD3/ PO3	TXD3/CTS0#/RTS0#/ SMOSI3/SS0#/ SSDA3/SSISCK0			
K2		P22	EDREQ0	MTIOC3B/MTCLKC/ GTIOC1A-B/TIOCC3/ TMO0/PO2	SCK0/ USB0_OVRCURB/ AUDIO_MCLK			
K3		P20		MTIOC1A/TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/ SSDA0/USB0_ID/ SSIRXD0		IRQ8	
K4		P14		MTIOC3A/MTCLKA/ TIOCB5/TCLKA/ TMRI2/PO15	CTS1#/RTS1#/SS1#/ CTX1/ USB0_OVRCURA		IRQ4	
K5					USB0_DM			
K6					USB0_DP			
K7		P51	WR1#/BC1#/ WAIT#		SCK2			
K8		PC5	A21/CS2#/ WAIT#	MTIOC3B/MTCLKD/ GTIOC1A-D/TMRI2/ PO29	SCK8/RSPCKA-A/ RTS8#/ET0_ETXD2			
K9		PC3	A19	MTIOC4D/GTIOC1B-D/TCLKB/PO24	TXD5/SMOSI5/ SSDA5/ET0_TX_ER			
K10		PC2	A18	MTIOC4B/GTIOC2B-D/TCLKA/PO21	RXD5/SMISO5/ SSCL5/SSLA3-A/ ET0_RX_DV			

Note 1. The BCLK function is multiplexed with the I/O port function for pin P53, so the port function is not available if the external bus is enabled.

Table 1.10 List of Pin and Pin Functions (100-Pin LFQFP) (1/4)

Pin Number 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
1	AVCC1							
2	EMLE							
3	AVSS1							
4		PJ3	EDACK1	MTIOC3C	ET0_EXOUT CTS6#/RTS6#/CTS0#/ RTS0#/SS6#/SS0#			
5	VCL							
6	VBATT							
7	MD/FINED							
8	XCIN							
9	XCOUT							
10	RES#							
11	XTAL	P37						
12	VSS							
13	EXTAL	P36						
14	VCC							
15	UPSEL	P35					NMI	
16	TRST#	P34		MTIOC0A/TMCI3/ PO12/POE10#	SCK6/SCK0/ ET0_LINKSTA		IRQ4	
17		P33	EDREQ1	MTIOC0D/TIOCD0/ TMRI3/PO11/POE4#/ POE11#	RXD6/RXD0/SMISO6/ SMISO0/SSCL6/ SSCL0/CRX0		IRQ3-DS	
18		P32		MTIOC0C/TIOCC0/ TMO3/PO10/ RTCOUT/RTCIC2/ POE0#/POE10#	TXD6/TXD0/SMOSI6/ SMOSI0/SSDA6/ SSDA0/CTX0/ USB0_VBUSEN		IRQ2-DS	
19	TMS	P31		MTIOC4D/TMCI2/ PO9/RTCIC1	CTS1#/RTS1#/SS1#		IRQ1-DS	
20	TDI	P30		MTIOC4B/TMRI3/ PO8/RTCIC0/POE8#	RXD1/SMISO1/SSCL1		IRQ0-DS	
21	TCK	P27	CS7#	MTIOC2B/TMCI3/PO7	SCK1			
22	TDO	P26	CS6#	MTIOC2A/TMO1/PO6	TXD1/CTS3#/RTS3#/ SMOSI1/SS3#/SSDA1			
23		P25	CS5#/ EDACK1	MTIOC4C/MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/ SSCL3/SSIDATA1			ADTRG0#
24		P24	CS4#/ EDREQ1	MTIOC4A/MTCLKA/ TIOCB4/TMRI1/PO4	SCK3/ USB0_VBUSEN/ SSISCK1			
25		P23	EDACK0	MTIOC3D/MTCLKD/ GTIOC0A-B/TIOCD3/ PO3	TXD3/CTS0#/RTS0#/ SMOSI3/SS0#/ SSDA3/SSISCK0			
26		P22	EDREQ0	MTIOC3B/MTCLKC/ GTIOC1A-B/TIOCC3/ TMO0/PO2	SCK0/ USB0_OVRCURB/ AUDIO_MCLK			
27		P21		MTIOC1B/MTIOC4A/ GTIOC2A-B/TIOCA3/ TMCI0/PO1	RXD0/SMISO0/ SSCL0/ USB0_EXICEN/ SSIWS0		IRQ9	
28		P20		MTIOC1A/TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/ SSDA0/USB0_ID/ SSIRXD0		IRQ8	
29		P17		MTIOC3A/MTIOC3B/ MTIOC4B/ GTIOC0B-B/TIOCB0/ TCLKD/TMO1/PO15/ POE8#	SCK1/TXD3/SMOSI3/ SSDA3/SDA2-DS/ SSITXD0		IRQ7	ADTRG1#

Table 1.10 List of Pin and Pin Functions (100-Pin LFQFP) (2/4)

Pin Number 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
30		P16		MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/ TMO2/PO14/ RTCOUT	TXD1/RXD3/SMOSI1/ SMISO3/SSDA1/ SSCL3/SCL2-DS/ USB0_VBUS/ USB0_VBUSEN/ USB0_OVRCURB		IRQ6	ADTRG0#
31		P15		MTIOC0B/MTCLKB/ GTETRG-B/TIOCB2/ TCLKB/TMCI2/PO13	RXD1/SCK3/SMISO1/ SSCL1/CRX1-DS/ SSIWS1		IRQ5	
32		P14		MTIOC3A/MTCLKA/ TIOCB5/TCLKA/ TMRI2/PO15	CTS1#/RTS1#/SS1#/ CTX1/ USB0_OVRCURA		IRQ4	
33		P13		MTIOC0B/TIOCA5/ TMO3/PO13	TXD2/SMOSI2/ SSDA2/SDA0[FM+]		IRQ3	ADTRG1#
34		P12		TMCI1	RXD2/SMISO2/ SSCL2/SCL0[FM+]		IRQ2	
35	VCC_USB							
36					USB0_DM			
37					USB0_DP			
38	VSS_USB							
39		P55	WAIT#/ EDREQ0	MTIOC4D/TMO3	CRX1/ET0_EXOUT		IRQ10	
40		P54	ALE/EDACK0	MTIOC4B/TMCI1	CTS2#/RTS2#/SS2#/ CTX1/ET0_LINKSTA			
41		P53*1	BCLK					
42		P52	RD#		RXD2/SMISO2/SSCL2			
43		P51	WR1#/BC1#/ WAIT#		SCK2			
44		P50	WR0#/WR#		TXD2/SMOSI2/SSDA2			
45	UB	PC7	A23/CS0#	MTIOC3A/MTCLKB/ GTIOC3A-D/TMO2/ TOC0/PO31/CACREF	TXD8/MISOA-A/ ET0_COL		IRQ14	
46		PC6	A22/CS1#	MTIOC3C/MTCLKA/ GTIOC3B-D/TMCI2/ TIC0/PO30	RXD8/MOSIA-A/ ET0_ETXD3		IRQ13	
47		PC5	A21/CS2#/ WAIT#	MTIOC3B/MTCLKD/ GTIOC1A-D/TMRI2/ PO29	SCK8/RSPCKA-A/ RTS8#/ET0_ETXD2			
48		PC4	A20/CS3#	MTIOC3D/MTCLKC/ GTETRG-D/TMCI1/ PO25/POE0#	SCK5/CTS8#/ SSLA0-A/ ET0_TX_CLK			
49		PC3	A19	MTIOC4D/ GTIOC1B-D/TCLKB/ PO24	TXD5/SMOSI5/ SSDA5/ET0_TX_ER			
50		PC2	A18	MTIOC4B/ GTIOC2B-D/TCLKA/ PO21	RXD5/SMISO5/ SSCL5/SSLA3-A/ ET0_RX_DV			
51		PC1	A17	MTIOC3A/TCLKD/ PO18	SCK5/SSLA2-A/ ET0_ERXD2		IRQ12	
52		PC0	A16	MTIOC3C/TCLKC/ PO17	CTS5#/RTS5#/SS5#/ SSLA1-A/ET0_ERXD3		IRQ14	
53		PB7	A15	MTIOC3B/TIOCB5/ PO31	TXD9/ET0_CRS/ RMII0_CRS_DV			
54		PB6	A14	MTIOC3D/TIOCA5/ PO30	RXD9/ET0_ETXD1/ RMII0_TXD1			
55		PB5	A13	MTIOC2A/MTIOC1B/ TIOCB4/TMRI1/PO29/ POE4#	SCK9/RTS9#/ ET0_ETXD0/ RMII0_TXD0			
56		PB4	A12	TIOCA4/PO28	CTS9#/ET0_TX_EN/ RMII0_TXD_EN			

Table 1.10 List of Pin and Pin Functions (100-Pin LFQFP) (3/4)

Pin Number 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
57		PB3	A11	MTIOC0A/MTIOC4A/ TIOCD3/TCLKD/ TMO0/PO27/POE11#	SCK6/ET0_RX_ER/ RMII0_RX_ER			
58		PB2	A10	TIOCC3/TCLKC/ PO26	CTS6#/RTS6#SS6#/ ET0_RX_CLK/ REF50CK0			
59		PB1	A9	MTIOC0C/MTIOC4C/ TIOCB3/TMCI0/PO25	TXD6/SMOSI6/ SSDA6/ET0_ERXD0/ RMII0_RXD0		IRQ4-DS	
60	VCC							
61		PB0	A8	MTIC5W/TIOCA3/ PO24	RXD6/SMISO6/ SSCL6/ET0_ERXD1/ RMII0_RXD1		IRQ12	
62	VSS							
63		PA7	A7	TIOCB2/PO23	MISOA-B/ET0_WOL			
64		PA6	A6	MTIC5V/MTCLKB/ GTETRG-C/TIOCA2/ TMCI3/PO22/POE10#	CTS5#/RTS5#/SS5#/ MOSIA-B/ ET0_EXOUT			
65		PA5	A5	MTIOC6B/TIOCB1/ GTIOC0A-C/PO21	RSPCKA-B/ ET0_LINKSTA			
66		PA4	A4	MTIC5U/MTCLKA/ TIOCA1/TMRI0/PO20	TXD5/SMOSI5/ SSDA5/SSLA0-B/ ET0_MDC		IRQ5-DS	
67		PA3	A3	MTIOC0D/MTCLKD/ TIOCD0/TCLKB/PO19	RXD5/SMISO5/ SSCL5/ET0_MDIO		IRQ6-DS	
68		PA2	A2	MTIOC7A/ GTIOC1A-C/PO18	RXD5/SMISO5/ SSCL5/SSLA3-B			
69		PA1	A1	MTIOC0B/MTCLKC/ MTIOC7B/ GTIOC2A-C/TIOCB0/ PO17	SCK5/SSLA2-B/ ET0_WOL		IRQ11	
70		PA0	A0/BC0#	MTIOC4A/MTIOC6D/ GTIOC0B-C/TIOCA0/ CACREF/PO16	SSLA1-B/ ET0_TX_EN/ RMII0_TXD_EN			
71		PE7	D15[A15/D15]	MTIOC6A/ GTIOC3A-E/TOC1		MMC_RES#-B/ SDHI_WP-B	IRQ7	AN105
72		PE6	D14[A14/D14]	MTIOC6C/GTIOC3B-E/TIC1		MMC_CD-B/ SDHI_CD-B	IRQ6	AN104
73		PE5	D13[A13/D13]	MTIOC4C/MTIOC2B/ GTIOC0A-A	ET0_RX_CLK/ REF50CK0		IRQ5	AN103
74		PE4	D12[A12/D12]	MTIOC4D/MTIOC1A/ GTIOC1A-A/PO28	ET0_ERXD2			AN102
75		PE3	D11[A11/D11]	MTIOC4B/ GTIOC2A-A/PO26/ POE8#/TOC3	CTS12#/RTS12#/ SS12#/ET0_ERXD3	MMC_D7-B		AN101
76		PE2	D10[A10/D10]	MTIOC4A/ GTIOC0B-A/PO23/ TIC3	RXD12/SMISO12/ SSCL12/RXDX12	MMC_D6-B	IRQ7-DS	AN100
77		PE1	D9[A9/D9]	MTIOC4C/MTIOC3B/ GTIOC1B-A/PO18	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12	MMC_D5-B		ANEX1
78		PE0	D8[A8/D8]	MTIOC3D/ GTIOC2B-A	SCK12	MMC_D4-B		ANEX0
79		PD7	D7[A7/D7]	MTIC5U/POE0#		MMC_D1-B/ SDHI_D1-B/ QIO1-B/ QMI-B	IRQ7	AN107
80		PD6	D6[A6/D6]	MTIC5V/MTIOC8A/ POE4#		MMC_D0-B/ SDHI_D0-B/ QIO0-B/ QMO-B	IRQ6	AN106

Table 1.10 List of Pin and Pin Functions (100-Pin LFQFP) (4/4)

Pin Number 100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timer (MTU, GPT, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	Communication (ETHERC, SCIg, SCIh, RSPI, RIIC, CAN, USB, SSI)	Memory Interface Camera Interface (QSPI, SDHI, MMCIF, PDC)	Interrupt	S12ADC, R12DA
81		PD5	D5[A5/D5]	MTIC5W/MTIOC8C/ POE10#		MMC_CLK-B/ SDHI_CLK-B/ QSPCLK-B	IRQ5	AN113
82		PD4	D4[A4/D4]	MTIOC8B/POE11#		MMC_CMD-B/ SDHI_CMD-B/ QSSL-B	IRQ4	AN112
83		PD3	D3[A3/D3]	MTIOC8D/ GTIOC0A-E/POE8#/ TOC2		MMC_D3-B/ SDHI_D3-B/ QIO3-B	IRQ3	AN111
84		PD2	D2[A2/D2]	MTIOC4D/ GTIOC0B-E/TIC2	CRX0	MMC_D2-B/ SDHI_D2-B/ QIO2-B	IRQ2	AN110
85		PD1	D1[A1/D1]	MTIOC4B/ GTIOC1A-E/POE0#	CTX0		IRQ1	AN109
86		PD0	D0[A0/D0]	GTIOC1B-E/POE4#			IRQ0	AN108
87		P47					IRQ15-DS	AN007
88		P46					IRQ14-DS	AN006
89		P45					IRQ13-DS	AN005
90		P44					IRQ12-DS	AN004
91		P43					IRQ11-DS	AN003
92		P42					IRQ10-DS	AN002
93		P41					IRQ9-DS	AN001
94	VREFL0							
95		P40					IRQ8-DS	AN000
96	VREFH0							
97	AVCC0							
98		P07					IRQ15	ADTRG0#
99	AVSS0							
100	P05						IRQ13	DA1

Note 1. The BCLK function is multiplexed with the I/O port function for pin P53, so the port function is not available if the external bus is enabled.

2. CPU

The RXv2 instruction set architecture (RXv2) has upward compatibility with the RXv1 instruction set architecture (RXv1).

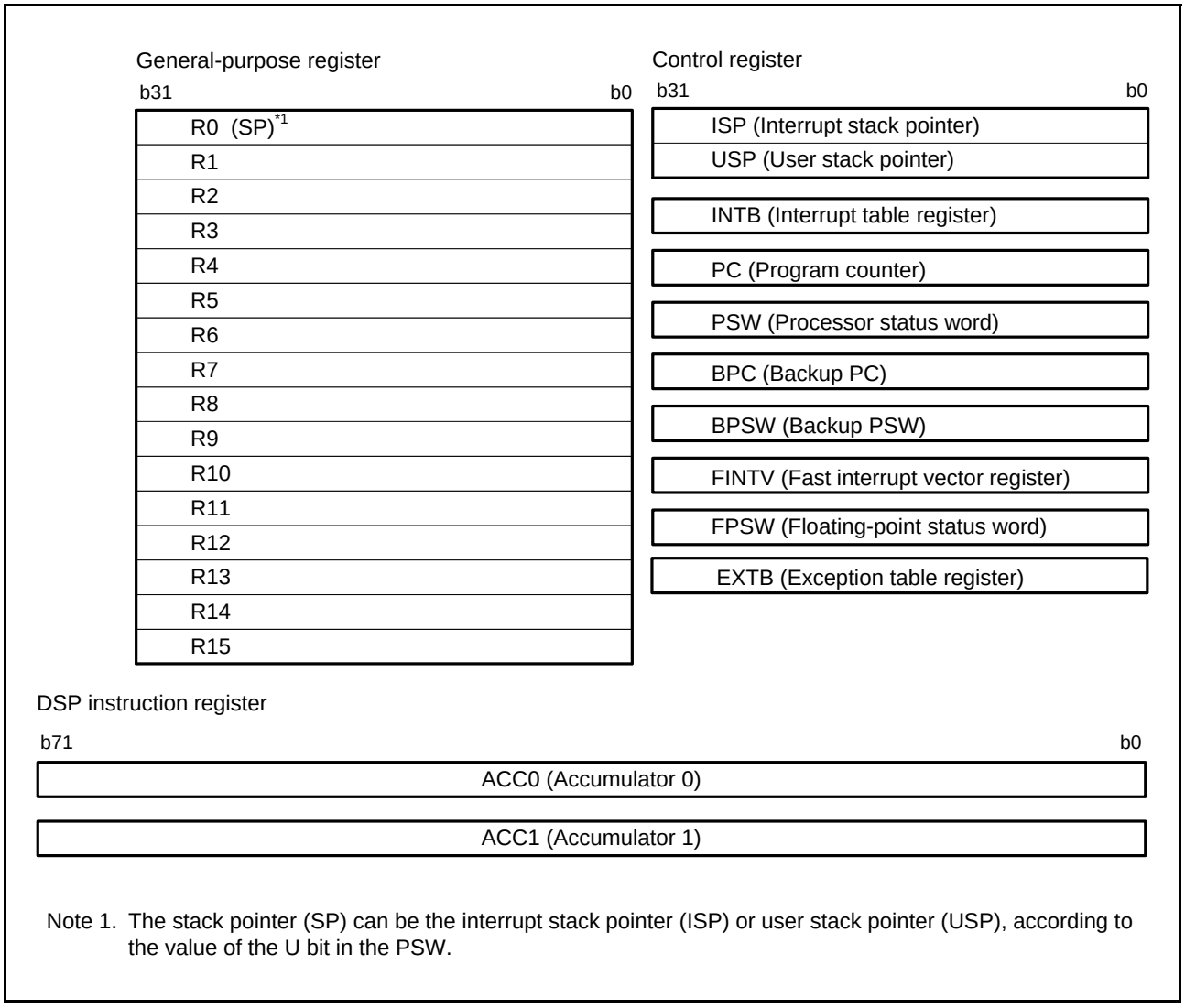
- Adoption of variable-length instruction format
  As with RXv1, the RXv2 CPU has short formats for frequently used instructions, facilitating the development of efficient programs that take up less memory.
- Powerful instruction set
  The RXv2 supports 109 selected instructions. Moreover, DSP instructions and floating-point operation instructions are added, thus realizing high-speed arithmetic processing.
- Versatile addressing modes
  The RXv2 CPU has 11 versatile addressing modes, with register-register operations, register-memory operations, and bitwise operations included. Data transfer between memory locations is also possible.

2.1 Features

- Minimum instruction execution rate: One clock cycle
- Address space: 4-Gbyte linear addresses
- Register set of the CPU
  General purpose: Sixteen 32-bit registers
  Control: Ten 32-bit registers
  Accumulator: Two 72-bit registers
- Variable-length instruction format (lengths from one to eight bytes)
- 109 instructions/11 addressing modes
  Basic instructions: 75
  Floating-point operation instructions: 11
  DSP instructions: 23
- Processor modes
  Supervisor mode and user mode
- Vector tables
  Exception vector table and interrupt vector table
- Memory protection unit
- Data arrangement
  Selectable as little endian or big endian

2.2 Register Set of the CPU

The RXv2 CPU has sixteen general-purpose registers, ten control registers, and two accumulator used for DSP instructions.



Note 1. The stack pointer (SP) can be the interrupt stack pointer (ISP) or user stack pointer (USP), according to the value of the U bit in the PSW.

Figure 2.1 Register Set of the CPU

2.2.1 General-Purpose Registers (R0 to R15)

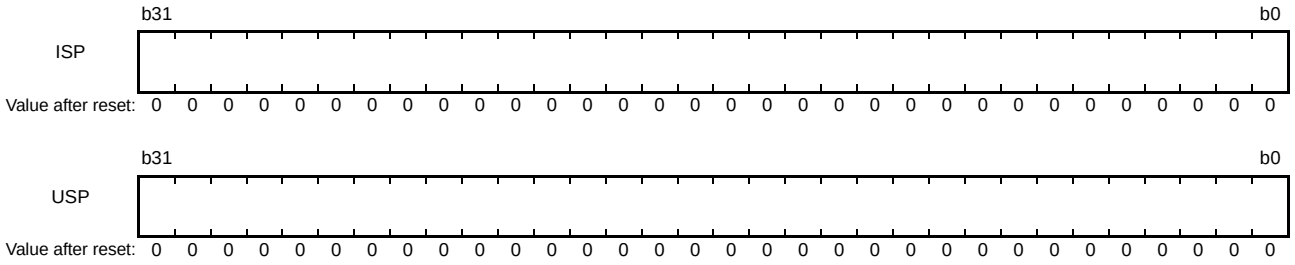
This CPU has sixteen 32-bit general-purpose registers (R0 to R15). R0 to R15 can be used as data registers or address registers.
R0, a general-purpose register, also functions as the stack pointer (SP).
The stack pointer is switched to operate as the interrupt stack pointer (ISP) or user stack pointer (USP) by the value of the stack pointer select bit (U) in the processor status word (PSW).

2.2.2 Control Registers

This CPU has the following ten control registers.

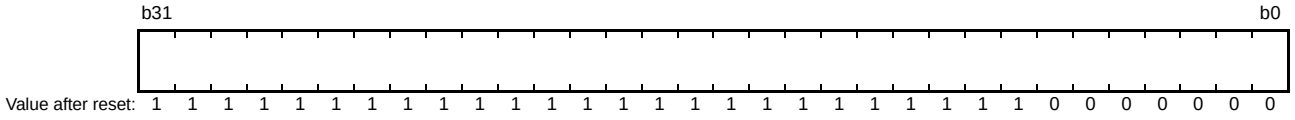
- Interrupt stack pointer (ISP)
- User stack pointer (USP)
- Exception table register (EXTB)
- Interrupt table register (INTB)
- Program counter (PC)
- Processor status word (PSW)
- Backup PC (BPC)
- Backup PSW (BPSW)
- Fast interrupt vector register (FINTV)
- Floating-point status word (FPSW)

2.2.2.1 Interrupt Stack Pointer (ISP)/User Stack Pointer (USP)



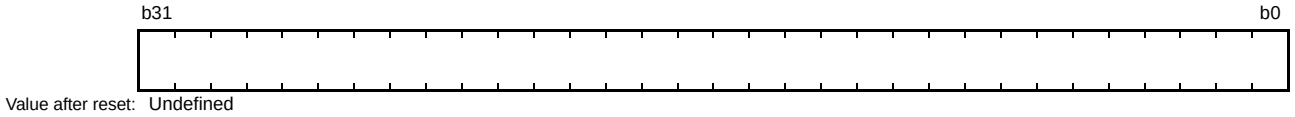
The stack pointer (SP) can be either of two types, the interrupt stack pointer (ISP) or the user stack pointer (USP). Whether the stack pointer operates as the ISP or USP depends on the value of the stack pointer select bit (U) in the processor status word (PSW).

2.2.2.2 Exception Table Register (EXTB)



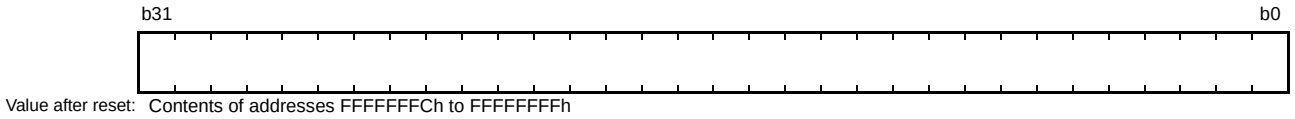
The exception table register (EXTB) specifies the address where the exception vector table starts.

2.2.2.3 Interrupt Table Register (INTB)



The interrupt table register (INTB) specifies the address where the interrupt vector table starts.

2.2.2.4 Program Counter (PC)



The program counter (PC) indicates the address of the instruction being executed.

2.2.2.5 Processor Status Word (PSW)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	IPL[3:0]				—	—	—	PM	—	—	U	I
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	O	S	Z	C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	C	Carry Flag	0: No carry has occurred. 1: A carry has occurred.	R/W
b1	Z	Zero Flag	0: Result is non-zero. 1: Result is 0.	R/W
b2	S	Sign Flag	0: Result is a positive value or 0. 1: Result is a negative value.	R/W
b3	O	Overflow Flag	0: No overflow has occurred. 1: An overflow has occurred.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	I*1	Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/W
b17	U*1	Stack Pointer Select	0: Interrupt stack pointer (ISP) is selected. 1: User stack pointer (USP) is selected.	R/W
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	PM*1, *2, *3	Processor Mode Select	0: Supervisor mode is selected. 1: User mode is selected.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b24	IPL[3:0]*1	Processor Interrupt Priority Level	b27 b24 0 0 0 0: Priority level 0 (lowest) 0 0 0 1: Priority level 1 0 0 1 0: Priority level 2 0 0 1 1: Priority level 3 0 1 0 0: Priority level 4 0 1 0 1: Priority level 5 0 1 1 0: Priority level 6 0 1 1 1: Priority level 7 1 0 0 0: Priority level 8 1 0 0 1: Priority level 9 1 0 1 0: Priority level 10 1 0 1 1: Priority level 11 1 1 0 0: Priority level 12 1 1 0 1: Priority level 13 1 1 1 0: Priority level 14 1 1 1 1: Priority level 15 (highest)	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. In user mode, writing to the IPL[3:0], PM, U, and I bits by an MVTC or a POPC instruction is ignored. Writing to the IPL[3:0] bits by an MVTIPL instruction generates a privileged instruction exception.
Note 2. In supervisor mode, writing to the PM bit by an MVTC or a POPC instruction is ignored, but writing to the other bits is possible.
Note 3. Switching from supervisor mode to user mode requires execution of an RTE instruction after having set the PSW.PM bit saved on the stack to 1 or executing an RTFI instruction after having set the BPSW.PM bit to 1.

The processor status word (PSW) indicates the results of instruction execution or the state of the CPU.

C Flag (Carry Flag)

This flag retains the state of the bit after a carry, borrow, or shift-out has occurred.

Z Flag (Zero Flag)

This flag is set to 1 if the result of an operation is 0; otherwise its value is set to 0.

S Flag (Sign Flag)

This flag is set to 1 if the result of an operation is negative; otherwise its value is set to 0.

O Flag (Overflow Flag)

This flag is set to 1 if the result of an operation overflows; otherwise its value is set to 0.

I Bit (Interrupt Enable)

This bit enables interrupt requests. When a WAIT instruction is executed, the value of this bit becomes 1. It becomes 0 when an exception is accepted.

U Bit (Stack Pointer Select)

This bit specifies the stack pointer as either the ISP or USP. When an exception request is accepted, this bit is set to 0. When the processor mode is switched from supervisor mode to user mode, this bit is set to 1.

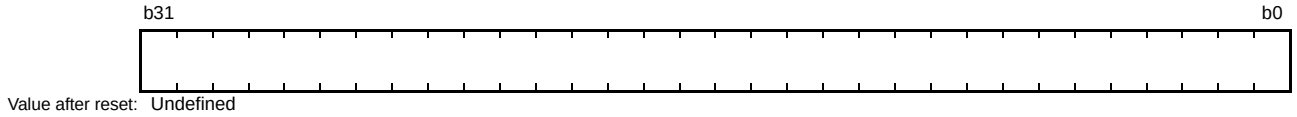
PM Bit (Processor Mode Select)

This bit specifies the processor mode. When an exception is accepted, the value of this bit becomes 0.

IPL[3:0] Bits (Processor Interrupt Priority Level)

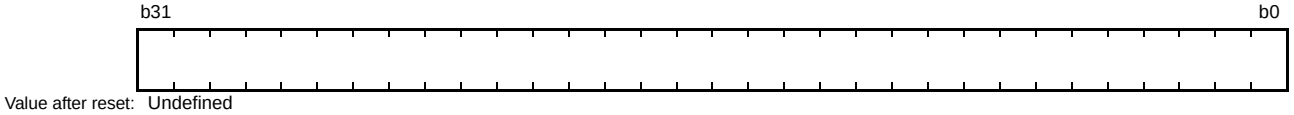
The IPL[3:0] bits specify the processor interrupt priority level as one of sixteen levels from zero to fifteen, wherein priority level zero is the lowest and priority level fifteen the highest. When the priority level of a requested interrupt is higher than the processor interrupt priority level, the interrupt is enabled. Setting the IPL[3:0] bits to level fifteen (Fh) disables all interrupt requests. The IPL[3:0] bits are set to level fifteen (Fh) when a non-maskable interrupt is generated. When interrupts in general are generated, the bits are set to the priority levels of accepted interrupts.

2.2.2.6 Backup PC (BPC)



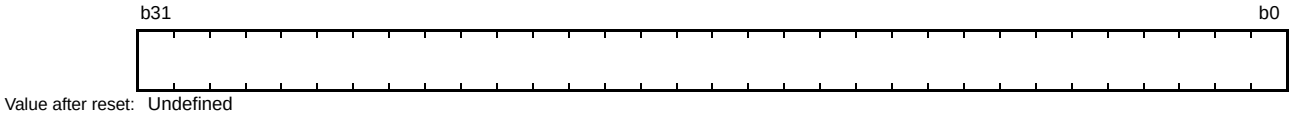
The backup PC (BPC) is provided to speed up response to interrupts.
After a fast interrupt has been generated, the contents of the program counter (PC) are saved in the BPC register.

2.2.2.7 Backup PSW (BPSW)



The backup PSW (BPSW) is provided to speed up response to interrupts.
After a fast interrupt has been generated, the contents of the processor status word (PSW) are saved in the BPSW. The allocation of bits in the BPSW corresponds to that in the PSW.

2.2.2.8 Fast Interrupt Vector Register (FINTV)



The fast interrupt vector register (FINTV) is provided to speed up response to interrupts.
The FINTV register specifies a branch destination address when a fast interrupt has been generated.

2.2.2.9 Floating-Point Status Word (FPSW)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FS	FX	FU	FZ	FO	FV	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	EX	EU	EZ	EO	EV	—	DN	CE	CX	CU	CZ	CO	CV	RM[1:0]	
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	RM[1:0]	Floating-Point Rounding-Mode Setting	b1 b0 0 0: Rounding towards the nearest value 0 1: Rounding towards 0 1 0: Rounding towards +∞ 1 1: Rounding towards –∞	R/W
b2	CV	Invalid Operation Cause Flag	0: No invalid operation has been encountered. 1: Invalid operation has been encountered.	R/(W) *1
b3	CO	Overflow Cause Flag	0: No overflow has occurred. 1: Overflow has occurred.	R/(W) *1
b4	CZ	Division-by-Zero Cause Flag	0: No division-by-zero has occurred. 1: Division-by-zero has occurred.	R/(W) *1
b5	CU	Underflow Cause Flag	0: No underflow has occurred. 1: Underflow has occurred.	R/(W) *1
b6	CX	Inexact Cause Flag	0: No inexact exception has been generated. 1: Inexact exception has been generated.	R/(W) *1
b7	CE	Unimplemented Processing Cause Flag	0: No unimplemented processing has been encountered. 1: Unimplemented process has been encountered.	R/(W) *1
b8	DN	0 Flush Bit of Denormalized Number	0: A denormalized number is handled as a denormalized number. 1: A denormalized number is handled as 0.*2	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	EV	Invalid Operation Exception Enable	0: Invalid operation exception is masked. 1: Invalid operation exception is enabled.	R/W
b11	EO	Overflow Exception Enable	0: Overflow exception is masked. 1: Overflow exception is enabled.	R/W
b12	EZ	Division-by-Zero Exception Enable	0: Division-by-zero exception is masked. 1: Division-by-zero exception is enabled.	R/W
b13	EU	Underflow Exception Enable	0: Underflow exception is masked. 1: Underflow exception is enabled.	R/W
b14	EX	Inexact Exception Enable	0: Inexact exception is masked. 1: Inexact exception is enabled.	R/W
b25 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26	FV*3	Invalid Operation Flag	0: No invalid operation has been encountered. 1: Invalid operation has been encountered.*8	R/W
b27	FO*4	Overflow Flag	0: No overflow has occurred. 1: Overflow has occurred.*8	R/W
b28	FZ*5	Division-by-Zero Flag	0: No division-by-zero has occurred. 1: Division-by-zero has occurred.*8	R/W
b29	FU*6	Underflow Flag	0: No underflow has occurred. 1: Underflow has occurred.*8	R/W
b30	FX*7	Inexact Flag	0: No inexact exception has been generated. 1: Inexact exception has been generated.*8	R/W

Bit	Symbol	Bit Name	Description	R/W
b31	FS	Floating-Point Error Summary Flag	This bit reflects the logical OR of the FU, FZ, FO, and FV flags.	R

Note 1. Writing 0 to the bit clears it. Writing 1 to the bit does not affect its value.
Note 2. Positive denormalized numbers are treated as +0, negative denormalized numbers as –0.
Note 3. When the EV bit is set to 0, the FV flag is enabled.
Note 4. When the EO bit is set to 0, the FO flag is enabled.
Note 5. When the EZ bit is set to 0, the FZ flag is enabled.
Note 6. When the EU bit is set to 0, the FU flag is enabled.
Note 7. When the EX bit is set to 0, the FX flag is enabled.
Note 8. Once the bit has been set to 1, this value is retained until it is set to 0 by software.

The floating-point status word (FPSW) indicates the results of floating-point operations.
When an exception handling enable bit (Ej) enables the exception handling (Ej = 1), the exception cause can be identified by checking the corresponding Cj flag in the exception handling routine. If the exception handling is masked (Ej = 0), the occurrence of exception can be checked by reading the Fj flag at the end of a series of processing. Once the Fj flag has been set to 1, this value is retained until it is set to 0 by software (j = X, U, Z, O, or V).

RM[1:0] Bits (Floating-Point Rounding-Mode Setting)

These bits specify the floating-point rounding-mode.

Explanation of Floating-Point Rounding Modes

- Rounding towards the nearest value (the default behavior) : An inexact result is rounded to the available value that is closest to the result which would be obtained with an infinite number of digits. If two available values are equally close, rounding is to the even alternative.
- Rounding towards 0 : An inexact result is rounded to the smallest available absolute value, i.e. in the direction of zero (simple truncation).
- Rounding towards +∞ : An inexact result is rounded to the nearest available value in the direction of positive infinity.
- Rounding towards –∞ : An inexact result is rounded to the nearest available value in the direction of negative infinity.

(1) Rounding to the nearest value is specified as the default mode and returns the most accurate value.
(2) Modes such as rounding towards 0, rounding towards +∞, and rounding towards –∞ are used to ensure precision when interval arithmetic is employed.

CV Flag (Invalid Operation Cause Flag), CO Flag (Overflow Cause Flag), CZ Flag (Division-by-Zero Cause Flag), CU Flag (Underflow Cause Flag), CX Flag (Inexact Cause Flag), and CE Flag (Unimplemented Processing Cause Flag)

Floating-point exceptions include the five specified in the IEEE754 standard, namely overflow, underflow, inexact, division-by-zero, and invalid operation. For a further floating-point exception that is generated upon detection of unimplemented processing, the corresponding flag (CE) is set to 1.

- The bit that has been set to 1 is set to 0 when the FPU instruction is executed.
- When 0 is written to the bit by the MVTC and POPC instructions, the bit is set to 0; the bit retains the previous value when 1 is written by the instruction.

DN Flag (0 Flush Bit of Denormalized Number)

When this bit is set to 0, a denormalized number is handled as a denormalized number. When this bit is set to 1, a denormalized number is handled as 0.

EV Bit (Invalid Operation Exception Enable), EO Bit (Overflow Exception Enable), EZ Bit (Division-by-Zero Exception Enable), EU Bit (Underflow Exception Enable), and EX Bit (Inexact Exception Enable)

When any of five floating-point exceptions specified in the IEEE754 standard is generated by the floating-point

operation instruction, the bit decides whether the CPU will start handling the exception. When the bit is set to 0, the exception handling is masked; when the bit is set to 1, the exception handling is enabled.

FV Flag (Invalid Operation Flag), FO Flag (Overflow Flag), FZ Flag (Division-by-Zero Flag), FU Flag (Underflow Flag), and FX Flag (Inexact Flag)

While the exception handling enable bit (Ej) is 0 (exception handling is masked), if any of five floating-point exceptions specified in the IEEE754 standard is generated, the corresponding bit is set to 1.

- When Ej is 1 (exception handling is enabled), the value of the flag remains.
- When the corresponding flag is set to 1, it remains 1 until it is set to 0 by software. (accumulation flag)

FS Flag (Floating-Point Error Summary Flag)

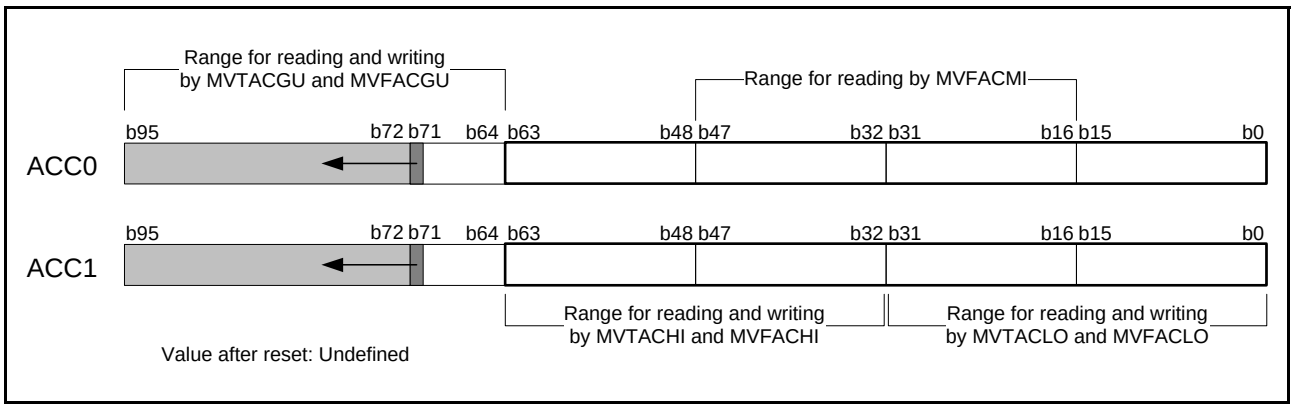
This bit reflects the logical OR of the FU, FZ, FO, and FV flags.

2.2.3 Accumulator

The accumulator (ACC0 or ACC1) is a 72-bit register used for DSP instructions. The accumulator is handled as a 96-bit register for reading and writing. At this time, when bits 95 to 72 of the accumulator are read, the value where the value of bit 71 is sign extended is read. Writing to bits 95 to 72 of the accumulator is ignored. ACC0 is also used for the multiply and multiply-and-accumulate instructions; EMUL, EMULU, FMUL, MUL, and RMPA, in which case the prior value in ACC0 is modified by execution of the instruction.

Use the MVTACGU, MVTACHI, and MVTACLO instructions for writing to the accumulator. The MVTACGU, MVTACHI, and MVTACLO instructions write data to bits 95 to 64, the higher-order 32 bits (bits 63 to 32), and the lower-order 32 bits (bits 31 to 0), respectively.

Use the MVFACGU, MVFACHI, MVFACMI, and MVFACLO instructions for reading data from the accumulator. The MVFACGU, MVFACHI, MVFACMI, and MVFACLO instructions read data from the guard bits (bits 95 to 64), higher-order 32 bits (bits 63 to 32), the middle 32 bits (bits 47 to 16), and the lower-order 32 bits (bits 31 to 0), respectively.



Note: The value of bit 71 is sign extended for bits 95 to 72 and the extended value is always read. Writing to this area is ignored.

2.3 Processor Mode

The RXv2 CPU supports two processor modes, supervisor and user. These processor modes and the memory protection function enable the realization of a hierarchical CPU resource protection and memory protection mechanism. Each processor mode imposes a level on rights of access to memory and the instructions that can be executed. Supervisor mode carries greater rights than user mode. The initial state after a reset is supervisor mode.

2.3.1 Supervisor Mode

In supervisor mode, all CPU resources are accessible and all instructions are available. However, writing to the processor mode select bit (PM) in the processor status word (PSW) by executing an MVTC or a POPC instruction will be ignored. For details on how to write to the PM bit, refer to section 2.2.2.5, Processor Status Word (PSW).

2.3.2 User Mode

In user mode, write access to the CPU resources listed below is restricted. The restriction applies to any instruction capable of write access.

- Some bits (bits IPL[3:0], PM, U, and I) in the processor status word (PSW)
- Interrupt stack pointer (ISP)
- Exception table register (EXTB)
- Interrupt table register (INTB)
- Backup PSW (BPSW)
- Backup PC (BPC)
- Fast interrupt vector register (FINTV)

2.3.3 Privileged Instruction

Privileged instructions can only be executed in supervisor mode. Executing a privileged instruction in user mode produces a privileged instruction exception. Privileged instructions include the RTFI, MVTIPL, RTE, and WAIT instructions.

2.3.4 Switching Between Processor Modes

Manipulating the processor mode select bit (PM) in the processor status word (PSW) switches the processor mode. However, rewriting to the PM bit by executing an MVTC or a POPC instruction is prohibited. Switch the processor mode by following the procedures described below.

(1) Switching from user mode to supervisor mode

After an exception has been generated, the PSW.PM bit is set to 0 and the CPU switches to supervisor mode. The hardware pre-processing is executed in supervisor mode. The state of the processor mode before the exception was generated is retained in the copy of PSW.PM bit is saved on the stack.

(2) Switching from supervisor mode to user mode

Executing an RTE instruction when the value of the copy of the PSW.PM bit that has been preserved on the stack is 1 or an RTFI instruction when the value of the copy of the PSW.PM bit that has been preserved in the backup PSW (BPSW) is 1 causes a transition to user mode. In the transition to user mode, the value of the stack pointer designation bit (the U bit in the PSW) becomes 1.

2.4 Data Types

The RXv2 CPU can handle four types of data: integer, floating-point, bit, and string.
For details, refer to RX Family RXv2 Instruction Set Architecture User's Manual: Software.

2.4.1 Integer

An integer can be signed or unsigned. For signed integers, negative values are represented by two's complements.

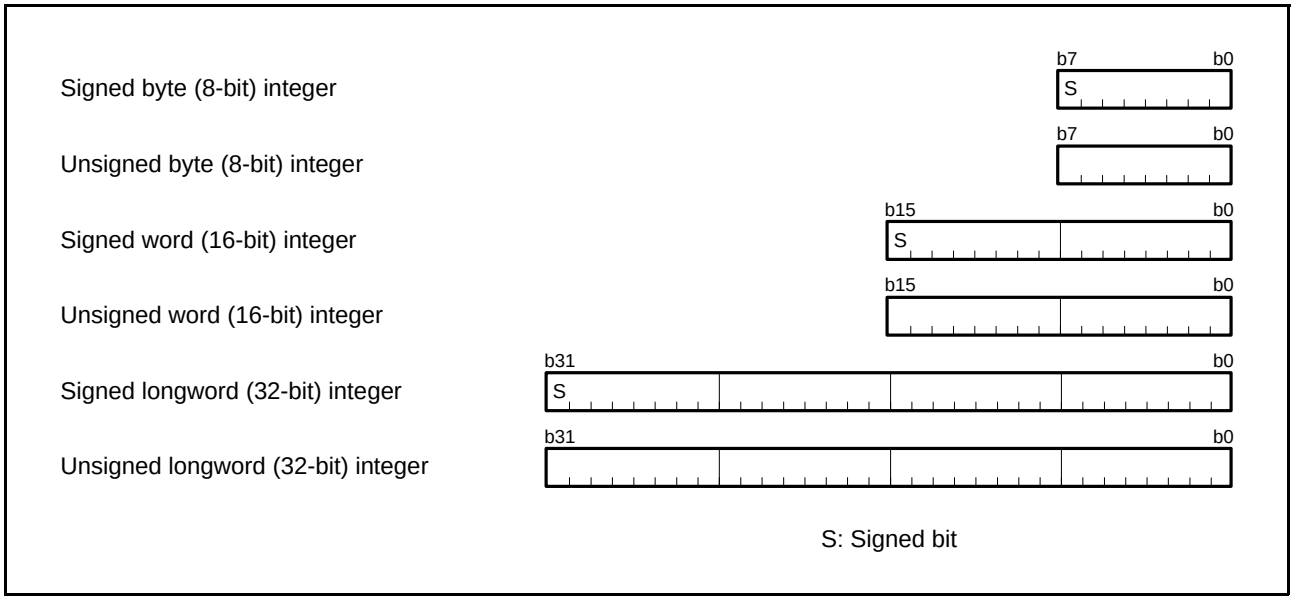


Figure 2.2 Integer

2.4.2 Floating-Points

Floating-point support is for the single-precision floating-point type specified in the IEEE754 standard; operands of this type can be used in eleven floating-point operation instructions: FADD, FCMP, FDIV, FMUL, FSQRT, FSUB, FTOI, FTOU, ITOF, ROUND, and UTOF.

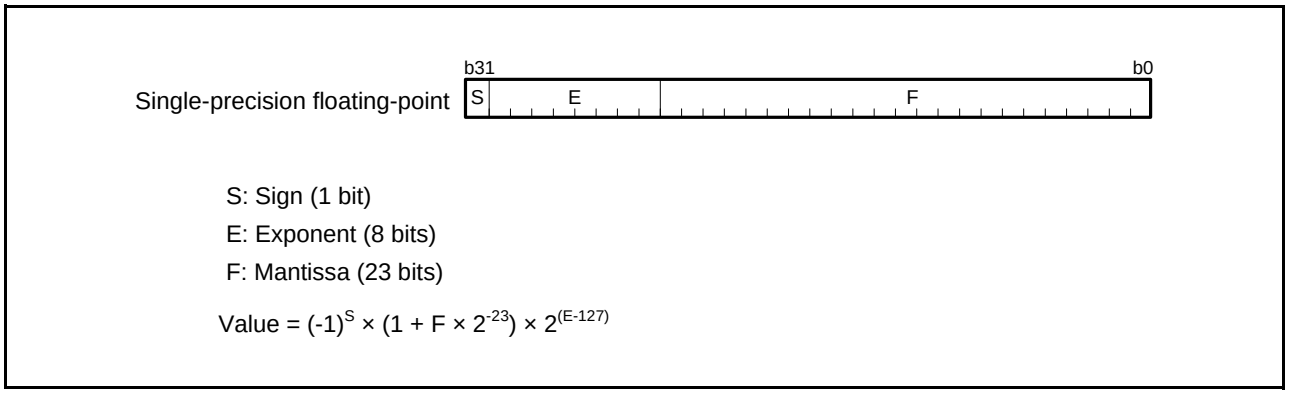


Figure 2.3 Floating-Point

The floating-point format supports the values listed below.

0 < E < 255 (normal numbers)
E = 0 and F = 0 (signed zero)
E = 0 and F > 0 (denormalized numbers)*1
E = 255 and F = 0 (infinity)
E = 255 and F > 0 (NaN: Not-a-Number)

Note 1. The number is treated as 0 when the FPSW.DN bit is 1. When the DN bit is 0, an unimplemented processing exception is generated.

2.4.3 Bitwise Operations

Five bit-manipulation instructions are provided for bitwise operations: BCLR, BMC*nd*, BNOT, BSET, and BTST.
A bit in a register is specified as the destination register and a bit number in the range from 31 to 0.
A bit in memory is specified as the destination address and a bit number from 7 to 0. The addressing modes available to specify addresses are register indirect and register relative.

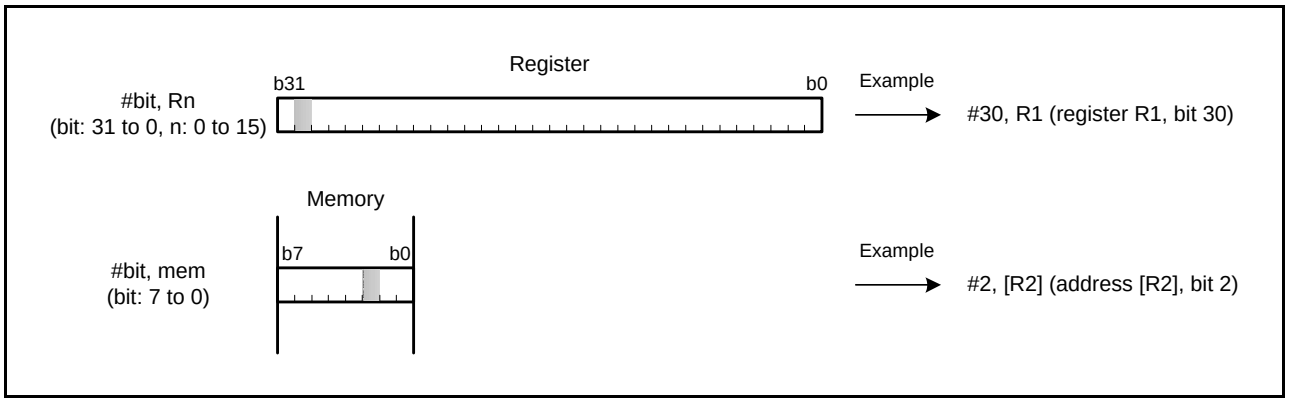


Figure 2.4 Bit

2.4.4 Strings

The string data type consists of an arbitrary number of consecutive byte (8-bit), word (16-bit), or longword (32-bit) units. Seven string manipulation instructions are provided for use with strings: SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE.

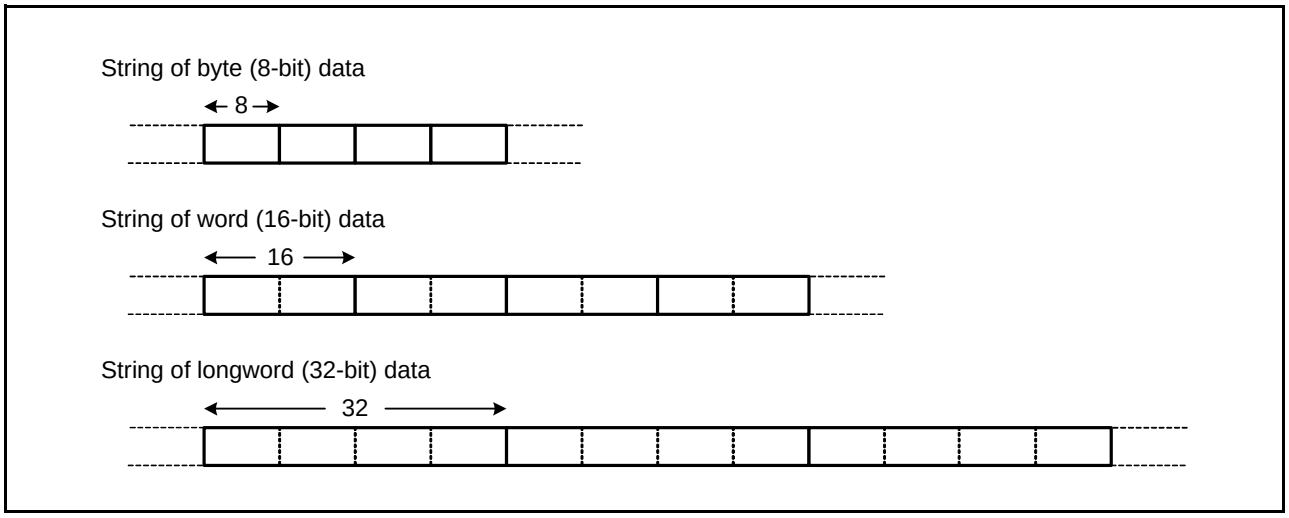


Figure 2.5 String

2.5 Endian

For the RXv2 CPU, instructions are little endian, but the treatment of data is selectable as little or big endian.

2.5.1 Switching the Endian

As arrangements of bytes, the RX64M Group supports both big endian, where the higher-order byte (MSB) is at location 0, and little endian, where the lower-order byte (LSB) is at location 0.
For details on the endian setting, see section 3, Operating Modes.
Operations for access differ according to the endian setting and, depending on the instruction, whether 8-, 16- or 32-bit access has been selected. Operations for access in the various possible cases are described in Table 2.1 to Table 2.12.
In the tables,

LL indicates bits D7 to D0 of the general-purpose register,
LH indicates bits D15 to D8 of the general-purpose register,
HL indicates bits D23 to D16 of the general-purpose register, and
HH indicates bits D31 to D24 of the general-purpose register.

	D31 to D24	D23 to D16	D15 to D8	D7 to D0
General purpose register: Rm	HH	HL	LH	LL

Table 2.1 32-Bit Read Operations when Little Endian has been Selected

Operation / Address of src	Reading a 32-bit unit from address 0	Reading a 32-bit unit from address 1	Reading a 32-bit unit from address 2	Reading a 32-bit unit from address 3	Reading a 32-bit unit from address 4
Address 0	Transfer to LL	—	—	—	—
Address 1	Transfer to LH	Transfer to LL	—	—	—
Address 2	Transfer to HL	Transfer to LH	Transfer to LL	—	—
Address 3	Transfer to HH	Transfer to HL	Transfer to LH	Transfer to LL	—
Address 4	—	Transfer to HH	Transfer to HL	Transfer to LH	Transfer to LL
Address 5	—	—	Transfer to HH	Transfer to HL	Transfer to LH
Address 6	—	—	—	Transfer to HH	Transfer to HL
Address 7	—	—	—	—	Transfer to HH

Table 2.2 32-Bit Read Operations when Big Endian has been Selected

Operation / Address of src	Reading a 32-bit unit from address 0	Reading a 32-bit unit from address 1	Reading a 32-bit unit from address 2	Reading a 32-bit unit from address 3	Reading a 32-bit unit from address 4
Address 0	Transfer to HH	—	—	—	—
Address 1	Transfer to HL	Transfer to HH	—	—	—
Address 2	Transfer to LH	Transfer to HL	Transfer to HH	—	—
Address 3	Transfer to LL	Transfer to LH	Transfer to HL	Transfer to HH	—
Address 4	—	Transfer to LL	Transfer to LH	Transfer to HL	Transfer to HH
Address 5	—	—	Transfer to LL	Transfer to LH	Transfer to HL
Address 6	—	—	—	Transfer to LL	Transfer to LH
Address 7	—	—	—	—	Transfer to LL

Table 2.3 32-Bit Write Operations when Little Endian has been Selected

Operation / Address of dest	Writing a 32-bit unit to address 0	Writing a 32-bit unit to address 1	Writing a 32-bit unit to address 2	Writing a 32-bit unit to address 3	Writing a 32-bit unit to address 4
Address 0	Transfer from LL	—	—	—	—
Address 1	Transfer from LH	Transfer from LL	—	—	—
Address 2	Transfer from HL	Transfer from LH	Transfer from LL	—	—
Address 3	Transfer from HH	Transfer from HL	Transfer from LH	Transfer from LL	—
Address 4	—	Transfer from HH	Transfer from HL	Transfer from LH	Transfer from LL
Address 5	—	—	Transfer from HH	Transfer from HL	Transfer from LH
Address 6	—	—	—	Transfer from HH	Transfer from HL
Address 7	—	—	—	—	Transfer from HH

Table 2.4 32-Bit Write Operations when Big Endian has been Selected

Operation / Address of dest	Writing a 32-bit unit to address 0	Writing a 32-bit unit to address 1	Writing a 32-bit unit to address 2	Writing a 32-bit unit to address 3	Writing a 32-bit unit to address 4
Address 0	Transfer from HH	—	—	—	—
Address 1	Transfer from HL	Transfer from HH	—	—	—
Address 2	Transfer from LH	Transfer from HL	Transfer from HH	—	—
Address 3	Transfer from LL	Transfer from LH	Transfer from HL	Transfer from HH	—
Address 4	—	Transfer from LL	Transfer from LH	Transfer from HL	Transfer from HH
Address 5	—	—	Transfer from LL	Transfer from LH	Transfer from HL
Address 6	—	—	—	Transfer from LL	Transfer from LH
Address 7	—	—	—	—	Transfer from LL

Table 2.5 16-Bit Read Operations when Little Endian has been Selected

Operation / Address of src	Reading a 16-bit unit from address 0	Reading a 16-bit unit from address 1	Reading a 16-bit unit from address 2	Reading a 16-bit unit from address 3	Reading a 16-bit unit from address 4	Reading a 16-bit unit from address 5	Reading a 16-bit unit from address 6
Address 0	Transfer to LL	—	—	—	—	—	—
Address 1	Transfer to LH	Transfer to LL	—	—	—	—	—
Address 2	—	Transfer to LH	Transfer to LL	—	—	—	—
Address 3	—	—	Transfer to LH	Transfer to LL	—	—	—
Address 4	—	—	—	Transfer to LH	Transfer to LL	—	—
Address 5	—	—	—	—	Transfer to LH	Transfer to LL	—
Address 6	—	—	—	—	—	Transfer to LH	Transfer to LL
Address 7	—	—	—	—	—	—	Transfer to LH

Table 2.6 16-Bit Read Operations when Big Endian has been Selected

Operation / Address of src	Reading a 16-bit unit from address 0	Reading a 16-bit unit from address 1	Reading a 16-bit unit from address 2	Reading a 16-bit unit from address 3	Reading a 16-bit unit from address 4	Reading a 16-bit unit from address 5	Reading a 16-bit unit from address 6
Address 0	Transfer to LH	—	—	—	—	—	—
Address 1	Transfer to LL	Transfer to LH	—	—	—	—	—
Address 2	—	Transfer to LL	Transfer to LH	—	—	—	—
Address 3	—	—	Transfer to LL	Transfer to LH	—	—	—
Address 4	—	—	—	Transfer to LL	Transfer to LH	—	—
Address 5	—	—	—	—	Transfer to LL	Transfer to LH	—
Address 6	—	—	—	—	—	Transfer to LL	Transfer to LH
Address 7	—	—	—	—	—	—	Transfer to LL

Table 2.7 16-Bit Write Operations when Little Endian has been Selected

Operation / Address of dest	Writing a 16-bit unit to address 0	Writing a 16-bit unit to address 1	Writing a 16-bit unit to address 2	Writing a 16-bit unit to address 3	Writing a 16-bit unit to address 4	Writing a 16-bit unit to address 5	Writing a 16-bit unit to address 6
Address 0	Transfer from LL	—	—	—	—	—	—
Address 1	Transfer from LH	Transfer from LL	—	—	—	—	—
Address 2	—	Transfer from LH	Transfer from LL	—	—	—	—
Address 3	—	—	Transfer from LH	Transfer from LL	—	—	—
Address 4	—	—	—	Transfer from LH	Transfer from LL	—	—
Address 5	—	—	—	—	Transfer from LH	Transfer from LL	—
Address 6	—	—	—	—	—	Transfer from LH	Transfer from LL
Address 7	—	—	—	—	—	—	Transfer from LH

Table 2.8 16-Bit Write Operations when Big Endian has been Selected

Operation / Address of dest	Writing a 16-bit unit to address 0	Writing a 16-bit unit to address 1	Writing a 16-bit unit to address 2	Writing a 16-bit unit to address 3	Writing a 16-bit unit to address 4	Writing a 16-bit unit to address 5	Writing a 16-bit unit to address 6
Address 0	Transfer from LH	—	—	—	—	—	—
Address 1	Transfer from LL	Transfer from LH	—	—	—	—	—
Address 2	—	Transfer from LL	Transfer from LH	—	—	—	—
Address 3	—	—	Transfer from LL	Transfer from LH	—	—	—
Address 4	—	—	—	Transfer from LL	Transfer from LH	—	—
Address 5	—	—	—	—	Transfer from LL	Transfer from LH	—
Address 6	—	—	—	—	—	Transfer from LL	Transfer from LH
Address 7	—	—	—	—	—	—	Transfer from LL

Table 2.9 8-Bit Read Operations when Little Endian has been Selected

Operation / Address of src	Reading an 8-bit unit from address 0	Reading an 8-bit unit from address 1	Reading an 8-bit unit from address 2	Reading an 8-bit unit from address 3
Address 0	Transfer to LL	—	—	—
Address 1	—	Transfer to LL	—	—
Address 2	—	—	Transfer to LL	—
Address 3	—	—	—	Transfer to LL

Table 2.10 8-Bit Read Operations when Big Endian has been Selected

Operation / Address of src	Reading an 8-bit unit from address 0	Reading an 8-bit unit from address 1	Reading an 8-bit unit from address 2	Reading an 8-bit unit from address 3
Address 0	Transfer to LL	—	—	—
Address 1	—	Transfer to LL	—	—
Address 2	—	—	Transfer to LL	—
Address 3	—	—	—	Transfer to LL

Table 2.11 8-Bit Write Operations when Little Endian has been Selected

Operation / Address of dest	Writing an 8-bit unit to address 0	Writing an 8-bit unit to address 1	Writing an 8-bit unit to address 2	Writing an 8-bit unit to address 3
Address 0	Transfer from LL	—	—	—
Address 1	—	Transfer from LL	—	—
Address 2	—	—	Transfer from LL	—
Address 3	—	—	—	Transfer from LL

Table 2.12 8-Bit Write Operations when Big Endian has been Selected

Operation / Address of dest	Writing an 8-bit unit to address 0	Writing an 8-bit unit to address 1	Writing an 8-bit unit to address 2	Writing an 8-bit unit to address 3
Address 0	Transfer from LL	—	—	—
Address 1	—	Transfer from LL	—	—
Address 2	—	—	Transfer from LL	—
Address 3	—	—	—	Transfer from LL

2.5.2 Access to I/O Registers

The addresses of I/O registers are fixed, and this is regardless of whether the setting is for little endian or big endian. Accordingly, changes to the endian do not affect access to I/O registers. For the arrangements of I/O registers, refer to the descriptions of registers in the relevant sections.

2.5.3 Notes on Access to I/O Registers

Ensure that access to I/O registers is in accord with the following rules.

- With I/O registers for which a bus width of eight bits is indicated, use instructions having operands of the same width (eight bits). That is, access these registers by using instructions with .B as the size specifier (.size), or with .B or .UB as the size-extension specifier (.memex).
- With I/O registers for which a bus width of 16 bits is indicated, use instructions having operands of the same width (16 bits). That is, access these registers by using instructions with .W as the size specifier (.size), or with .W or .UW as the size-extension specifier (.memex).
- With I/O registers for which a bus width of 32 bits is indicated, use instructions having operands of the same width (32 bits). That is, access these registers by using instructions with .L as the size specifier (.size), or with .L size-extension specifier (.memex).

2.5.4 Data Arrangement

2.5.4.1 Data Arrangement in Registers

Figure 2.6 shows the relation between the sizes of registers and bit numbers.

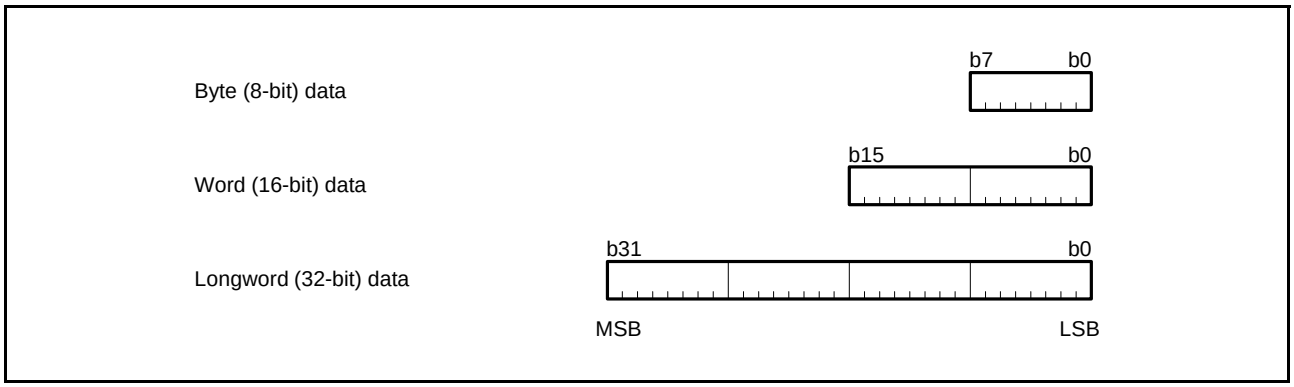


Figure 2.6 Data Arrangement in Registers

2.5.4.2 Data Arrangement in Memory

Data in memory have three sizes: byte (8-bit), word (16-bit), and longword (32-bit). The data arrangement is selectable as little endian or big endian. Figure 2.7 shows the arrangement of data in memory.

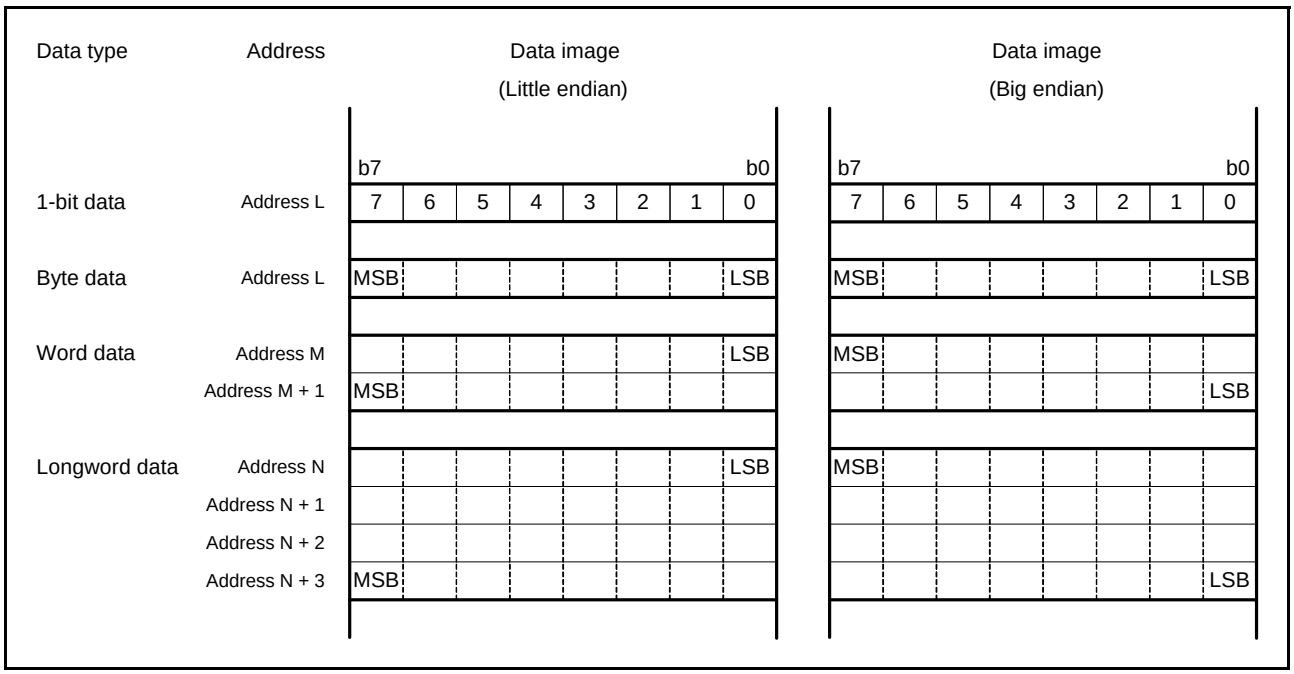


Figure 2.7 Data Arrangement in Memory

2.5.5 Notes on the Allocation of Instruction Codes

The allocation of instruction codes to an external space where the endian differs from that of the chip is prohibited. If the instruction codes are allocated to the external space, they must be allocated to areas where the endian setting is the same as that for the chip.

2.6 Vector Table

There are two types of vector table: exception and interrupt. Each vector in the vector table consists of four bytes and specifies the address where the corresponding exception handling routine starts.

2.6.1 Exception Vector Table

In the exception vector table, the individual vectors for the privileged instruction exception, access exception, undefined instruction exception, floating-point exception, non-maskable interrupt, and reset are allocated to the 124-byte area where the value indicated by the exception table register (EXTB) is used as the starting address (ExtBase). The reset vector is always allocated to FFFFFFFCh, regardless of the value of the exception vector table.
Figure 2.8 shows the exception vector table.

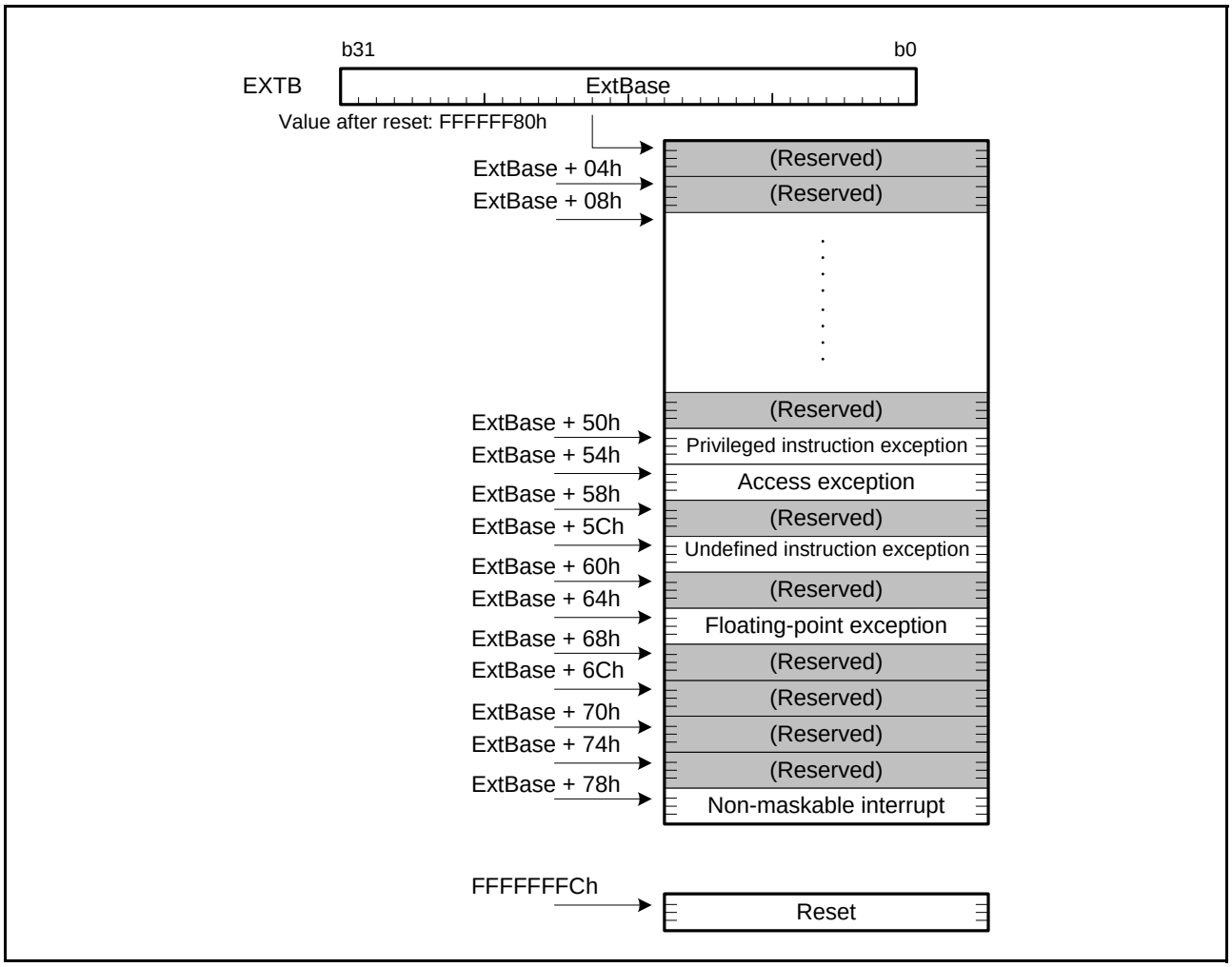


Figure 2.8 Exception Vector Table

2.6.2 Interrupt Vector Table

The address where the interrupt vector table is placed can be adjusted. The table is a 1,024-byte region that contains all vectors for unconditional traps and interrupts and starts at the address (IntBase) specified in the interrupt table register (INTB). Figure 2.9 shows the interrupt vector table.

Each vector in the interrupt vector table has a vector number from 0 to 255. Each of the INT instructions, which act as the sources of unconditional traps, is allocated to the vector that has the same number as is specified as the operand of the instruction itself (from 0 to 255). The BRK instruction is allocated to the vector with number 0. Furthermore, vector numbers (from 0 to 255) are allocated to interrupt requests in a fixed way for each product. For more on interrupt vector numbers, see section 15.3.1, Interrupt Vector Table.

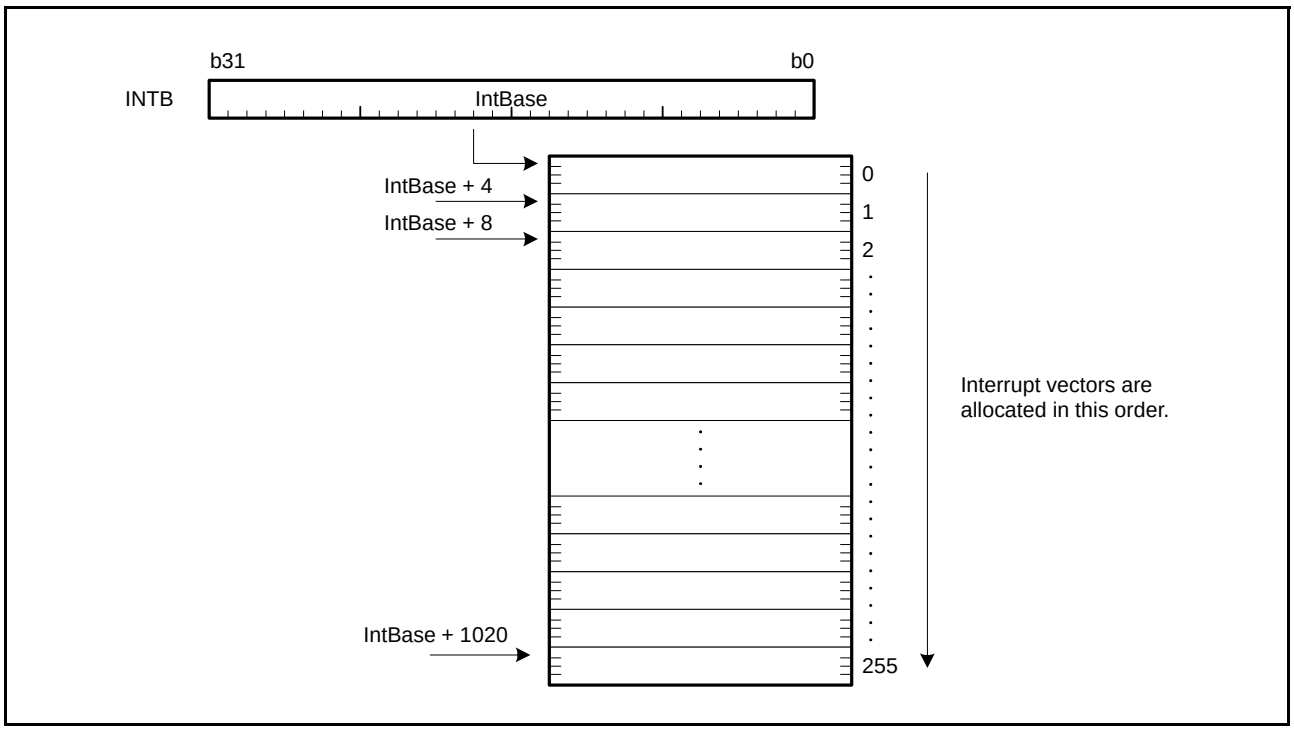


Figure 2.9 Interrupt Vector Table

2.7 Operation of Instructions

2.7.1 Restrictions on RMPA and String-Manipulation Instructions

2.7.1.1 Transfer Size and Data Prefetching

The RMPA instruction and the string-manipulation instructions (SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions) transfer data in longword units to speed up the reading of data from and writing of data to the memory. If the last of the data to be processed is less than a longword, data transfer proceeds with the sizes described below:

- RMPA, SSTR, SUNTIL, and SWHILE instructions: Size specified by the size specifier
- SCMPU, SMOVB, SMOVF, and SMOVU instructions: Byte

Additionally, in the above processing, the RMPA instruction and the string-manipulation instructions other than the SSTR instruction (that is, the SCMPU, SMOVB, SMOVF, SMOVU, SUNTIL, and SWHILE instructions) prefetch data when reading data from the memory. Data is prefetched from the prefetching start position with three bytes as the upper limit. The prefetching start positions of each operation are shown below.

- RMPA instruction: The multiplicand address specified by R1, and the multiplier address specified by R2
- SCMPU instruction: The source address specified by R1 for comparison, and the destination address specified by R2 for comparison
- SUNTIL and SWHILE instructions: The destination address specified by R1 for comparison
- SMOVB, SMOVF, and SMOVU instructions: The source address specified by R2 for transfer

2.7.1.2 Access to the External Space

Although the external space has a per-area ending-switching facility (only for data), the allocation of data to be handled by RMPA or string-manipulation instructions (SCMPU, SMOVB, SMOVF, SMOVU,SSTR, SUNTIL, and SWHILE instructions) to an area where the endian differs from that of the chip is prohibited, and operation is not guaranteed if this restriction is not observed. If data to be handled by RMPA or string-manipulation instructions are allocated to the external space, they must be allocated to areas where the endian setting is the same as that for the chip.

2.7.1.3 Access to I/O Registers

The allocation of data to be handled by RMPA or string-manipulation instructions (SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions) to I/O registers is prohibited, and operation is not guaranteed if this restriction is not observed.

2.8 Number of Cycles

2.8.1 Instruction and Number of Cycle

Table 2.13 to Table 2.20 show the number of cycles in operation of each instruction. The listed numbers of cycles for access to memory are the numbers of cycles during no-wait access. The operands in the table below indicate the following meanings.

#IMM: Immediate
flag: bit, flag
Rs, Rs2, Rd, Rd2, Ri, Rb: General-purpose register
As, Ad: Accumulator
CR: Control register
dsp: displacement
pcdsp: displacement

Table 2.13 Number of Cycles for Arithmetic/logic Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Number of Cycles
Arithmetic/logic instructions (register-register, immediate-register)	• {ABS, NEG, NOT} “Rd”/“Rs, Rd” • {ADC, MAX, MIN, ROTL, ROTR, XOR} “#IMM, Rd”/“Rs, Rd” • ADD “#IMM, Rd”/“Rs, Rd”/“#IMM, Rs, Rd”/“Rs, Rs2, Rd” • {AND, MUL, OR, SUB} “#IMM, Rd”/“Rs, Rd”/“Rs, Rs2, Rd” • {CMP, TST} “#IMM, Rs”/“Rs, Rs2” • NOP • {ROLC, RORC, SAT} “Rd” • SBB “Rs, Rd” • {SHAR, SHLL, SHLR} “#IMM, Rd”/“Rs, Rd”/“#IMM, Rs, Rd”	1
	• DIV “#IMM, Rd”/“Rs, Rd”	3 to 20*1
	• DIVU “#IMM, Rd”/“Rs, Rd”	2 to 18*1
	• {EMUL, EMULU} “#IMM, Rd”/“Rs, Rd”	2
	• SATR	3
Arithmetic/logic instructions (memory source operand)	• {ADC, ADD, AND, MAX, MIN, MUL, OR, SBB, SUB, XOR} “[Rs], Rd”/“dsp[Rs], Rd” • {CMP, TST} “[Rs], Rs2”/“dsp[Rs], Rs2”	3
	• DIV “[Rs], Rd / dsp[Rs], Rd”	5 to 22
	• DIVU“[Rs], Rd / dsp[Rs], Rd”	4 to 20
	• {EMUL, EMULU} “[Rs], Rd”/“dsp[Rs], Rd”	4
	• RMPA.B	6+7×floor(n/4)+4×(n%4) n: Number of processing bytes*2
	• RMPA.W	6+5×floor(n/2)+4×(n%2) n: Number of processing words*2
	• RMPA.L	6+4n n: Number of processing longwords*2

Note 1. The number of cycles for the dividing instruction varies according to the divisor and dividend.
Note 2. floor (x): Max. integer that is smaller than x.

Table 2.14 Number of Cycles for Transfer Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Number of Cycles
Transfer instructions (register-register, immediate-register)	• MOV “#IMM, Rd”/“Rs, Rd” • {MOVU, REVL, REVW} “Rs, Rd” • SCC*nd* “Rd” • {STNZ, STZ} “#IMM, Rd”/ “Rs, Rd”	1
	• XCHG “Rs, Rd”	2
Transfer instructions (load operation)	• {MOV, MOVU} “[Rs], Rd”/“dsp[Rs], Rd”/“[Rs+], Rd”/“[-Rs], Rd”/“[Ri, Rb], Rd” • LDL “[Rs], Rd” • POP “Rd”	Throughput: 1 Latency: 2*1
	• POPC “CR”	Throughput: 3 Latency: 4*1
	• POPM “Rd-Rd2”	Throughput: n Latency: n+1 n: Number of registers*1, *2
Transfer instructions (store operation)	• MOV “Rs, [Rd]”/“Rs, dsp[Rd]”/“Rs, [Rd+]”/“Rs, [-Rd]”/“Rs, [Ri, Rb]”/“#IMM, dsp[Rd]”/“#IMM, [Rd]” • PUSH “Rs” • PUSHC “CR” • SCCnd “[Rd]”/“dsp[Rd]” • STC “Rs, [Rd]”	1
	• PUSHM “Rs-Rs2”	n n: Number of registers*3
Transfer instructions (memory-register)	• XCHG “[Rs], Rd”/“dsp[Rs], Rd”	2
Transfer instructions (memory-memory)	• MOV “[Rs], [Rd]”/“dsp[Rs], [Rd]”/“[Rs], dsp[Rd]”/“dsp[Rs], dsp[Rd]” • PUSH “[Rs]”/“dsp[Rs]”	3

Note 1. When the load data is used by the subsequent instruction, the number of cycles described as “latency” is counted as the number of cycles for the memory load instruction. For the cycles other than the memory load instruction, the number of cycles described as “throughput” is counted.

Note 2. The POPM instruction is converted into multiple load operations. The processing is the same as the one for the load operations of the MOV instruction, where the operation is repeated for the number of specified registers.

Note 3. The PUSHM instruction is converted into multiple store operations. The processing is the same as the one for the store operations of the MOV instruction, where the operation is repeated for the number of specified registers.

Table 2.15 Number of Cycles for Bit Manipulation Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Number of Cycles
Bit manipulation instructions (register)	• {BCLR, BNOT, BSET} “#IMM, Rd”/“Rs, Rd” • BM*Cnd* “#IMM, Rd” • BTST “#IMM, Rs”/“Rs, Rs2”	1
Bit manipulation instructions (memory source operand)	• {BCLR, BNOT, BSET} “#IMM, [Rd]”/“#IMM, dsp[Rd]”/“Rs, [Rd]”/“Rs, dsp[Rd]” • BM*Cnd* “#IMM, [Rd]”/“#IMM, dsp[Rd]” • BTST “#IMM, [Rs]”/“#IMM, dsp[Rs]”/“Rs, [Rs2]”/“Rs, dsp[Rs2]”	3

Table 2.16 Number of Cycles for Branch Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Number of Cycles
Branch instructions	• B*Cnd* “pcdsp” • {BRA, BSR} “pcdsp”/“Rs” • {JMP, JSR} “Rs”	Branch taken: 3 Branch not taken: 1
	• RTE	6
	• RTFI	3
	• RTS	5
	• RTSD “#IMM”	5
	• RTSD “#IMM, Rd-Rd2”	Throughput: n<5?5:1+n Latency: n<4?5:2+n n: Number of registers*1

?: Conditional operator

Note 1. When the load data is used by the subsequent instruction, the number of cycles described as “latency” is counted as the number of cycles for the memory load instruction. For the cycles other than the memory load instruction, the number of cycles described as “throughput” is counted.

Table 2.17 Number of Cycles for Floating-Point Operation Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Number of Cycles
Floating-point operation instructions (register-register, immediate-register)	• {FADD, FSUB} “#IMM, Rd”/ “Rs, Rd”/ “Rs, Rs2, Rd”	2
	• FCMP “#IMM, Rs”/“Rs, Rs2”	1
	• FDIV “#IMM, Rd”/“Rs, Rd”	16
	• FMUL “#IMM, Rd”/ “Rs, Rd” / “Rs, Rs2, Rd”	2
	• FSQRT “Rs, Rd”	16
	• {FTOI, ROUND, ITOF} “Rs, Rd”	2
	• {FTOU, UTOF} “Rs, Rd”	2
Floating-point operation instructions (memory source operand)	• {FADD, FSUB} “[Rs], Rd”/“dsp[Rs], Rd”	4
	• FCMP “[Rs], Rs2”/“dsp[Rs], Rs2”	3
	• FDIV “[Rs], Rd”/“dsp[Rs], Rd”	18
	• FMUL “[Rs], Rd”/“dsp[Rs], Rd”	4
	• FSQRT “[Rs], Rd”/“dsp[Rs], Rd”	18
	• {FTOI, ROUND, ITOF} “[Rs], Rd”/“dsp[Rs], Rd”	4
	• {FTOU, UTOF} “[Rs], Rd”/“dsp[Rs], Rd”	4

Table 2.18 Number of Cycles for DSP Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Number of Cycles
DSP instructions	• {EMULA, EMACA, EMSBA, MULLH, MULHI, MULLO, MACLH, MACHI, MACLO, MSBLH, MSBHI, MSBLO}“Rs, Rs2, Ad” • {MVFACHI, MVFACMI, MVFACLO, MVFACGU} “#IMM, As, Rd” • {MVTACHI, MVTACLO, MVTACGU} “As, Rd” • {RDACW, RDACL, RACW, RACL} “#IMM, Ad”	1

Table 2.19 Number of Cycles for String Manipulation Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Number of Cycles
String manipulation instructions*1	• SCMPU	2+4×floor(n/4)+4×(n%4) n: Number of comparison bytes*2
	• SMOVB	n>3?6+3×floor(n/4)+3×(n%4):2+3n n: Number of transfer bytes*2
	• SMOVF, SMOVU	2+3×floor(n/4)+3×(n%4) n: Number of transfer bytes*2
	• SSTR.B	2+floor(n/4)+n%4 n: Number of transfer bytes*2
	• SSTR.W	2+floor(n/2)+n%2 n: Number of transfer words*2
	• SSTR.L	2+n n: Number of transfer longwords
	• SUNTIL.B, SWHILE.B	3+3×floor(n/4)+3×(n%4) n: Number of comparison bytes*2
	• SUNTIL.W, SWHILE.W	3+3×floor(n/2)+3×(n%2) n: Number of comparison words*2
	• SUNTIL.L, SWHILE.L	3+3×n n: Number of comparison longwords

?: Conditional operator

Note 1. Each of the SCMPU, SMOVU, SWHILE, and SUNTIL instructions ends the execution regardless of the specified cycles, if the end condition is satisfied during execution.
Note 2. floor (x): Max. integer that is smaller than x.

Table 2.20 Number of Cycles for System Manipulation Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Number of Cycles
System manipulation instructions	• {CLRPSW, SETPSW}“flag” • MVTC “#IMM, CR”/“Rs, CR” • MVFC “CR, Rd” • MVTIPL “#IMM”	1
	• RTE	6
	• RTFI	3

2.8.2 Numbers of Cycles for Response to Interrupts

Table 2.21 lists numbers of cycles taken by processing for response to interrupts.

Table 2.21 Numbers of Cycles for Response to Interrupts

Type of Interrupt Request/Details of Processing	Fast Interrupt	Other Interrupts
ICU Judgment of priority order	2 cycles	
CPU Number of cycles from notification to acceptance of the interrupt request	N cycles (varies with the instruction being executed at the time the interrupt was received)	
CPU Pre-processing by hardware Saving the current PC and PSW values in RAM (or in control registers in the case of the fast interrupt) Reading of the vector Branching to the start of the exception handling routine	4 cycles	6 cycles

Times calculated from the values in Table 2.21 will be applicable when access to memory from the CPU is processed with no waiting. This MCU has a code flash memory and RAM that allow no-wait access. Numbers of cycles for response to interrupts can be minimized by placing program code (and vectors) in code flash memory and the stack in RAM. Furthermore, place the addresses where the exception handling routine start on 8-byte boundaries.
For information on the number of cycles from notification to acceptance of the interrupt request, indicated by N in the table above, see Table 2.13 to Table 2.20.
The timing of interrupt acceptance depends on the execution state of the instruction. For more information on this, see section 14.3.1, Acceptance Timing and Saved PC Value.

3. Operating Modes

3.1 Operating Mode Types and Selection

There are two types of operating-mode selection: one is be selected by the level on pins at the time of release from the reset state, and the other is selected by software after release from the reset state.
Table 3.1 shows the relationship between levels on the mode-setting pins (MD and PC7/UB) on release from the reset state and the operating mode selected at that time. For details on each of the operating modes, see section 3.3, Details of Operating Modes. Operation starts with the on-chip ROM (code flash memory and data flash memory) enabled and the external bus disabled, regardless of the mode in which operation started. Set the SYSCR0.EXBE bit to 1 (external bus enabled) to enable the external bus.

Table 3.1 Selection of Operating Modes by the Mode-Setting Pins on Release from the Reset State

Mode-Setting Pin			SYSCR0 Initial State	
MD*1	PC7/UB*2	Operating Mode	ROME	EXBE
High	—	Single-chip mode	1 (On-chip ROM enabled)	0 (External bus disabled)
Low	Low	Boot mode (SCI interface)		
	High	Boot mode (USB interface)		
		User boot mode		

Note 1. Do not change the level on the MD pin while the MCU is operating.
Note 2. The PC7 pin, which is multiplexed with the UB pin function, may also be used as a general port pin.

Table 3.2 gives a list of the operating mode settings that can be made with system control register 0 (SYSCR0). For details on each of the operating modes, see section 3.3, Details of Operating Modes.

Table 3.2 Selection of Operating Modes by Register Setting

SYSCR0		
ROME	EXBE	Operating Mode
0 (On-chip ROM disabled)*1	0 (external bus disabled)	Single-chip mode, user boot mode
1 (On-chip ROM enabled)	0 (external bus disabled)	
0 (On-chip ROM disabled)*1	1 (external bus enabled)	On-chip ROM disabled extended mode
1 (On-chip ROM enabled)	1 (external bus enabled)	On-chip ROM enabled extended mode

Note 1. Once the ROME bit is set to 0, it cannot be reverted to 1.

The endian is selectable in single-chip mode and user boot mode. Endian is selected by the endian selection bits (MDE[2:0]) in the endian select register (MDE). Table 3.3 lists the correspondence between the setting and endian. For details on selection of endian, see section 7.2.5, Endian Select Register (MDE).

Table 3.3 Selection of Endian

Setting of the MDE[2:0] Bits	Selected Endian
000b	Big endian
111b	Little endian

3.2 Register Descriptions

3.2.1 Mode Monitor Register (MDMONR)

Address(es): 0008 0000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MD
Value after reset:	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Bit Name	Description	R/W
b0	MD	MD Pin Status Flag	0: The MD pin is low. 1: The MD pin is high.	R
b7 to b1	—	Reserved	These bits are read as 0.	R
b8	—	Reserved	The read value is undefined.	R
b15 to b9	—	Reserved	These bits are read as 0.	R

Note 1. This affects the level on the MD pin at the time of release from the reset state.

3.2.2 Mode Status Register (MDSR)

Address(es): 0008 0002h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	UBTS	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0/1*1	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 1.	R
b4 to b1	—	Reserved	These bits are read as 0.	R
b5	UBTS	User Boot Mode Startup Flag	0: Started with single-chip mode. 1: Started with user boot mode.	R
b15 to b6	—	Reserved	These bits are read as 0.	R

Note 1. Depends on the operating mode at startup.

3.2.3 System Control Register 0 (SYSCR0)

Address(es): 0008 0006h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	EXBE	ROME
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	ROME	On-Chip ROM Enable	0: The on-chip ROM is disabled. 1: The on-chip ROM is enabled.	R/W
b1	EXBE	External Bus Enable	0: The external bus is disabled. 1: The external bus is enabled.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	SYSCR0 Key Code	These bits control permission and prohibition of writing to the SYSCR0 register. To modify the SYSCR0 register, write 5Ah to the eight higher-order bits and the desired value to the eight lower-order bits as a 16-bit unit.	R/(W)*1

Note 1. Write data is not retained.

ROME Bit (On-Chip ROM Enable)

The ROME bit enables or disables the on-chip ROM (code flash memory and data flash memory).
Once set to 0, it cannot be reverted to 1.
A 0 should not be written to this bit while a program is being executed from the on-chip ROM (code flash memory and data flash memory). After writing a 0 to this bit, be sure to disable the on-chip ROM by changing the ROME bit to 0 before proceeding with further processing.

EXBE Bit (External Bus Enable)

The EXBE bit enables or disables the external bus.
Do not write 0 to this bit while a program is running from an external address space. Write 0 to this bit after access to the external bus is completed. Furthermore, when an external address space is included in the range of transfer by the bus masters other than the CPU (DMAC, DTC, EXDMAC, and EDMAC), prohibit DMA transfer before writing 0 to this bit. After writing to the EXBE bit, confirm that its value has actually changed before proceeding with further processing.
When the EXBE bit is set to 1, the related I/O port settings must also be changed as required. For details, see section 23, Multi-Function Pin Controller (MPC).

3.2.4 System Control Register 1 (SYSCR1)

Address(es): 0008 0008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SBYRAME	ECCRAME	—	—	—	—	—	RAME
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	RAME	RAM Enable	0: The RAM is disabled. 1: The RAM is enabled.	R/W
b5 to b1	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b6	ECCRAME	ECCRAM Enable	0: The ECCRAM is disabled. 1: The ECCRAM is enabled.	R/W
b7	SBYRAME	Standby RAM Enable	0: The standby RAM is disabled. 1: The standby RAM is enabled.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RAME Bit (RAM Enable)

The RAME bit enables or disables the RAM.

A 0 should not be written to this bit during access to the RAM. When accessing the RAM immediately after changing the RAME bit from 0 to 1, make sure that the RAME bit is 1 before the access.

Even when the RAME bit is set to 0, the RAM retains its value. However, the voltage must be maintained at no less than the specified RAM standby voltage (VRAM), which is stipulated in section 64, Electrical Characteristics.

ECCRAME Bit (ECCRAM Enable)

The ECCRAME bit enables or disables the ECCRAM.

A 0 should not be written to this bit during access to the ECCRAM. When accessing the ECCRAM immediately after changing the ECCRAME bit from 0 to 1, make sure that the ECCRAME bit is 1 before the access.

Even when the ECCRAME bit is set to 0, the ECCRAM retains its value. However, the voltage must be maintained at no less than the specified RAM standby voltage (VRAM), which is stipulated in section 64, Electrical Characteristics.

SBYRAME Bit (Standby RAM Enable)

The SBYRAME bit enables or disables the standby RAM.

A 0 should not be written to this bit during access to the standby RAM. When accessing the standby RAM immediately after changing the SBYRAME bit from 0 to 1, make sure that the SBYRAME bit is 1 before the access.

Even when the SBYRAME bit is set to 0, the standby RAM retains its value. However, the voltage must be maintained at no less than the specified RAM standby voltage (VRAM), which is stipulated in section 64, Electrical Characteristics.

3.3 Details of Operating Modes

3.3.1 Single-Chip Mode

In single-chip mode, the external bus is disabled (SYSCR0.EXBE bit = 0) so that all I/O port pins are available for use as input or output port pins, inputs or outputs for peripheral functions, or as interrupt inputs.
If the high level is on the MD pin at the time of release from the reset state, the chip starts in single-chip mode. The on-chip ROM is enabled (SYSCR0.ROME bit = 1) at this time. The on-chip ROM can be disabled by software (by clearing the SYSCR0.ROME bit to 0), but it cannot be re-enabled (by setting the SYSCR0.ROME bit to 1) once this is done.
Setting the SYSCR0.EXBE bit to 1 (enabling the external bus) causes a transition to on-chip ROM enabled extended mode or to on-chip ROM disabled extended mode, making the external bus available.

3.3.2 On-Chip ROM Enabled Extended Mode

In this mode, the on-chip ROM is enabled (SYSCR0.ROME bit = 1) and the external bus extension is enabled (SYSCR0.EXBE bit = 1). This mode allows some I/O ports to be used as data bus input/output, address bus output, or bus control signal input/output. For details, see section 23, Multi-Function Pin Controller (MPC).
After the chip has started up in single-chip mode, setting the SYSCR0.EXBE bit to 1 (external bus enabled) causes it to make the transition to on-chip ROM enabled extended mode.
Writing 0 to the SYSCR0.EXBE bit (external bus disabled) causes a transition to single-chip mode (on-chip ROM enabled).
Writing 0 to the SYSCR0.ROME bit (on-chip ROM disabled) causes a transition to on-chip ROM disabled extended mode.

3.3.3 On-Chip ROM Disabled Extended Mode

In this mode, the on-chip ROM is disabled (SYSCR0.ROME bit = 0) and the external bus extension is enabled (SYSCR0.EXBE bit = 1). This mode allows some I/O ports to be used as data bus input/output, address bus output, or bus control signal input/output. For details, see section 23, Multi-Function Pin Controller (MPC).
After the chip has started up in single-chip mode, setting the SYSCR0.EXBE bit to 1 (external bus enabled) and setting the SYSCR0.ROME bit to 0 (on-chip ROM disabled) causes it to make the transition to on-chip ROM disabled extended mode.
In this mode, the on-chip ROM cannot be enabled by setting the SYSCR0.ROME bit to 1.
Writing 0 to the SYSCR0.EXBE bit (external bus disabled) causes a transition to single-chip mode (on-chip ROM disabled).

3.3.4 Boot Mode

In this mode, the on-chip flash memory modifying program (boot program) stored in a dedicated area within the MCU operates. The on-chip ROM (code flash memory and data flash memory) can be modified from outside the MCU by using a universal asynchronous receiver/transmitter (SCI1). For details, see section 63, Flash Memory.
The chip starts up in boot mode if the low level is on both the MD and UB pins on release from the reset state.

3.3.5 USB Boot Mode

In this mode, the on-chip flash memory modifying program (USB boot program) stored in the boot area at the time of shipping operates. On-chip ROM (code flash memory and data flash memory) can be modified from outside the MCU by using the USB. For details, see section 63, Flash Memory.
The chip starts up in boot mode (with the USB interface) if the low level is on the MD pin and the high level is on the UB pin on release from the reset state while UB code A is not 5573 6572h, 426F 6F74h.

3.3.6 User Boot Mode

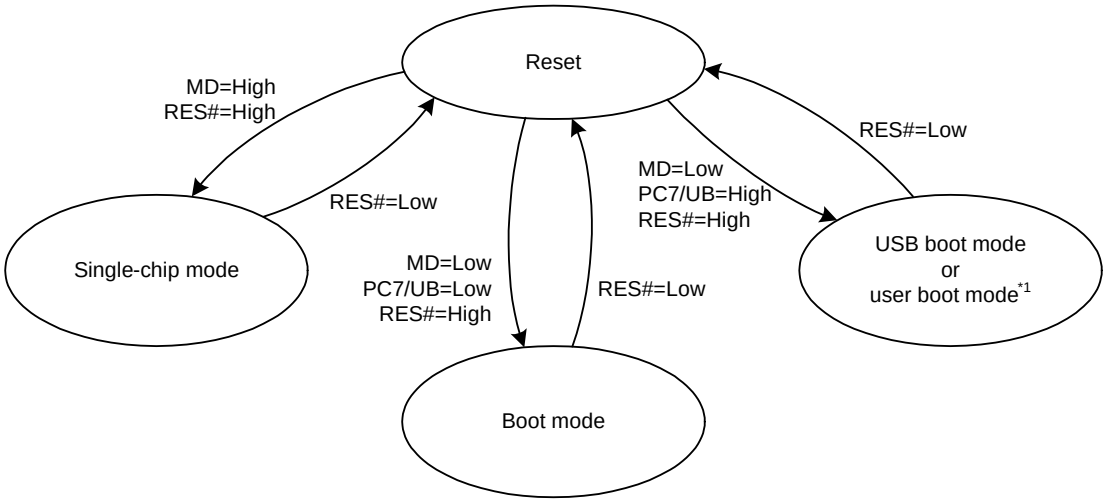
In user boot mode, an on-chip flash memory modifying program (user boot program) created by the user operates. After release from the reset state, the chip starts up in a state equivalent to single-chip mode.
After programming the prescribed values for UB code A and the UB code B, the chip starts up in user boot mode if the low level is on the MD pin and the high level is on the PC7 pin on release from the reset state. For UB code A and UB code b, see section 7, Option-Setting Memory.
After the chip has started up in user boot mode, setting the SYSCR0.EXBE bit to 1 (external bus enabled) causes it to make the transition to on-chip ROM enabled extended mode.

Note: In user boot mode, do not make a transition to software standby mode or deep software standby mode.
Note: The setting in the OFS0/OFS1 registers is ineffective in user boot mode, and the value becomes FFFF FFFFh.

3.4 Transitions of Operating Modes

3.4.1 Operating Mode Transitions Determined by the Mode-Setting Pins

Figure 3.1 shows operating mode transitions determined by the settings of the MD pin and the PC7/UB pin.



Note 1. To start the chip in user boot mode, settings for the UB code A and UB code B in the option setting memory are required along with the levels set on the mode-setting pins.

Figure 3.1 Mode-Setting Pin Level and Operating Mode

3.4.2 Operating Mode Transitions According to Register Setting

Figure 3.2 shows operating mode transitions according to the setting of the ROME and EXBE bits in SYSCR0.

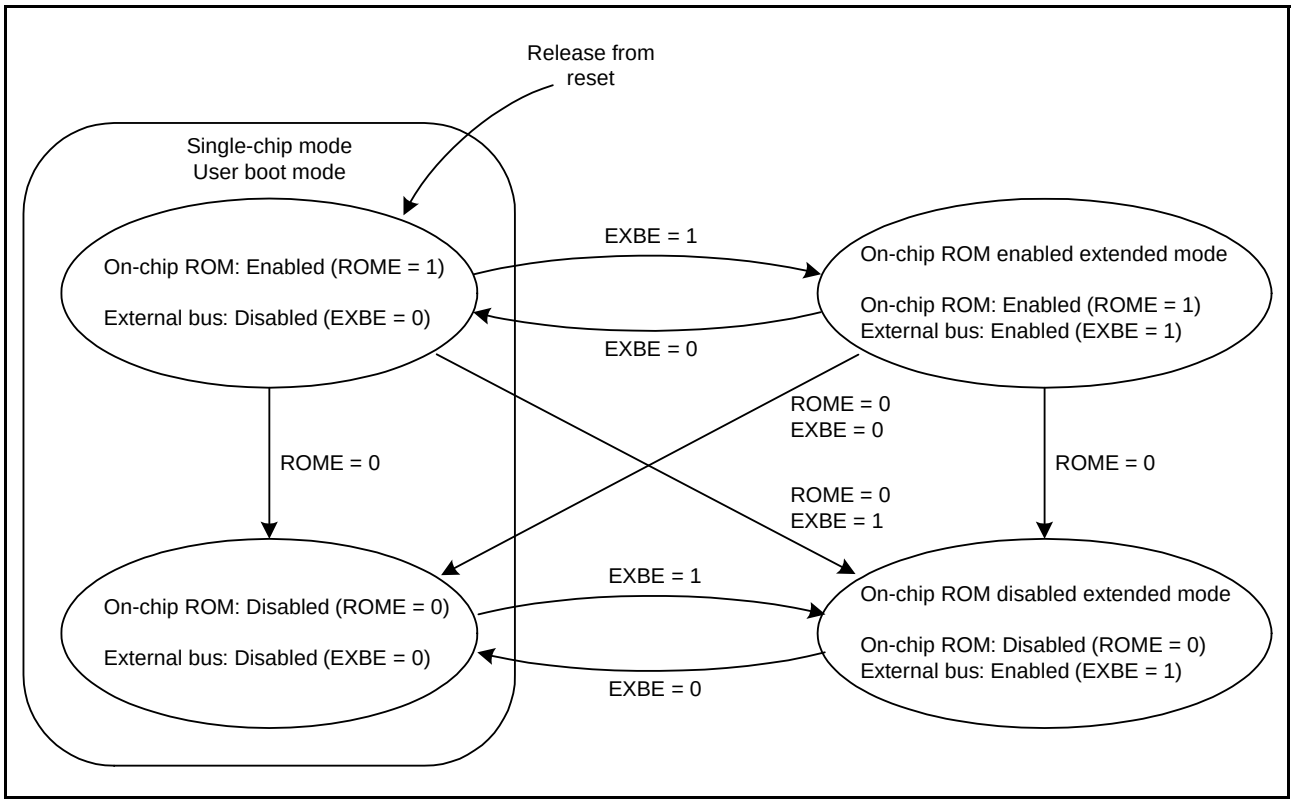
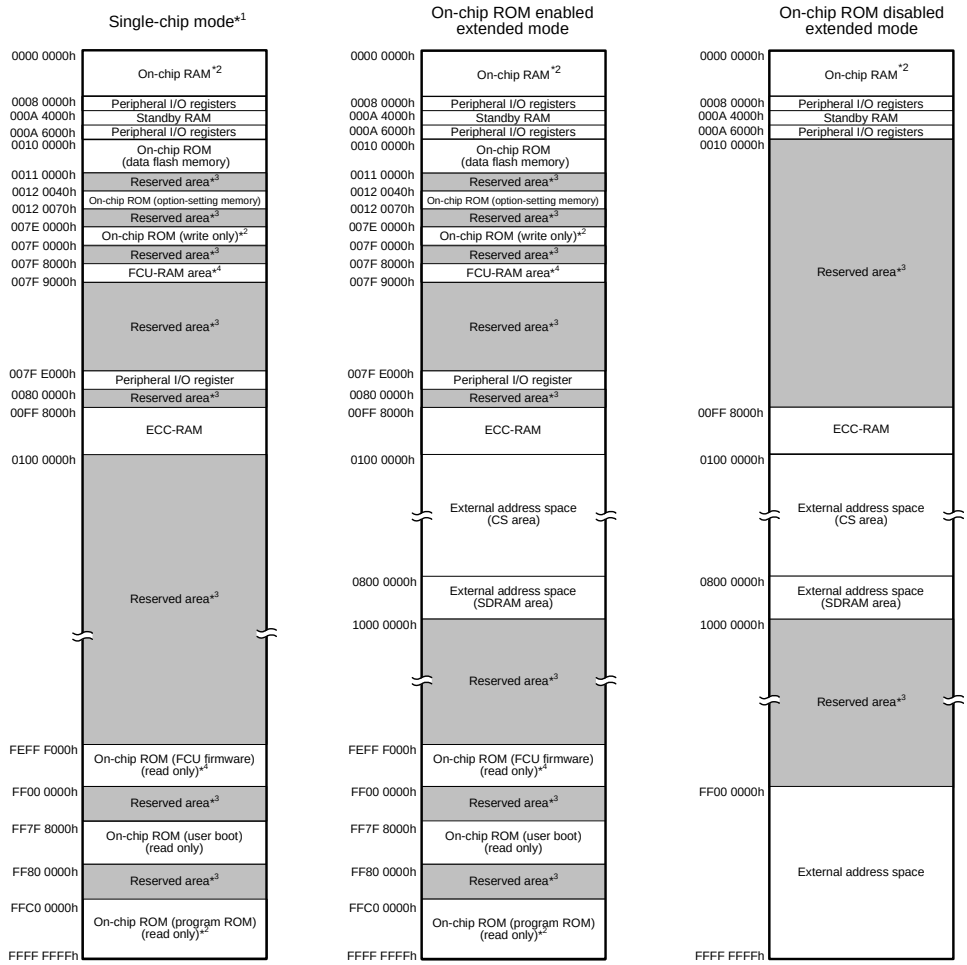


Figure 3.2 Setting of SYSCR0.ROME and EXBE Bits and Operating Modes

4. Address Space

4.1 Address Space

This MCU has a 4-Gbyte address space, consisting of the range of addresses from 0000 0000h to FFFF FFFFh. That is, linear access to an address space of up to 4 Gbytes is possible, and this contains both program and data areas.
Figure 4.1 shows the memory maps in the respective operating modes. Accessible areas will differ according to the operating mode and states of control bits.



Note 1. The address space in boot mode and user boot mode/USB boot mode is the same as the address space in single-chip mode.
Note 2. The capacity of ROM/RAM differs depending on the products.

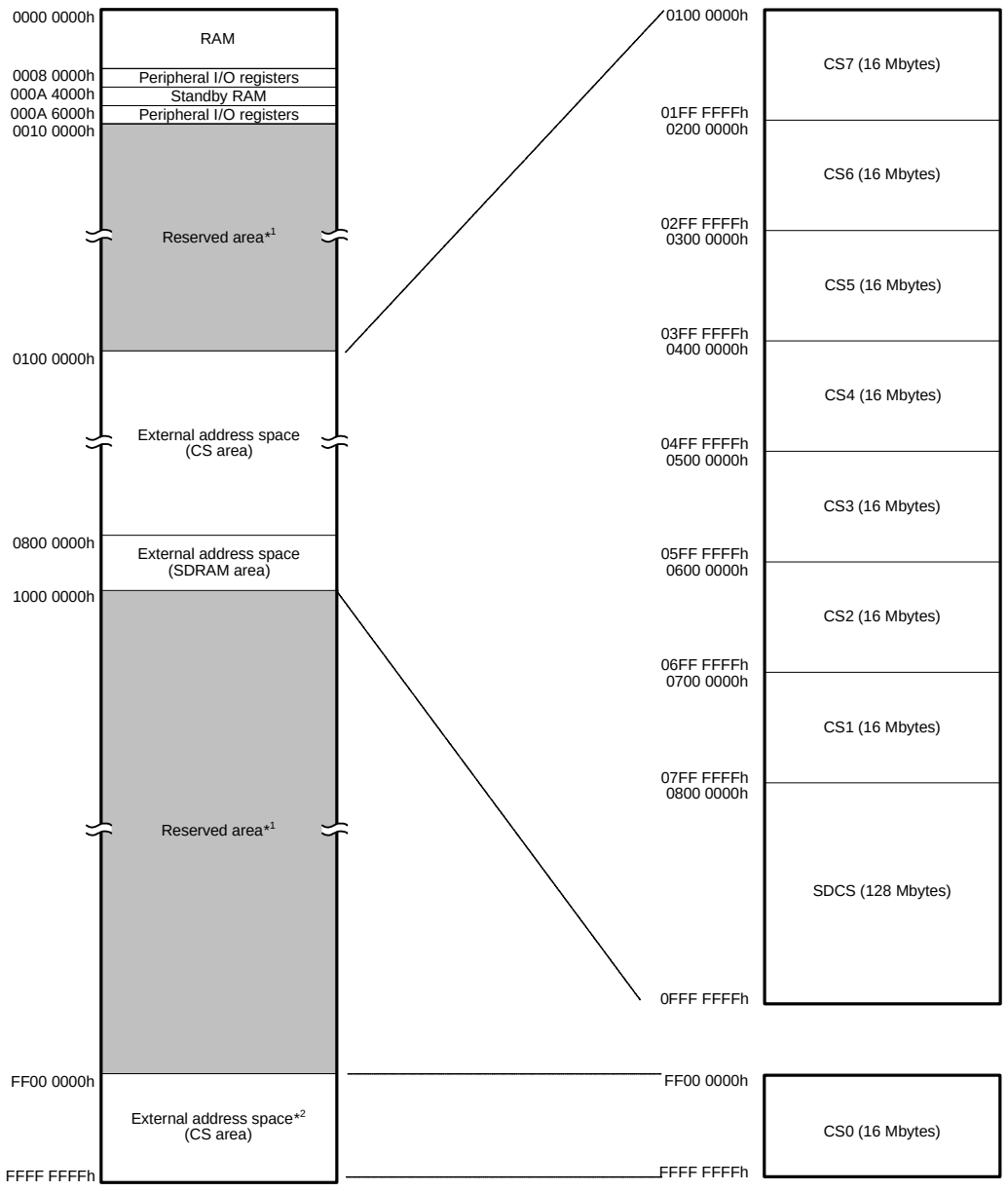
Code Flash Memory Capacity	Address	Data Flash Memory Capacity	Address	RAM Capacity	Address
4 Mbytes	FFC0 0000h to FFFF FFFFh	64 Kbytes	0010 0000h to 0010 FFFFh	512 Kbytes	0000 0000h to 0007 FFFFh
3 Mbytes	FFD0 0000h to FFFF FFFFh				
2.5 Mbytes	FFD8 0000h to FFFF FFFFh				
2 Mbytes	FFE0 0000h to FFFF FFFFh				

Note 3. Reserved areas should not be accessed.
Note 4. The FCU-RAM and the on-chip ROM (FCU firmware) are reserved in products that do not include the FCU-RAM. For details on the FCU, see section 63, Flash Memory.

Figure 4.1 Memory Map in Each Operating Mode

4.2 External Address Space

The external address space is divided into CS areas (CS0 to CS7) and SDRAM area (SDCS). The CS areas are divided into up to eight areas (CS0 to CS7), each corresponding to the CSn# signal output from a CSn# (n = 0 to 7) pin. Figure 4.2 shows the address ranges corresponding to the individual CS areas (CS0 to CS7) and SDRAM areas (SDCS) in on-chip ROM disabled extended mode.



Note 1. Reserved areas should not be accessed.
Note 2. The CS0 area is disabled in on-chip ROM enabled extended mode.
In this mode, the address space for addresses above 1000 0000h is as shown in figure on this section, Memory Map in Each Operating Mode.

Figure 4.2 Correspondence between External Address Spaces and CS Areas (In On-Chip ROM Disabled Extended Mode)

5. I/O Registers

This section gives information on the on-chip I/O register addresses. The information is given as shown below. Notes on writing to registers are also given at the end.

(1) I/O register addresses (address order)

- Registers are listed from the lower allocation addresses.
- Registers are classified according to module symbols.
- The number of access cycles indicates the number of cycles based on the specified reference clock.
- Among the internal I/O register area, addresses not listed in the list of registers are reserved. Reserved addresses must not be accessed. Do not access these addresses; otherwise, the operation when accessing these bits and subsequent operations cannot be guaranteed.

(2) Notes on writing to I/O registers

When writing to an I/O register, the CPU starts executing the subsequent instruction before completing I/O register write. This may cause the subsequent instruction to be executed before the post-update I/O register value is reflected on the operation.
As described in the following examples, special care is required for the cases in which the subsequent instruction must be executed after the post-update I/O register value is actually reflected.

[Examples of cases requiring special care]

- The subsequent instruction must be executed while an interrupt request is disabled with the IENj bit in IERn of the ICU (interrupt request enable bit) set to 0.
- A WAIT instruction is executed immediately after the preprocessing for causing a transition to the low power consumption state.

In the above cases, after writing to an I/O register, wait until the write operation is completed using the following procedure and then execute the subsequent instruction.

(a) Write to an I/O register.
(b) Read the value from the I/O register to a general register.
(c) Execute the operation using the value read.
(d) Execute the subsequent instruction.

[Instruction examples]

- Byte-size I/O registers

```
MOV.L  #SFR_ADDR, R1
MOV.B  #SFR_DATA, [R1]
CMP    [R1].UB, R1
;; Next process
```

- Word-size I/O registers

```
MOV.L  #SFR_ADDR, R1
MOV.W  #SFR_DATA, [R1]
CMP    [R1].W, R1
;; Next process
```

- Longword-size I/O registers

```
MOV.L  #SFR_ADDR, R1
MOV.L  #SFR_DATA, [R1]
CMP    [R1].L, R1
;; Next process
```

If multiple registers are written to and a subsequent instruction should be executed after the write operations are entirely completed, only read the I/O register that was last written to and execute the operation using the value; it is not necessary to read or execute operation for all the registers that were written to.

(3) Number of Access Cycles to I/O Registers

For the number of I/O register access cycles, refer to Table 5.1, List of I/O Registers (Address Order).
The number of access cycles to I/O registers is obtained by following equation.*1

Number of access cycles to I/O registers = Number of bus cycles for internal main bus 1 +
Number of divided clock synchronization cycles +
Number of bus cycles for internal peripheral busses 1 to 6

The number of bus cycles of internal peripheral bus 1 to 6 differs according to the register to be accessed.
When peripheral functions connected to internal peripheral bus 2 to 6 or registers for the external bus control unit (except for bus error related registers) are accessed, the number of divided clock synchronization cycles is added.
The number of divided clock synchronization cycles differs depending on the frequency ratio between ICLK and PCLK (or FCLK, BCLK) or bus access timing.
In the peripheral function unit, when the frequency ratio of ICLK is equal to or greater than that of PCLK (or FCLK), the sum of the number of bus cycles for internal main bus 1 and the number of the divided clock synchronization cycles will be one cycle of PCLK (or FCLK) at a maximum. Therefore, one PCLK (or FCLK) has been added to the number of access states shown in Table 5.1.
When the frequency ratio of ICLK is lower than that of PCLK (or FCLK), the subsequent bus access is started from the ICLK cycle following the completion of the access to the peripheral functions. Therefore, the access cycles are described on an ICLK basis.
In the external bus control unit, the sum of the number of bus cycles for internal main bus 1 and the number of divided clock synchronization cycles will be one cycle of BCLK at a maximum. Therefore, one BCLK is added to the number of access cycles shown in Table 5.1.

Note 1. This applies to the number of cycles when the access from the CPU does not conflict with the instruction fetching to the external memory or bus access from the different bus master (DMAC or DTC).

(4) Notes on Sleep Mode and Mode Transitions

During sleep mode or mode transitions, do not write to the registers related to system control (indicated by 'SYSTEM' in the Module Symbol column in Table 5.1, List of I/O Registers (Address Order)).

(5) Restrictions in Relation to RMPA and String-Manipulation Instructions

The allocation of data to be handled by RMPA or string-manipulation instructions to I/O registers is prohibited, and operation is not guaranteed if this restriction is not observed.

5.1 I/O Register Addresses (Address Order)

Table 5.1 List of I/O Registers (Address Order) (1 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 0000h	SYSTEM	Mode Monitor Register	MDMONR	16	16	3 ICLK		Operating Modes	161
0008 0002h	SYSTEM	Mode Status Register	MDSR	16	16	3 ICLK		Operating Modes	161
0008 0006h	SYSTEM	System Control Register 0	SYSCR0	16	16	3 ICLK		Operating Modes	162
0008 0008h	SYSTEM	System Control Register 1	SYSCR1	16	16	3 ICLK		Operating Modes	163
0008 000Ch	SYSTEM	Standby Control Register	SBYCR	16	16	3 ICLK		Low Power Consumption	349
0008 0010h	SYSTEM	Module Stop Control Register A	MSTPCRA	32	32	3 ICLK		Low Power Consumption	350
0008 0014h	SYSTEM	Module Stop Control Register B	MSTPCRB	32	32	3 ICLK		Low Power Consumption	352
0008 0018h	SYSTEM	Module Stop Control Register C	MSTPCRC	32	32	3 ICLK		Low Power Consumption	354
0008 001Ch	SYSTEM	Module Stop Control Register D	MSTPCRD	32	32	3 ICLK		Low Power Consumption	355
0008 0020h	SYSTEM	System Clock Control Register	SCKCR	32	32	3 ICLK		Clock Generation Circuit	295
0008 0024h	SYSTEM	System Clock Control Register 2	SCKCR2	16	16	3 ICLK		Clock Generation Circuit	297
0008 0026h	SYSTEM	System Clock Control Register 3	SCKCR3	16	16	3 ICLK		Clock Generation Circuit	298
0008 0028h	SYSTEM	PLL Control Register	PLLCR	16	16	3 ICLK		Clock Generation Circuit	299
0008 002Ah	SYSTEM	PLL Control Register 2	PLLCR2	8	8	3 ICLK		Clock Generation Circuit	300
0008 0030h	SYSTEM	External Bus Clock Control Register	BCKCR	8	8	3 ICLK		Clock Generation Circuit	301
0008 0032h	SYSTEM	Main Clock Oscillator Control Register	MOSCCR	8	8	3 ICLK		Clock Generation Circuit	302
0008 0033h	SYSTEM	Sub-Clock Oscillator Control Register	SOSCCR	8	8	3 ICLK		Clock Generation Circuit	304
0008 0034h	SYSTEM	Low-Speed On-Chip Oscillator Control Register	LOCOCR	8	8	3 ICLK		Clock Generation Circuit	305

Table 5.1 List of I/O Registers (Address Order) (2 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 0035h	SYSTEM	IWDT-Dedicated On-Chip Oscillator Control Register	ILOCOCR	8	8	3 ICLK		Clock Generation Circuit	306
0008 0036h	SYSTEM	High-Speed On-Chip Oscillator Control Register	HOCOCR	8	8	3 ICLK		Clock Generation Circuit	307
0008 0037h	SYSTEM	High-Speed On-Chip Oscillator Control Register 2	HOCOCR2	8	8	3 ICLK		Clock Generation Circuit	308
0008 003Ch	SYSTEM	Oscillation Stabilization Flag Register	OSCOVFSR	8	8	3 ICLK		Clock Generation Circuit	309
0008 0040h	SYSTEM	Oscillation Stop Detection Control Register	OSTDCR	8	8	3 ICLK		Clock Generation Circuit	311
0008 0041h	SYSTEM	Oscillation Stop Detection Status Register	OSTDSR	8	8	3 ICLK		Clock Generation Circuit	312
0008 00A0h	SYSTEM	Operating Power Control Register	OPCCR	8	8	3 ICLK		Low Power Consumption	357
0008 00A1h	SYSTEM	Sleep Mode Return Clock Source Switching Register	RSTCKCR	8	8	3 ICLK		Low Power Consumption	360
0008 00A2h	SYSTEM	Main Clock Oscillator Wait Control Register	MOSCWTCR	8	8	3 ICLK		Clock Generation Circuit	313
0008 00A3h	SYSTEM	Sub-Clock Oscillator Wait Control Register	SOSCWTCR	8	8	3 ICLK		Clock Generation Circuit	314
0008 00C0h	SYSTEM	Reset Status Register 2	RSTSR2	8	8	3 ICLK		Resets	246
0008 00C2h	SYSTEM	Software Reset Register	SWRR	16	16	3 ICLK		Resets	247
0008 00E0h	SYSTEM	Voltage Monitoring 1 Circuit Control Register 1	LVD1CR1	8	8	3 ICLK		LVDA	274
0008 00E1h	SYSTEM	Voltage Monitoring 1 Circuit Status Register	LVD1SR	8	8	3 ICLK		LVDA	275
0008 00E2h	SYSTEM	Voltage Monitoring 2 Circuit Control Register 1	LVD2CR1	8	8	3 ICLK		LVDA	275
0008 00E3h	SYSTEM	Voltage Monitoring 2 Circuit Status Register	LVD2SR	8	8	3 ICLK		LVDA	276
0008 03FEh	SYSTEM	Protect Register	PRCR	16	16	3 ICLK		Register Write Protection Function	398
0008 1200h	RAM	RAM Operating Mode Control Register	RAMMODE	8	8	2 ICLK		RAM	2739
0008 1201h	RAM	RAM Error Status Register	RAMSTS	8	8	2 ICLK		RAM	2740
0008 1204h	RAM	RAM Protection Register	RAMPRCR	8	8	2 ICLK		RAM	2741
0008 1208h	RAM	RAM Error Address Capture Register	RAMECAD	32	32	2 ICLK		RAM	2740
0008 12C0h	ECCRAM	ECCRAM Operating Mode Control Register	ECCRAMMODE	8	8	2 ICLK		RAM	2735
0008 12C1h	ECCRAM	ECCRAM 2-Bit Error Status Register	ECCRAM2STS	8	8	2 ICLK		RAM	2735
0008 12C2h	ECCRAM	ECCRAM 1-Bit Error Information Update Enable Register	ECCRAM1STSEN	8	8	2 ICLK		RAM	2736

Table 5.1 List of I/O Registers (Address Order) (3 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 12C3h	ECCRAM	ECCRAM 1-Bit Error Status Register	ECCRAM1STS	8	8	2 ICLK		RAM	2736
0008 12C4h	ECCRAM	ECCRAM Protection Register	ECCRAMPRCR	8	8	2 ICLK		RAM	2737
0008 12C8h	ECCRAM	ECCRAM 2-Bit Error Address Capture Register	ECCRAM2ECAD	32	32	2 ICLK		RAM	2737
0008 12CCh	ECCRAM	ECCRAM 1-Bit Error Address Capture Register	ECCRAM1ECAD	32	32	2 ICLK		RAM	2738
0008 12D0h	ECCRAM	ECCRAM Protection Register 2	ECCRAMPRCR2	8	8	2 ICLK		RAM	2738
0008 12D4h	ECCRAM	ECCRAM Test Control Register	ECCRAMETST	8	8	2 ICLK		RAM	2739
0008 1300h	BSC	Bus Error Status Clear Register	BERCLR	8	8	2 ICLK		Buses	513
0008 1304h	BSC	Bus Error Monitoring Enable Register	BEREN	8	8	2 ICLK		Buses	513
0008 1308h	BSC	Bus Error Status Register 1	BERSR1	8	8	2 ICLK		Buses	514
0008 130Ah	BSC	Bus Error Status Register 2	BERSR2	16	16	2 ICLK		Buses	514
0008 1310h	BSC	Bus Priority Control Register	BUSPRI	16	16	2 ICLK		Buses	515
0008 2000h	DMAC0	DMA Source Address Register	DMSAR	32	32	2 ICLK		DMACAa	615
0008 2004h	DMAC0	DMA Destination Address Register	DMDAR	32	32	2 ICLK		DMACAa	615
0008 2008h	DMAC0	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		DMACAa	616
0008 200Ch	DMAC0	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		DMACAa	617
0008 2010h	DMAC0	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		DMACAa	618
0008 2013h	DMAC0	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		DMACAa	619
0008 2014h	DMAC0	DMA Address Mode Register	DMAMD	16	16	2 ICLK		DMACAa	621
0008 2018h	DMAC0	DMA Offset Register	DMOFR	32	32	2 ICLK		DMACAa	624
0008 201Ch	DMAC0	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		DMACAa	624
0008 201Dh	DMAC0	DMA Software Start Register	DMREQ	8	8	2 ICLK		DMACAa	625
0008 201Eh	DMAC0	DMA Status Register	DMSTS	8	8	2 ICLK		DMACAa	626
0008 201Fh	DMAC0	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		DMACAa	627
0008 2040h	DMAC1	DMA Source Address Register	DMSAR	32	32	2 ICLK		DMACAa	615
0008 2044h	DMAC1	DMA Destination Address Register	DMDAR	32	32	2 ICLK		DMACAa	615
0008 2048h	DMAC1	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		DMACAa	616
0008 204Ch	DMAC1	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		DMACAa	617
0008 2050h	DMAC1	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		DMACAa	618
0008 2053h	DMAC1	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		DMACAa	619
0008 2054h	DMAC1	DMA Address Mode Register	DMAMD	16	16	2 ICLK		DMACAa	621
0008 205Ch	DMAC1	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		DMACAa	624
0008 205Dh	DMAC1	DMA Software Start Register	DMREQ	8	8	2 ICLK		DMACAa	625
0008 205Eh	DMAC1	DMA Status Register	DMSTS	8	8	2 ICLK		DMACAa	626
0008 205Fh	DMAC1	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		DMACAa	627
0008 2080h	DMAC2	DMA Source Address Register	DMSAR	32	32	2 ICLK		DMACAa	615
0008 2084h	DMAC2	DMA Destination Address Register	DMDAR	32	32	2 ICLK		DMACAa	615
0008 2088h	DMAC2	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		DMACAa	616
0008 208Ch	DMAC2	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		DMACAa	617
0008 2090h	DMAC2	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		DMACAa	618
0008 2093h	DMAC2	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		DMACAa	619
0008 2094h	DMAC2	DMA Address Mode Register	DMAMD	16	16	2 ICLK		DMACAa	621
0008 209Ch	DMAC2	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		DMACAa	624
0008 209Dh	DMAC2	DMA Software Start Register	DMREQ	8	8	2 ICLK		DMACAa	625
0008 209Eh	DMAC2	DMA Status Register	DMSTS	8	8	2 ICLK		DMACAa	626
0008 209Fh	DMAC2	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		DMACAa	627
0008 20C0h	DMAC3	DMA Source Address Register	DMSAR	32	32	2 ICLK		DMACAa	615

Table 5.1 List of I/O Registers (Address Order) (4 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 20C4h	DMAC3	DMA Destination Address Register	DMDAR	32	32	2 ICLK		DMACAa	615
0008 20C8h	DMAC3	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		DMACAa	616
0008 20CCh	DMAC3	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		DMACAa	617
0008 20D0h	DMAC3	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		DMACAa	618
0008 20D3h	DMAC3	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		DMACAa	619
0008 20D4h	DMAC3	DMA Address Mode Register	DMAMD	16	16	2 ICLK		DMACAa	621
0008 20DCh	DMAC3	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		DMACAa	624
0008 20DDh	DMAC3	DMA Software Start Register	DMREQ	8	8	2 ICLK		DMACAa	625
0008 20DEh	DMAC3	DMA Status Register	DMSTS	8	8	2 ICLK		DMACAa	626
0008 20DFh	DMAC3	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		DMACAa	627
0008 2100h	DMAC4	DMA Source Address Register	DMSAR	32	32	2 ICLK		DMACAa	615
0008 2104h	DMAC4	DMA Destination Address Register	DMDAR	32	32	2 ICLK		DMACAa	615
0008 2108h	DMAC4	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		DMACAa	616
0008 210Ch	DMAC4	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		DMACAa	617
0008 2110h	DMAC4	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		DMACAa	618
0008 2113h	DMAC4	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		DMACAa	619
0008 2114h	DMAC4	DMA Address Mode Register	DMAMD	16	16	2 ICLK		DMACAa	621
0008 211Ch	DMAC4	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		DMACAa	624
0008 211Dh	DMAC4	DMA Software Start Register	DMREQ	8	8	2 ICLK		DMACAa	625
0008 211Eh	DMAC4	DMA Status Register	DMSTS	8	8	2 ICLK		DMACAa	626
0008 211Fh	DMAC4	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		DMACAa	627
0008 2140h	DMAC5	DMA Source Address Register	DMSAR	32	32	2 ICLK		DMACAa	615
0008 2144h	DMAC5	DMA Destination Address Register	DMDAR	32	32	2 ICLK		DMACAa	615
0008 2148h	DMAC5	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		DMACAa	616
0008 214Ch	DMAC5	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		DMACAa	617
0008 2150h	DMAC5	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		DMACAa	618
0008 2153h	DMAC5	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		DMACAa	619
0008 2154h	DMAC5	DMA Address Mode Register	DMAMD	16	16	2 ICLK		DMACAa	621
0008 215Ch	DMAC5	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		DMACAa	624
0008 215Dh	DMAC5	DMA Software Start Register	DMREQ	8	8	2 ICLK		DMACAa	625
0008 215Eh	DMAC5	DMA Status Register	DMSTS	8	8	2 ICLK		DMACAa	626
0008 215Fh	DMAC5	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		DMACAa	627
0008 2180h	DMAC6	DMA Source Address Register	DMSAR	32	32	2 ICLK		DMACAa	615
0008 2184h	DMAC6	DMA Destination Address Register	DMDAR	32	32	2 ICLK		DMACAa	615
0008 2188h	DMAC6	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		DMACAa	616
0008 218Ch	DMAC6	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		DMACAa	617
0008 2190h	DMAC6	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		DMACAa	618
0008 2193h	DMAC6	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		DMACAa	619
0008 2194h	DMAC6	DMA Address Mode Register	DMAMD	16	16	2 ICLK		DMACAa	621
0008 219Ch	DMAC6	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		DMACAa	624
0008 219Dh	DMAC6	DMA Software Start Register	DMREQ	8	8	2 ICLK		DMACAa	625
0008 219Eh	DMAC6	DMA Status Register	DMSTS	8	8	2 ICLK		DMACAa	626
0008 219Fh	DMAC6	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		DMACAa	627
0008 21C0h	DMAC7	DMA Source Address Register	DMSAR	32	32	2 ICLK		DMACAa	615
0008 21C4h	DMAC7	DMA Destination Address Register	DMDAR	32	32	2 ICLK		DMACAa	615
0008 21C8h	DMAC7	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		DMACAa	616
0008 21CCh	DMAC7	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		DMACAa	617
0008 21D0h	DMAC7	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		DMACAa	618
0008 21D3h	DMAC7	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		DMACAa	619
0008 21D4h	DMAC7	DMA Address Mode Register	DMAMD	16	16	2 ICLK		DMACAa	621

Table 5.1 List of I/O Registers (Address Order) (5 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 21DCh	DMAC7	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		DMACAa	624
0008 21DDh	DMAC7	DMA Software Start Register	DMREQ	8	8	2 ICLK		DMACAa	625
0008 21DEh	DMAC7	DMA Status Register	DMSTS	8	8	2 ICLK		DMACAa	626
0008 21DFh	DMAC7	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		DMACAa	627
0008 2200h	DMAC	DMAC Module Start Register	DMAST	8	8	2 ICLK		DMACAa	628
0008 2204h	DMAC	DMAC74 Interrupt Status Monitor Register	DMIST	8	8	2 ICLK		DMACAa	629
0008 2400h	DTC	DTC Control Register	DTCCR	8	8	2 ICLK		DTCa	726
0008 2404h	DTC	DTC Vector Base Register	DTCVBR	32	32	2 ICLK		DTCa	727
0008 2408h	DTC	DTC Address Mode Register	DTCADMOD	8	8	2 ICLK		DTCa	727
0008 240Ch	DTC	DTC Module Start Register	DTCST	8	8	2 ICLK		DTCa	728
0008 240Eh	DTC	DTC Status Register	DTCSTS	16	16	2 ICLK		DTCa	729
0008 2800h	EXDMAC0	EXDMA Source Address Register	EDMSAR	32	32	1, 2 BCLK		EXDMACa	655
0008 2804h	EXDMAC0	EXDMA Destination Address Register	EDMDAR	32	32	1, 2 BCLK		EXDMACa	655
0008 2808h	EXDMAC0	EXDMA Transfer Count Register	EDMCRA	32	32	1, 2 BCLK		EXDMACa	656
0008 280Ch	EXDMAC0	EXDMA Block Transfer Count Register	EDMCRB	16	16	1, 2 BCLK		EXDMACa	658
0008 2810h	EXDMAC0	EXDMA Transfer Mode Register	EDMTMD	16	16	1, 2 BCLK		EXDMACa	659
0008 2812h	EXDMAC0	EXDMA Output Setting Register	EDMOMD	8	8	1, 2 BCLK		EXDMACa	660
0008 2813h	EXDMAC0	EXDMA Interrupt Setting Register	EDMINT	8	8	1, 2 BCLK		EXDMACa	661
0008 2814h	EXDMAC0	EXDMA Address Mode Register	EDMAMD	32	32	1, 2 BCLK		EXDMACa	663
0008 2818h	EXDMAC0	EXDMA Offset Register	EDMOFR	32	32	1, 2 BCLK		EXDMACa	666
0008 281Ch	EXDMAC0	EXDMA Transfer Enable Register	EDMCNT	8	8	1, 2 BCLK		EXDMACa	667
0008 281Dh	EXDMAC0	EXDMA Software Start Register	EDMREQ	8	8	1, 2 BCLK		EXDMACa	668
0008 281Eh	EXDMAC0	EXDMA Status Register	EDMSTS	8	8	1, 2 BCLK		EXDMACa	669
0008 2820h	EXDMAC0	EXDMA External Request Sense Mode Register	EDMRMD	8	8	1, 2 BCLK		EXDMACa	670
0008 2821h	EXDMAC0	EXDMA External Request Flag Register	EDMERF	8	8	1, 2 BCLK		EXDMACa	671
0008 2822h	EXDMAC0	EXDMA Peripheral Request Flag Register	EDMPRF	8	8	1, 2 BCLK		EXDMACa	672
0008 2840h	EXDMAC1	EXDMA Source Address Register	EDMSAR	32	32	1, 2 BCLK		EXDMACa	655
0008 2844h	EXDMAC1	EXDMA Destination Address Register	EDMDAR	32	32	1, 2 BCLK		EXDMACa	655
0008 2848h	EXDMAC1	EXDMA Transfer Count Register	EDMCRA	32	32	1, 2 BCLK		EXDMACa	656
0008 284Ch	EXDMAC1	EXDMA Block Transfer Count Register	EDMCRB	16	16	1, 2 BCLK		EXDMACa	658
0008 2850h	EXDMAC1	EXDMA Transfer Mode Register	EDMTMD	16	16	1, 2 BCLK		EXDMACa	659
0008 2852h	EXDMAC1	EXDMA Output Setting Register	EDMOMD	8	8	1, 2 BCLK		EXDMACa	660
0008 2853h	EXDMAC1	EXDMA Interrupt Setting Register	EDMINT	8	8	1, 2 BCLK		EXDMACa	661
0008 2854h	EXDMAC1	EXDMA Address Mode Register	EDMAMD	32	32	1, 2 BCLK		EXDMACa	663
0008 285Ch	EXDMAC1	EXDMA Transfer Enable Register	EDMCNT	8	8	1, 2 BCLK		EXDMACa	667

Table 5.1 List of I/O Registers (Address Order) (6 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 285Dh	EXDMAC1	EXDMA Software Start Register	EDMREQ	8	8	1, 2 BCLK		EXDMACa	668
0008 285Eh	EXDMAC1	EXDMA Status Register	EDMSTS	8	8	1, 2 BCLK		EXDMACa	669
0008 2860h	EXDMAC1	EXDMA External Request Sense Mode Register	EDMRMD	8	8	1, 2 BCLK		EXDMACa	670
0008 2861h	EXDMAC1	EXDMA External Request Flag Register	EDMERF	8	8	1, 2 BCLK		EXDMACa	671
0008 2862h	EXDMAC1	EXDMA Peripheral Request Flag Register	EDMPRF	8	8	1, 2 BCLK		EXDMACa	672
0008 2A00h	EXDMAC	EXDMAC Module Start Register	EDMAST	8	8	1, 2 BCLK		EXDMACa	673
0008 2BE0h	EXDMAC	Cluster Buffer Register 0	CLSBR0	32	32	1, 2 BCLK		EXDMACa	674
0008 2BE4h	EXDMAC	Cluster Buffer Register 1	CLSBR1	32	32	1, 2 BCLK		EXDMACa	674
0008 2BE8h	EXDMAC	Cluster Buffer Register 2	CLSBR2	32	32	1, 2 BCLK		EXDMACa	674
0008 2BECh	EXDMAC	Cluster Buffer Register 3	CLSBR3	32	32	1, 2 BCLK		EXDMACa	674
0008 2BF0h	EXDMAC	Cluster Buffer Register 4	CLSBR4	32	32	1, 2 BCLK		EXDMACa	674
0008 2BF4h	EXDMAC	Cluster Buffer Register 5	CLSBR5	32	32	1, 2 BCLK		EXDMACa	674
0008 2BF8h	EXDMAC	Cluster Buffer Register 6	CLSBR6	32	32	1, 2 BCLK		EXDMACa	674
0008 2BFCh	EXDMAC	Cluster Buffer Register 7	CLSBR7	32	32	1, 2 BCLK		EXDMACa	674
0008 3002h	BSC	CS0 Mode Register	CS0MOD	16	16	1, 2 BCLK		Buses	493
0008 3004h	BSC	CS0 Wait Control Register 1	CS0WCR1	32	32	1, 2 BCLK		Buses	495
0008 3008h	BSC	CS0 Wait Control Register 2	CS0WCR2	32	32	1, 2 BCLK		Buses	498
0008 3012h	BSC	CS1 Mode Register	CS1MOD	16	16	1, 2 BCLK		Buses	493
0008 3014h	BSC	CS1 Wait Control Register 1	CS1WCR1	32	32	1, 2 BCLK		Buses	495
0008 3018h	BSC	CS1 Wait Control Register 2	CS1WCR2	32	32	1, 2 BCLK		Buses	498
0008 3022h	BSC	CS2 Mode Register	CS2MOD	16	16	1, 2 BCLK		Buses	493
0008 3024h	BSC	CS2 Wait Control Register 1	CS2WCR1	32	32	1, 2 BCLK		Buses	495
0008 3028h	BSC	CS2 Wait Control Register 2	CS2WCR2	32	32	1, 2 BCLK		Buses	498
0008 3032h	BSC	CS3 Mode Register	CS3MOD	16	16	1, 2 BCLK		Buses	493
0008 3034h	BSC	CS3 Wait Control Register 1	CS3WCR1	32	32	1, 2 BCLK		Buses	495
0008 3038h	BSC	CS3 Wait Control Register 2	CS3WCR2	32	32	1, 2 BCLK		Buses	498
0008 3042h	BSC	CS4 Mode Register	CS4MOD	16	16	1, 2 BCLK		Buses	493
0008 3044h	BSC	CS4 Wait Control Register 1	CS4WCR1	32	32	1, 2 BCLK		Buses	495
0008 3048h	BSC	CS4 Wait Control Register 2	CS4WCR2	32	32	1, 2 BCLK		Buses	498
0008 3052h	BSC	CS5 Mode Register	CS5MOD	16	16	1, 2 BCLK		Buses	493
0008 3054h	BSC	CS5 Wait Control Register 1	CS5WCR1	32	32	1, 2 BCLK		Buses	495
0008 3058h	BSC	CS5 Wait Control Register 2	CS5WCR2	32	32	1, 2 BCLK		Buses	498
0008 3062h	BSC	CS6 Mode Register	CS6MOD	16	16	1, 2 BCLK		Buses	493
0008 3064h	BSC	CS6 Wait Control Register 1	CS6WCR1	32	32	1, 2 BCLK		Buses	495
0008 3068h	BSC	CS6 Wait Control Register 2	CS6WCR2	32	32	1, 2 BCLK		Buses	498
0008 3072h	BSC	CS7 Mode Register	CS7MOD	16	16	1, 2 BCLK		Buses	493
0008 3074h	BSC	CS7 Wait Control Register 1	CS7WCR1	32	32	1, 2 BCLK		Buses	495
0008 3078h	BSC	CS7 Wait Control Register 2	CS7WCR2	32	32	1, 2 BCLK		Buses	498
0008 3802h	BSC	CS0 Control Register	CS0CR	16	16	1, 2 BCLK		Buses	488
0008 380Ah	BSC	CS0 Recovery Cycle Register	CS0REC	16	16	1, 2 BCLK		Buses	489
0008 3812h	BSC	CS1 Control Register	CS1CR	16	16	1, 2 BCLK		Buses	488
0008 381Ah	BSC	CS1 Recovery Cycle Register	CS1REC	16	16	1, 2 BCLK		Buses	489

Table 5.1 List of I/O Registers (Address Order) (7 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 3822h	BSC	CS2 Control Register	CS2CR	16	16	1, 2 BCLK		Buses	488
0008 382Ah	BSC	CS2 Recovery Cycle Register	CS2REC	16	16	1, 2 BCLK		Buses	489
0008 3832h	BSC	CS3 Control Register	CS3CR	16	16	1, 2 BCLK		Buses	488
0008 383Ah	BSC	CS3 Recovery Cycle Register	CS3REC	16	16	1, 2 BCLK		Buses	489
0008 3842h	BSC	CS4 Control Register	CS4CR	16	16	1, 2 BCLK		Buses	488
0008 384Ah	BSC	CS4 Recovery Cycle Register	CS4REC	16	16	1, 2 BCLK		Buses	489
0008 3852h	BSC	CS5 Control Register	CS5CR	16	16	1, 2 BCLK		Buses	488
0008 385Ah	BSC	CS5 Recovery Cycle Register	CS5REC	16	16	1, 2 BCLK		Buses	489
0008 3862h	BSC	CS6 Control Register	CS6CR	16	16	1, 2 BCLK		Buses	488
0008 386Ah	BSC	CS6 Recovery Cycle Register	CS6REC	16	16	1, 2 BCLK		Buses	489
0008 3872h	BSC	CS7 Control Register	CS7CR	16	16	1, 2 BCLK		Buses	488
0008 387Ah	BSC	CS7 Recovery Cycle Register	CS7REC	16	16	1, 2 BCLK		Buses	489
0008 3880h	BSC	CS Recovery Cycle Insertion Enable Register	CSRECEN	16	16	1, 2 BCLK		Buses	491
0008 3C00h	BSC	SDC Control Register	SDCCR	8	8	1, 2 BCLK		Buses	501
0008 3C01h	BSC	SDC Mode Register	SDCMOD	8	8	1, 2 BCLK		Buses	502
0008 3C02h	BSC	SDRAM Access Mode Register	SDAMOD	8	8	1, 2 BCLK		Buses	503
0008 3C10h	BSC	SDRAM Self-Refresh Control Register	SDSELF	8	8	1, 2 BCLK		Buses	503
0008 3C14h	BSC	SDRAM Refresh Control Register	SDRFCR	16	16	1, 2 BCLK		Buses	504
0008 3C16h	BSC	SDRAM Auto-Refresh Control Register	SDRFEN	8	8	1, 2 BCLK		Buses	505
0008 3C20h	BSC	SDRAM Initialization Sequence Control Register	SDICR	8	8	1, 2 BCLK		Buses	506
0008 3C24h	BSC	SDRAM Initialization Register	SDIR	16	16	1, 2 BCLK		Buses	507
0008 3C40h	BSC	SDRAM Address Register	SDADR	8	8	1, 2 BCLK		Buses	508
0008 3C44h	BSC	SDRAM Timing Register	SDTR	32	32	1, 2 BCLK		Buses	509
0008 3C48h	BSC	SDRAM Mode Register	SDMOD	16	16	1, 2 BCLK		Buses	511
0008 3C50h	BSC	SDRAM Status Register	SDSR	8	8	1, 2 BCLK		Buses	512
0008 6400h	MPU	Region-0 Start Page Number Register	RSPAGE0	32	32	1 ICLK		MPU	595
0008 6404h	MPU	Region-0 End Page Number Register	REPAGE0	32	32	1 ICLK		MPU	596
0008 6408h	MPU	Region-1 Start Page Number Register	RSPAGE1	32	32	1 ICLK		MPU	595
0008 640Ch	MPU	Region-1 End Page Number Register	REPAGE1	32	32	1 ICLK		MPU	596
0008 6410h	MPU	Region-2 Start Page Number Register	RSPAGE2	32	32	1 ICLK		MPU	595
0008 6414h	MPU	Region-2 End Page Number Register	REPAGE2	32	32	1 ICLK		MPU	596
0008 6418h	MPU	Region-3 Start Page Number Register	RSPAGE3	32	32	1 ICLK		MPU	595
0008 641Ch	MPU	Region-3 End Page Number Register	REPAGE3	32	32	1 ICLK		MPU	596
0008 6420h	MPU	Region-4 Start Page Number Register	RSPAGE4	32	32	1 ICLK		MPU	595
0008 6424h	MPU	Region-4 End Page Number Register	REPAGE4	32	32	1 ICLK		MPU	596
0008 6428h	MPU	Region-5 Start Page Number Register	RSPAGE5	32	32	1 ICLK		MPU	595
0008 642Ch	MPU	Region-5 End Page Number Register	REPAGE5	32	32	1 ICLK		MPU	596
0008 6430h	MPU	Region-6 Start Page Number Register	RSPAGE6	32	32	1 ICLK		MPU	595
0008 6434h	MPU	Region-6 End Page Number Register	REPAGE6	32	32	1 ICLK		MPU	596
0008 6438h	MPU	Region-7 Start Page Number Register	RSPAGE7	32	32	1 ICLK		MPU	595
0008 643Ch	MPU	Region-7 End Page Number Register	REPAGE7	32	32	1 ICLK		MPU	596
0008 6500h	MPU	Memory-Protection Enable Register	MPEN	32	32	1 ICLK		MPU	597
0008 6504h	MPU	Background Access Control Register	MPBAC	32	32	1 ICLK		MPU	598
0008 6508h	MPU	Memory-Protection Error Status-Clearing Register	MPECLR	32	32	1 ICLK		MPU	599
0008 650Ch	MPU	Memory-Protection Error Status Register	MPESTS	32	32	1 ICLK		MPU	600
0008 6514h	MPU	Data Memory-Protection Error Address Register	MPDEA	32	32	1 ICLK		MPU	601
0008 6520h	MPU	Region Search Address Register	MPSA	32	32	1 ICLK		MPU	602
0008 6524h	MPU	Region Search Operation Register	MPOPS	16	16	1 ICLK		MPU	602
0008 6526h	MPU	Region Invalidation Operation Register	MPOPI	16	16	1 ICLK		MPU	603
0008 6528h	MPU	Instruction-Hit Region Register	MHITI	32	32	1 ICLK		MPU	604

Table 5.1 List of I/O Registers (Address Order) (8 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 652Ch	MPU	Data-Hit Region Register	MHITD	32	32	1 ICLK		MPU	606
0008 7010h to 0008 70FFh	ICU	Interrupt Request Registers 016 to 255	IR016 to 255	8	8	2 ICLK		ICUA	411
0008 711Ah to 0008 71FFh	ICU	DTC Transfer Request Enable Registers 026 to 255	DTCER026 to DTCER255	8	8	2 ICLK		ICUA	417
0008 7202h to 0008 721Fh	ICU	Interrupt Request Enable Registers 02 to 1F	IER02 to IER1F	8	8	2 ICLK		ICUA	413
0008 72E0h	ICU	Software Interrupt Generation Register	SWINTR	8	8	2 ICLK		ICUA	416
0008 72E1h	ICU	Software Interrupt 2 Generation Register	SWINT2R	8	8	2 ICLK		ICUA	416
0008 72F0h	ICU	Fast Interrupt Set Register	FIR	16	16	2 ICLK		ICUA	415
0008 7300h to 0008 73FFh	ICU	Interrupt Source Priority Registers 000 to 255	IPR000 to IPR255	8	8	2 ICLK		ICUA	414
0008 7400h	ICU	DMAC Trigger Select Register 0	DMRSR0	8	8	2 ICLK		ICUA	418
0008 7404h	ICU	DMAC Trigger Select Register 1	DMRSR1	8	8	2 ICLK		ICUA	418
0008 7408h	ICU	DMAC Trigger Select Register 2	DMRSR2	8	8	2 ICLK		ICUA	418
0008 740Ch	ICU	DMAC Trigger Select Register 3	DMRSR3	8	8	2 ICLK		ICUA	418
0008 7410h	ICU	DMAC Trigger Select Register 4	DMRSR4	8	8	2 ICLK		ICUA	418
0008 7414h	ICU	DMAC Trigger Select Register 5	DMRSR5	8	8	2 ICLK		ICUA	418
0008 7418h	ICU	DMAC Trigger Select Register 6	DMRSR6	8	8	2 ICLK		ICUA	418
0008 741Ch	ICU	DMAC Trigger Select Register 7	DMRSR7	8	8	2 ICLK		ICUA	418
0008 7500h to 0008 750Fh	ICU	IRQ Control Registers 0 to 15	IRQCR0 to 15	8	8	2 ICLK		ICUA	419
0008 7520h	ICU	IRQ Pin Digital Filter Enable Register 0	IRQFLTE0	8	8	2 ICLK		ICUA	420
0008 7521h	ICU	IRQ Pin Digital Filter Enable Register 1	IRQFLTE1	8	8	2 ICLK		ICUA	421
0008 7528h	ICU	IRQ Pin Digital Filter Setting Register 0	IRQFLTC0	16	16	2 ICLK		ICUA	422
0008 752Ah	ICU	IRQ Pin Digital Filter Setting Register 1	IRQFLTC1	16	16	2 ICLK		ICUA	423
0008 7580h	ICU	Non-Maskable Interrupt Status Register	NMISR	8	8	2 ICLK		ICUA	424
0008 7581h	ICU	Non-Maskable Interrupt Enable Register	NMIER	8	8	2 ICLK		ICUA	426
0008 7582h	ICU	Non-Maskable Interrupt Status Clear Register	NMICLR	8	8	2 ICLK		ICUA	428
0008 7583h	ICU	NMI Pin Interrupt Control Register	NMICR	8	8	2 ICLK		ICUA	428
0008 7590h	ICU	NMI Pin Digital Filter Enable Register	NMIFLTE	8	8	2 ICLK		ICUA	429
0008 7594h	ICU	NMI Pin Digital Filter Setting Register	NMIFLTC	8	8	2 ICLK		ICUA	429
0008 7600h	ICU	Group BE0 Interrupt Request Register	GRPBE0	32	32	2 ICLK to 1 PCLKB	2 ICLK	ICUA	430
0008 7630h	ICU	Group BL0 Interrupt Request Register	GRPBL0	32	32	2 ICLK to 1 PCLKB	2 ICLK	ICUA	430
0008 7634h	ICU	Group BL1 Interrupt Request Register	GRPBL1	32	32	2 ICLK to 1 PCLKB	2 ICLK	ICUA	430
0008 7640h	ICU	Group BE0 Interrupt Request Enable Register	GENBE0	32	32	2 ICLK to 1 PCLKB	2 ICLK	ICUA	432
0008 7670h	ICU	Group BL0 Interrupt Request Enable Register	GENBL0	32	32	2 ICLK to 1 PCLKB	2 ICLK	ICUA	432
0008 7674h	ICU	Group BL1 Interrupt Request Enable Register	GENBL1	32	32	2 ICLK to 1 PCLKB	2 ICLK	ICUA	432
0008 7680h	ICU	Group BE0 Interrupt Clear Register	GCRBE0	32	32	2 ICLK to 1 PCLKB	2 ICLK	ICUA	434
0008 7700h	ICU	Software Configurable Interrupt B Request Register 0	PIBR0	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	436
0008 7701h	ICU	Software Configurable Interrupt B Request Register 1	PIBR1	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	436
0008 7702h	ICU	Software Configurable Interrupt B Request Register 2	PIBR2	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	436
0008 7703h	ICU	Software Configurable Interrupt B Request Register 3	PIBR3	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	436
0008 7704h	ICU	Software Configurable Interrupt B Request Register 4	PIBR4	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	436
0008 7705h	ICU	Software Configurable Interrupt B Request Register 5	PIBR5	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	436
0008 7706h	ICU	Software Configurable Interrupt B Request Register 6	PIBR6	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	436

Table 5.1 List of I/O Registers (Address Order) (9 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 7707h	ICU	Software Configurable Interrupt B Request Register 7	PIBR7	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	436
0008 7708h	ICU	Software Configurable Interrupt B Request Register 8	PIBR8	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	436
0008 7709h	ICU	Software Configurable Interrupt B Request Register 9	PIBR9	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	436
0008 770Ah	ICU	Software Configurable Interrupt B Request Register A	PIBRA	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	436
0008 7780h	ICU	Software Configurable Interrupt B Source Select Register X128	SLIBXR128	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	438
0008 7781h	ICU	Software Configurable Interrupt B Source Select Register X129	SLIBXR129	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	438
0008 7782h	ICU	Software Configurable Interrupt B Source Select Register X130	SLIBXR130	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	438
0008 7783h	ICU	Software Configurable Interrupt B Source Select Register X131	SLIBXR131	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	438
0008 7784h	ICU	Software Configurable Interrupt B Source Select Register X132	SLIBXR132	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	438
0008 7785h	ICU	Software Configurable Interrupt B Source Select Register X133	SLIBXR133	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	438
0008 7786h	ICU	Software Configurable Interrupt B Source Select Register X134	SLIBXR134	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	438
0008 7787h	ICU	Software Configurable Interrupt B Source Select Register X135	SLIBXR135	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	438
0008 7788h	ICU	Software Configurable Interrupt B Source Select Register X136	SLIBXR136	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	438
0008 7789h	ICU	Software Configurable Interrupt B Source Select Register X137	SLIBXR137	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	438
0008 778Ah	ICU	Software Configurable Interrupt B Source Select Register X138	SLIBXR138	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	438
0008 778Bh	ICU	Software Configurable Interrupt B Source Select Register X139	SLIBXR139	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	438
0008 778Ch	ICU	Software Configurable Interrupt B Source Select Register X140	SLIBXR140	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	438
0008 778Dh	ICU	Software Configurable Interrupt B Source Select Register X141	SLIBXR141	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	438
0008 778Eh	ICU	Software Configurable Interrupt B Source Select Register X142	SLIBXR142	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	438
0008 778Fh	ICU	Software Configurable Interrupt B Source Select Register X143	SLIBXR143	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	438
0008 7790h	ICU	Software Configurable Interrupt B Source Select Register 144	SLIBR144	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 7791h	ICU	Software Configurable Interrupt B Source Select Register 145	SLIBR145	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 7792h	ICU	Software Configurable Interrupt B Source Select Register 146	SLIBR146	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 7793h	ICU	Software Configurable Interrupt B Source Select Register 147	SLIBR147	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 7794h	ICU	Software Configurable Interrupt B Source Select Register 148	SLIBR148	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 7795h	ICU	Software Configurable Interrupt B Source Select Register 149	SLIBR149	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 7796h	ICU	Software Configurable Interrupt B Source Select Register 150	SLIBR150	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 7797h	ICU	Software Configurable Interrupt B Source Select Register 151	SLIBR151	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 7798h	ICU	Software Configurable Interrupt B Source Select Register 152	SLIBR152	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 7799h	ICU	Software Configurable Interrupt B Source Select Register 153	SLIBR153	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 779Ah	ICU	Software Configurable Interrupt B Source Select Register 154	SLIBR154	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 779Bh	ICU	Software Configurable Interrupt B Source Select Register 155	SLIBR155	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439

Table 5.1 List of I/O Registers (Address Order) (10 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 779Ch	ICU	Software Configurable Interrupt B Source Select Register 156	SLIBR156	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 779Dh	ICU	Software Configurable Interrupt B Source Select Register 157	SLIBR157	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 779Eh	ICU	Software Configurable Interrupt B Source Select Register 158	SLIBR158	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 779Fh	ICU	Software Configurable Interrupt B Source Select Register 159	SLIBR159	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77A0h	ICU	Software Configurable Interrupt B Source Select Register 160	SLIBR160	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77A1h	ICU	Software Configurable Interrupt B Source Select Register 161	SLIBR161	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77A2h	ICU	Software Configurable Interrupt B Source Select Register 162	SLIBR162	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77A3h	ICU	Software Configurable Interrupt B Source Select Register 163	SLIBR163	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77A4h	ICU	Software Configurable Interrupt B Source Select Register 164	SLIBR164	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77A5h	ICU	Software Configurable Interrupt B Source Select Register 165	SLIBR165	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77A6h	ICU	Software Configurable Interrupt B Source Select Register 166	SLIBR166	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77A7h	ICU	Software Configurable Interrupt B Source Select Register 167	SLIBR167	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77A8h	ICU	Software Configurable Interrupt B Source Select Register 168	SLIBR168	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77A9h	ICU	Software Configurable Interrupt B Source Select Register 169	SLIBR169	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77AAh	ICU	Software Configurable Interrupt B Source Select Register 170	SLIBR170	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77ABh	ICU	Software Configurable Interrupt B Source Select Register 171	SLIBR171	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77ACh	ICU	Software Configurable Interrupt B Source Select Register 172	SLIBR172	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77ADh	ICU	Software Configurable Interrupt B Source Select Register 173	SLIBR173	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77AEh	ICU	Software Configurable Interrupt B Source Select Register 174	SLIBR174	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77AFh	ICU	Software Configurable Interrupt B Source Select Register 175	SLIBR175	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77B0h	ICU	Software Configurable Interrupt B Source Select Register 176	SLIBR176	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77B1h	ICU	Software Configurable Interrupt B Source Select Register 177	SLIBR177	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77B2h	ICU	Software Configurable Interrupt B Source Select Register 178	SLIBR178	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77B3h	ICU	Software Configurable Interrupt B Source Select Register 179	SLIBR179	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77B4h	ICU	Software Configurable Interrupt B Source Select Register 180	SLIBR180	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77B5h	ICU	Software Configurable Interrupt B Source Select Register 181	SLIBR181	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77B6h	ICU	Software Configurable Interrupt B Source Select Register 182	SLIBR182	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77B7h	ICU	Software Configurable Interrupt B Source Select Register 183	SLIBR183	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77B8h	ICU	Software Configurable Interrupt B Source Select Register 184	SLIBR184	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77B9h	ICU	Software Configurable Interrupt B Source Select Register 185	SLIBR185	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77BAh	ICU	Software Configurable Interrupt B Source Select Register 186	SLIBR186	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439

Table 5.1 List of I/O Registers (Address Order) (11 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 77BBh	ICU	Software Configurable Interrupt B Source Select Register 187	SLIBR187	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77BCh	ICU	Software Configurable Interrupt B Source Select Register 188	SLIBR188	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77BDh	ICU	Software Configurable Interrupt B Source Select Register 189	SLIBR189	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77BEh	ICU	Software Configurable Interrupt B Source Select Register 190	SLIBR190	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77BFh	ICU	Software Configurable Interrupt B Source Select Register 191	SLIBR191	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77C0h	ICU	Software Configurable Interrupt B Source Select Register 192	SLIBR192	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77C1h	ICU	Software Configurable Interrupt B Source Select Register 193	SLIBR193	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77C2h	ICU	Software Configurable Interrupt B Source Select Register 194	SLIBR194	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77C3h	ICU	Software Configurable Interrupt B Source Select Register 195	SLIBR195	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77C4h	ICU	Software Configurable Interrupt B Source Select Register 196	SLIBR196	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77C5h	ICU	Software Configurable Interrupt B Source Select Register 197	SLIBR197	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77C6h	ICU	Software Configurable Interrupt B Source Select Register 198	SLIBR198	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77C7h	ICU	Software Configurable Interrupt B Source Select Register 199	SLIBR199	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77C8h	ICU	Software Configurable Interrupt B Source Select Register 200	SLIBR200	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77C9h	ICU	Software Configurable Interrupt B Source Select Register 201	SLIBR201	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77CAh	ICU	Software Configurable Interrupt B Source Select Register 202	SLIBR202	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77CBh	ICU	Software Configurable Interrupt B Source Select Register 203	SLIBR203	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77CCh	ICU	Software Configurable Interrupt B Source Select Register 204	SLIBR204	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77CDh	ICU	Software Configurable Interrupt B Source Select Register 205	SLIBR205	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77CEh	ICU	Software Configurable Interrupt B Source Select Register 206	SLIBR206	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 77CFh	ICU	Software Configurable Interrupt B Source Select Register 207	SLIBR207	8	8	2 ICLK to 1 PCLKB	2 ICLK	ICUA	439
0008 7830h	ICU	Group AL0 Interrupt Request Register	GRPAL0	32	32	2 ICLK to 1 PCLKA	2 ICLK	ICUA	430
0008 7834h	ICU	Group AL1 Interrupt Request Register	GRPAL1	32	32	2 ICLK to 1 PCLKA	2 ICLK	ICUA	430
0008 7870h	ICU	Group AL0 Interrupt Request Enable Register	GENAL0	32	32	2 ICLK to 1 PCLKA	2 ICLK	ICUA	432
0008 7874h	ICU	Group AL1 Interrupt Request Enable Register	GENAL1	32	32	2 ICLK to 1 PCLKA	2 ICLK	ICUA	432
0008 7900h	ICU	Software Configurable Interrupt A Request Register 0	PIAR0	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	437
0008 7901h	ICU	Software Configurable Interrupt A Request Register 1	PIAR1	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	437
0008 7902h	ICU	Software Configurable Interrupt A Request Register 2	PIAR2	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	437
0008 7903h	ICU	Software Configurable Interrupt A Request Register 3	PIAR3	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	437
0008 7904h	ICU	Software Configurable Interrupt A Request Register 4	PIAR4	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	437
0008 7905h	ICU	Software Configurable Interrupt A Request Register 5	PIAR5	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	437
0008 7906h	ICU	Software Configurable Interrupt A Request Register 6	PIAR6	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	437
0008 7907h	ICU	Software Configurable Interrupt A Request Register 7	PIAR7	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	437

Table 5.1 List of I/O Registers (Address Order) (12 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 7908h	ICU	Software Configurable Interrupt A Request Register 8	PIAR8	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	437
0008 7909h	ICU	Software Configurable Interrupt A Request Register 9	PIAR9	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	437
0008 790Ah	ICU	Software Configurable Interrupt A Request Register A	PIARA	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	437
0008 790Bh	ICU	Software Configurable Interrupt A Request Register B	PIARB	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	437
0008 79D0h	ICU	Software Configurable Interrupt A Source Select Register 208	SLIAR208	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79D1h	ICU	Software Configurable Interrupt A Source Select Register 209	SLIAR209	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79D2h	ICU	Software Configurable Interrupt A Source Select Register 210	SLIAR210	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79D3h	ICU	Software Configurable Interrupt A Source Select Register 211	SLIAR211	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79D4h	ICU	Software Configurable Interrupt A Source Select Register 212	SLIAR212	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79D5h	ICU	Software Configurable Interrupt A Source Select Register 213	SLIAR213	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79D6h	ICU	Software Configurable Interrupt A Source Select Register 214	SLIAR214	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79D7h	ICU	Software Configurable Interrupt A Source Select Register 215	SLIAR215	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79D8h	ICU	Software Configurable Interrupt A Source Select Register 216	SLIAR216	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79D9h	ICU	Software Configurable Interrupt A Source Select Register 217	SLIAR217	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79DAh	ICU	Software Configurable Interrupt A Source Select Register 218	SLIAR218	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79DBh	ICU	Software Configurable Interrupt A Source Select Register 219	SLIAR219	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79DCh	ICU	Software Configurable Interrupt A Source Select Register 220	SLIAR220	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79DDh	ICU	Software Configurable Interrupt A Source Select Register 221	SLIAR221	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79DEh	ICU	Software Configurable Interrupt A Source Select Register 222	SLIAR222	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79DFh	ICU	Software Configurable Interrupt A Source Select Register 223	SLIAR223	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79E0h	ICU	Software Configurable Interrupt A Source Select Register 224	SLIAR224	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79E1h	ICU	Software Configurable Interrupt A Source Select Register 225	SLIAR225	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79E2h	ICU	Software Configurable Interrupt A Source Select Register 226	SLIAR226	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79E3h	ICU	Software Configurable Interrupt A Source Select Register 227	SLIAR227	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79E4h	ICU	Software Configurable Interrupt A Source Select Register 228	SLIAR228	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79E5h	ICU	Software Configurable Interrupt A Source Select Register 229	SLIAR229	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79E6h	ICU	Software Configurable Interrupt A Source Select Register 230	SLIAR230	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79E7h	ICU	Software Configurable Interrupt A Source Select Register 231	SLIAR231	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79E8h	ICU	Software Configurable Interrupt A Source Select Register 232	SLIAR232	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79E9h	ICU	Software Configurable Interrupt A Source Select Register 233	SLIAR233	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79EAh	ICU	Software Configurable Interrupt A Source Select Register 234	SLIAR234	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79EBh	ICU	Software Configurable Interrupt A Source Select Register 235	SLIAR235	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442

Table 5.1 List of I/O Registers (Address Order) (13 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 79ECh	ICU	Software Configurable Interrupt A Source Select Register 236	SLIAR236	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79EDh	ICU	Software Configurable Interrupt A Source Select Register 237	SLIAR237	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79EEh	ICU	Software Configurable Interrupt A Source Select Register 238	SLIAR238	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79EFh	ICU	Software Configurable Interrupt A Source Select Register 239	SLIAR239	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79F0h	ICU	Software Configurable Interrupt A Source Select Register 240	SLIAR240	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79F1h	ICU	Software Configurable Interrupt A Source Select Register 241	SLIAR241	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79F2h	ICU	Software Configurable Interrupt A Source Select Register 242	SLIAR242	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79F3h	ICU	Software Configurable Interrupt A Source Select Register 243	SLIAR243	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79F4h	ICU	Software Configurable Interrupt A Source Select Register 244	SLIAR244	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79F5h	ICU	Software Configurable Interrupt A Source Select Register 245	SLIAR245	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79F6h	ICU	Software Configurable Interrupt A Source Select Register 246	SLIAR246	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79F7h	ICU	Software Configurable Interrupt A Source Select Register 247	SLIAR247	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79F8h	ICU	Software Configurable Interrupt A Source Select Register 248	SLIAR248	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79F9h	ICU	Software Configurable Interrupt A Source Select Register 249	SLIAR249	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79FAh	ICU	Software Configurable Interrupt A Source Select Register 250	SLIAR250	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79FBh	ICU	Software Configurable Interrupt A Source Select Register 251	SLIAR251	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79FCh	ICU	Software Configurable Interrupt A Source Select Register 252	SLIAR252	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79FDh	ICU	Software Configurable Interrupt A Source Select Register 253	SLIAR253	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79FEh	ICU	Software Configurable Interrupt A Source Select Register 254	SLIAR254	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 79FFh	ICU	Software Configurable Interrupt A Source Select Register 255	SLIAR255	8	8	2 ICLK to 1 PCLKA	2 ICLK	ICUA	442
0008 7A00h	ICU	Software Configurable Interrupt Source Select Register Write Protect Register	SLIPRCR	8	8	2 ICLK to 1 PCLKA/B	2 ICLK	ICUA	447
0008 7A01h	ICU	EXDMAC Trigger Select Register	SELEXDR	8	8	2 ICLK to 1 PCLKA/B	2 ICLK	ICUA	446
0008 8000h	CMT	Compare Match Timer Start Register 0	CMSTR0	16	16	2, 3 PCLKB	2 ICLK	CMT	1402
0008 8002h	CMT0	Compare Match Timer Control Register	CMCR	16	16	2, 3 PCLKB	2 ICLK	CMT	1403
0008 8004h	CMT0	Compare Match Counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	CMT	1404
0008 8006h	CMT0	Compare Match Constant Register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	CMT	1404
0008 8008h	CMT1	Compare Match Timer Control Register	CMCR	16	16	2, 3 PCLKB	2 ICLK	CMT	1403
0008 800Ah	CMT1	Compare Match Counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	CMT	1404
0008 800Ch	CMT1	Compare Match Constant Register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	CMT	1404
0008 8010h	CMT	Compare Match Timer Start Register 1	CMSTR1	16	16	2, 3 PCLKB	2 ICLK	CMT	1402
0008 8012h	CMT2	Compare Match Timer Control Register	CMCR	16	16	2, 3 PCLKB	2 ICLK	CMT	1403
0008 8014h	CMT2	Compare Match Counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	CMT	1404
0008 8016h	CMT2	Compare Match Constant Register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	CMT	1404
0008 8018h	CMT3	Compare Match Timer Control Register	CMCR	16	16	2, 3 PCLKB	2 ICLK	CMT	1403
0008 801Ah	CMT3	Compare Match Counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	CMT	1404
0008 801Ch	CMT3	Compare Match Constant Register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	CMT	1404
0008 8020h	WDT	WDT Refresh Register	WDTRR	8	8	2, 3 PCLKB	2 ICLK	WDTA	1487

Table 5.1 List of I/O Registers (Address Order) (14 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 8022h	WDT	WDT Control Register	WDTCR	16	16	2, 3 PCLKB	2 ICLK	WDTA	1488
0008 8024h	WDT	WDT Status Register	WDTSR	16	16	2, 3 PCLKB	2 ICLK	WDTA	1491
0008 8026h	WDT	WDT Reset Control Register	WDTRCR	8	8	2, 3 PCLKB	2 ICLK	WDTA	1492
0008 8030h	IWDT	IWDT Refresh Register	IWDTRR	8	8	2, 3 PCLKB	2 ICLK	IWDTa	1502
0008 8032h	IWDT	IWDT Control Register	IWDTCR	16	16	2, 3 PCLKB	2 ICLK	IWDTa	1503
0008 8034h	IWDT	IWDT Status Register	IWDTSR	16	16	2, 3 PCLKB	2 ICLK	IWDTa	1506
0008 8036h	IWDT	IWDT Reset Control Register	IWDTRCR	8	8	2, 3 PCLKB	2 ICLK	IWDTa	1507
0008 8038h	IWDT	IWDT Count Stop Control Register	IWDTCSTPR	8	8	2, 3 PCLKB	2 ICLK	IWDTa	1508
0008 8040h	DA	D/A Data Register 0	DADR0	16	16	2, 3 PCLKB	2 ICLK	R12DA	2709
0008 8042h	DA	D/A Data Register 1	DADR1	16	16	2, 3 PCLKB	2 ICLK	R12DA	2709
0008 8044h	DA	D/A Control Register	DACR	8	8	2, 3 PCLKB	2 ICLK	R12DA	2710
0008 8045h	DA	DADRm Format Select Register	DADPR	8	8	2, 3 PCLKB	2 ICLK	R12DA	2711
0008 8046h	DA	D/A A/D Synchronous Start Control Register	DAADSCR	8	8	2, 3 PCLKB	2 ICLK	R12DA	2712
0008 8048h	DA	D/A Output Amplifier Control Register	DAAMPCR	8	8	2, 3 PCLKB	2 ICLK	R12DA	2713
0008 8100h	TPUA	Timer Start Register	TSTR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1289
0008 8101h	TPUA	Timer Synchronous Register	TSYR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1290
0008 8108h	TPU0	Noise Filter Control Register	NFCR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1291
0008 8109h	TPU1	Noise Filter Control Register	NFCR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1291
0008 810Ah	TPU2	Noise Filter Control Register	NFCR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1291
0008 810Bh	TPU3	Noise Filter Control Register	NFCR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1291
0008 810Ch	TPU4	Noise Filter Control Register	NFCR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1291
0008 810Dh	TPU5	Noise Filter Control Register	NFCR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1291
0008 8110h	TPU0	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1270
0008 8111h	TPU0	Timer Mode Register	TMDR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1274
0008 8112h	TPU0	Timer I/O Control Register H	TIORH	8	8	2, 3 PCLKB	2 ICLK	TPUa	1275
0008 8113h	TPU0	Timer I/O Control Register L	TIORL	8	8	2, 3 PCLKB	2 ICLK	TPUa	1275
0008 8114h	TPU0	Timer Interrupt Enable Register	TIER	8	8	2, 3 PCLKB	2 ICLK	TPUa	1284
0008 8115h	TPU0	Timer Status Register	TSR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1285
0008 8116h	TPU0	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 8118h	TPU0	Timer General Register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 811Ah	TPU0	Timer General Register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 811Ch	TPU0	Timer General Register C	TGRC	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 811Eh	TPU0	Timer General Register D	TGRD	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 8120h	TPU1	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1270
0008 8121h	TPU1	Timer Mode Register	TMDR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1274
0008 8122h	TPU1	Timer I/O Control Register	TIOR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1275
0008 8124h	TPU1	Timer Interrupt Enable Register	TIER	8	8	2, 3 PCLKB	2 ICLK	TPUa	1284
0008 8125h	TPU1	Timer Status Register	TSR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1285
0008 8126h	TPU1	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 8128h	TPU1	Timer General Register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 812Ah	TPU1	Timer General Register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 8130h	TPU2	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1270
0008 8131h	TPU2	Timer Mode Register	TMDR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1274
0008 8132h	TPU2	Timer I/O Control Register	TIOR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1275
0008 8134h	TPU2	Timer Interrupt Enable Register	TIER	8	8	2, 3 PCLKB	2 ICLK	TPUa	1284
0008 8135h	TPU2	Timer Status Register	TSR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1285
0008 8136h	TPU2	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 8138h	TPU2	Timer General Register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 813Ah	TPU2	Timer General Register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 8140h	TPU3	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1270

Table 5.1 List of I/O Registers (Address Order) (15 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 8141h	TPU3	Timer Mode Register	TMDR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1274
0008 8142h	TPU3	Timer I/O Control Register H	TIORH	8	8	2, 3 PCLKB	2 ICLK	TPUa	1275
0008 8143h	TPU3	Timer I/O Control Register L	TIORL	8	8	2, 3 PCLKB	2 ICLK	TPUa	1275
0008 8144h	TPU3	Timer Interrupt Enable Register	TIER	8	8	2, 3 PCLKB	2 ICLK	TPUa	1284
0008 8145h	TPU3	Timer Status Register	TSR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1285
0008 8146h	TPU3	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 8148h	TPU3	Timer General Register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 814Ah	TPU3	Timer General Register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 814Ch	TPU3	Timer General Register C	TGRC	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 814Eh	TPU3	Timer General Register D	TGRD	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 8150h	TPU4	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1270
0008 8151h	TPU4	Timer Mode Register	TMDR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1274
0008 8152h	TPU4	Timer I/O Control Register	TIOR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1275
0008 8154h	TPU4	Timer Interrupt Enable Register	TIER	8	8	2, 3 PCLKB	2 ICLK	TPUa	1284
0008 8155h	TPU4	Timer Status Register	TSR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1285
0008 8156h	TPU4	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 8158h	TPU4	Timer General Register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 815Ah	TPU4	Timer General Register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 8160h	TPU5	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1270
0008 8161h	TPU5	Timer Mode Register	TMDR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1274
0008 8162h	TPU5	Timer I/O Control Register	TIOR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1275
0008 8164h	TPU5	Timer Interrupt Enable Register	TIER	8	8	2, 3 PCLKB	2 ICLK	TPUa	1284
0008 8165h	TPU5	Timer Status Register	TSR	8	8	2, 3 PCLKB	2 ICLK	TPUa	1285
0008 8166h	TPU5	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 8168h	TPU5	Timer General Register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 816Ah	TPU5	Timer General Register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	TPUa	1288
0008 81E6h	PPG0	PPG Output Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	PPG	1356
0008 81E7h	PPG0	PPG Output Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	PPG	1358
0008 81E8h	PPG0	Next Data Enable Registers H	NDERH	8	8	2, 3 PCLKB	2 ICLK	PPG	1347
0008 81E9h	PPG0	Next Data Enable Registers L	NDERL	8	8	2, 3 PCLKB	2 ICLK	PPG	1347
0008 81EAh	PPG0	Output Data Registers H	PODRH	8	8	2, 3 PCLKB	2 ICLK	PPG	1349
0008 81EBh	PPG0	Output Data Registers L	PODRL	8	8	2, 3 PCLKB	2 ICLK	PPG	1349
0008 81ECh	PPG0	Next Data Registers H*1	NDRH	8	8	2, 3 PCLKB	2 ICLK	PPG	1352
0008 81EDh	PPG0	Next Data Registers L*2	NDRL	8	8	2, 3 PCLKB	2 ICLK	PPG	1353
0008 81EEh	PPG0	Next Data Registers H*1	NDRH2	8	8	2, 3 PCLKB	2 ICLK	PPG	1352
0008 81EFh	PPG0	Next Data Registers L*2	NDRL2	8	8	2, 3 PCLKB	2 ICLK	PPG	1353
0008 81F0h	PPG1	PPG Trigger Select Register	PTRSLR	8	8	2, 3 PCLKB	2 ICLK	PPG	1346
0008 81F6h	PPG1	PPG Output Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	PPG	1356
0008 81F7h	PPG1	PPG Output Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	PPG	1358
0008 81F8h	PPG1	Next Data Enable Registers H	NDERH	8	8	2, 3 PCLKB	2 ICLK	PPG	1347
0008 81F9h	PPG1	Next Data Enable Registers L	NDERL	8	8	2, 3 PCLKB	2 ICLK	PPG	1347
0008 81FAh	PPG1	Output Data Registers H	PODRH	8	8	2, 3 PCLKB	2 ICLK	PPG	1349
0008 81FBh	PPG1	Output Data Registers L	PODRL	8	8	2, 3 PCLKB	2 ICLK	PPG	1349
0008 81FCh	PPG1	Next Data Registers H*3	NDRH	8	8	2, 3 PCLKB	2 ICLK	PPG	1352
0008 81FDh	PPG1	Next Data Registers L*4	NDRL	8	8	2, 3 PCLKB	2 ICLK	PPG	1353
0008 81FEh	PPG1	Next Data Registers H*3	NDRH2	8	8	2, 3 PCLKB	2 ICLK	PPG	1352
0008 81FFh	PPG1	Next Data Registers L*4	NDRL2	8	8	2, 3 PCLKB	2 ICLK	PPG	1353
0008 8200h	TMR0	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	TMR	1380
0008 8201h	TMR1	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	TMR	1380
0008 8202h	TMR0	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	TMR	1383

Table 5.1 List of I/O Registers (Address Order) (16 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 8203h	TMR1	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	TMR	1384
0008 8204h	TMR0	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	TMR	1379
0008 8204h	TMR01	Time Constant Register A	TCORA	16	16	2, 3 PCLKB	2 ICLK	TMR	1379
0008 8205h	TMR1	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	TMR	1379
0008 8206h	TMR0	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	TMR	1379
0008 8206h	TMR01	Time Constant Register B	TCORB	16	16	2, 3 PCLKB	2 ICLK	TMR	1379
0008 8207h	TMR1	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	TMR	1379
0008 8208h	TMR0	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	TMR	1378
0008 8208h	TMR01	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	TMR	1378
0008 8209h	TMR1	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	TMR	1378
0008 820Ah	TMR0	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	TMR	1381
0008 820Ah	TMR01	Timer Counter Control Register	TCCR	16	16	2, 3 PCLKB	2 ICLK	TMR	1381
0008 820Bh	TMR1	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	TMR	1381
0008 820Ch	TMR0	Timer Counter Start Register	TCSTR	8	8	2, 3 PCLKB	2 ICLK	TMR	1385
0008 820Dh	TMR1	Timer Counter Start Register	TCSTR	8	8	2, 3 PCLKB	2 ICLK	TMR	1385
0008 8210h	TMR2	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	TMR	1380
0008 8211h	TMR3	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	TMR	1380
0008 8212h	TMR2	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	TMR	1383
0008 8213h	TMR3	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	TMR	1384
0008 8214h	TMR2	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	TMR	1379
0008 8214h	TMR23	Time Constant Register A	TCORA	16	16	2, 3 PCLKB	2 ICLK	TMR	1379
0008 8215h	TMR3	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	TMR	1379
0008 8216h	TMR2	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	TMR	1379
0008 8216h	TMR23	Time Constant Register B	TCORB	16	16	2, 3 PCLKB	2 ICLK	TMR	1379
0008 8217h	TMR3	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	TMR	1379
0008 8218h	TMR2	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	TMR	1378
0008 8218h	TMR23	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	TMR	1378
0008 8219h	TMR3	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	TMR	1378
0008 821Ah	TMR2	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	TMR	1381
0008 821Ah	TMR23	Timer Counter Control Register	TCCR	16	16	2, 3 PCLKB	2 ICLK	TMR	1381
0008 821Bh	TMR3	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	TMR	1381
0008 821Ch	TMR2	Timer Counter Start Register	TCSTR	8	8	2, 3 PCLKB	2 ICLK	TMR	1385
0008 821Dh	TMR3	Timer Counter Start Register	TCSTR	8	8	2, 3 PCLKB	2 ICLK	TMR	1385
0008 8280h	CRC	CRC Control Register	CRCCR	8	8	2, 3 PCLKB	2 ICLK	CRC	2409
0008 8281h	CRC	CRC Data Input Register	CRCDIR	8	8	2, 3 PCLKB	2 ICLK	CRC	2409
0008 8282h	CRC	CRC Data Output Register	CRCDOR	16	16	2, 3 PCLKB	2 ICLK	CRC	2410
0008 8300h	RIIC0	I^2C-Bus Control Register 1	ICCR1	8	8	2, 3 PCLKB	2 ICLK	RIICa	2151
0008 8301h	RIIC0	I^2C-Bus Control Register 2	ICCR2	8	8	2, 3 PCLKB	2 ICLK	RIICa	2153
0008 8302h	RIIC0	I^2C-Bus Mode Register 1	ICMR1	8	8	2, 3 PCLKB	2 ICLK	RIICa	2157
0008 8303h	RIIC0	I^2C-Bus Mode Register 2	ICMR2	8	8	2, 3 PCLKB	2 ICLK	RIICa	2158
0008 8304h	RIIC0	I^2C-Bus Mode Register 3	ICMR3	8	8	2, 3 PCLKB	2 ICLK	RIICa	2160
0008 8305h	RIIC0	I^2C-Bus Function Enable Register	ICFER	8	8	2, 3 PCLKB	2 ICLK	RIICa	2162
0008 8306h	RIIC0	I^2C-Bus Status Enable Register	ICSER	8	8	2, 3 PCLKB	2 ICLK	RIICa	2164
0008 8307h	RIIC0	I^2C-Bus Interrupt Enable Register	ICIER	8	8	2, 3 PCLKB	2 ICLK	RIICa	2166
0008 8308h	RIIC0	I^2C-Bus Status Register 1	ICSR1	8	8	2, 3 PCLKB	2 ICLK	RIICa	2168
0008 8309h	RIIC0	I^2C-Bus Status Register 2	ICSR2	8	8	2, 3 PCLKB	2 ICLK	RIICa	2171
0008 830Ah	RIIC0	Slave Address Register L0	SARL0	8	8	2, 3 PCLKB	2 ICLK	RIICa	2174
0008 830Bh	RIIC0	Slave Address Register U0	SARU0	8	8	2, 3 PCLKB	2 ICLK	RIICa	2175
0008 830Ch	RIIC0	Slave Address Register L1	SARL1	8	8	2, 3 PCLKB	2 ICLK	RIICa	2174
0008 830Dh	RIIC0	Slave Address Register U1	SARU1	8	8	2, 3 PCLKB	2 ICLK	RIICa	2175

Table 5.1 List of I/O Registers (Address Order) (17 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 830Eh	RIIC0	Slave Address Register L2	SARL2	8	8	2, 3 PCLKB	2 ICLK	RIICa	2174
0008 830Fh	RIIC0	Slave Address Register U2	SARU2	8	8	2, 3 PCLKB	2 ICLK	RIICa	2175
0008 8310h	RIIC0	I2C-Bus Bit Rate Low-Level Register	ICBRL	8	8	2, 3 PCLKB	2 ICLK	RIICa	2176
0008 8311h	RIIC0	I2C-Bus Bit Rate High-Level Register	ICBRH	8	8	2, 3 PCLKB	2 ICLK	RIICa	2177
0008 8312h	RIIC0	I2C-Bus Transmit Data Register	ICDRT	8	8	2, 3 PCLKB	2 ICLK	RIICa	2179
0008 8313h	RIIC0	I2C-Bus Receive Data Register	ICDRR	8	8	2, 3 PCLKB	2 ICLK	RIICa	2179
0008 8340h	RIIC2	I2C-Bus Control Register 1	ICCR1	8	8	2, 3 PCLKB	2 ICLK	RIICa	2151
0008 8341h	RIIC2	I2C-Bus Control Register 2	ICCR2	8	8	2, 3 PCLKB	2 ICLK	RIICa	2153
0008 8342h	RIIC2	I2C-Bus Mode Register 1	ICMR1	8	8	2, 3 PCLKB	2 ICLK	RIICa	2157
0008 8343h	RIIC2	I2C-Bus Mode Register 2	ICMR2	8	8	2, 3 PCLKB	2 ICLK	RIICa	2158
0008 8344h	RIIC2	I2C-Bus Mode Register 3	ICMR3	8	8	2, 3 PCLKB	2 ICLK	RIICa	2160
0008 8345h	RIIC2	I2C-Bus Function Enable Register	ICFER	8	8	2, 3 PCLKB	2 ICLK	RIICa	2162
0008 8346h	RIIC2	I2C-Bus Status Enable Register	ICSER	8	8	2, 3 PCLKB	2 ICLK	RIICa	2164
0008 8347h	RIIC2	I2C-Bus Interrupt Enable Register	ICIER	8	8	2, 3 PCLKB	2 ICLK	RIICa	2166
0008 8348h	RIIC2	I2C-Bus Status Register 1	ICSR1	8	8	2, 3 PCLKB	2 ICLK	RIICa	2168
0008 8349h	RIIC2	I2C-Bus Status Register 2	ICSR2	8	8	2, 3 PCLKB	2 ICLK	RIICa	2171
0008 834Ah	RIIC2	Slave Address Register L0	SARL0	8	8	2, 3 PCLKB	2 ICLK	RIICa	2174
0008 834Bh	RIIC2	Slave Address Register U0	SARU0	8	8	2, 3 PCLKB	2 ICLK	RIICa	2175
0008 834Ch	RIIC2	Slave Address Register L1	SARL1	8	8	2, 3 PCLKB	2 ICLK	RIICa	2174
0008 834Dh	RIIC2	Slave Address Register U1	SARU1	8	8	2, 3 PCLKB	2 ICLK	RIICa	2175
0008 834Eh	RIIC2	Slave Address Register L2	SARL2	8	8	2, 3 PCLKB	2 ICLK	RIICa	2174
0008 834Fh	RIIC2	Slave Address Register U2	SARU2	8	8	2, 3 PCLKB	2 ICLK	RIICa	2175
0008 8350h	RIIC2	I2C-Bus Bit Rate Low-Level Register	ICBRL	8	8	2, 3 PCLKB	2 ICLK	RIICa	2176
0008 8351h	RIIC2	I2C-Bus Bit Rate High-Level Register	ICBRH	8	8	2, 3 PCLKB	2 ICLK	RIICa	2177
0008 8352h	RIIC2	I2C-Bus Transmit Data Register	ICDRT	8	8	2, 3 PCLKB	2 ICLK	RIICa	2179
0008 8353h	RIIC2	I2C-Bus Receive Data Register	ICDRR	8	8	2, 3 PCLKB	2 ICLK	RIICa	2179
0008 8500h	MMCIF	Command Setting Register	CECMDSET	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2520
0008 8508h	MMCIF	Argument Register	CEARG	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2523
0008 850Ch	MMCIF	Automatically Issued CMD12 Argument Register	CEARGCMD12	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2523
0008 8510h	MMCIF	Command Control Register	CECMDCTRL	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2523
0008 8514h	MMCIF	Transfer Block Setting Register	CEBLOCKSET	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2524
0008 8518h	MMCIF	Clock Control Register	CECLKCTRL	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2525
0008 851Ch	MMCIF	Buffer Access Setting Register	CEBUFACC	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2526
0008 8520h	MMCIF	Response Register 3	CERESP3	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2527
0008 8524h	MMCIF	Response Register 2	CERESP2	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2527
0008 8528h	MMCIF	Response Register 1	CERESP1	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2527
0008 852Ch	MMCIF	Response Register 0	CERESP0	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2527
0008 8530h	MMCIF	Automatically Issued CMD12 Response Register	CERESPCMD12	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2527
0008 8534h	MMCIF	Data Register	CEDATA	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2527
0008 853Ch	MMCIF	Boot Operation Setting Register	CEBOOT	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2528
0008 8540h	MMCIF	Interrupt status Flag Register	CEINT	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2529
0008 8544h	MMCIF	Interrupt request Enable Register	CEINTEN	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2535
0008 8548h	MMCIF	Status Register 1	CEHOSTSTS1	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2536
0008 854Ch	MMCIF	Status Register 2	CEHOSTSTS2	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2537
0008 8570h	MMCIF	MMC Detection and Port Control Register	CEDETECT	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2540
0008 8574h	MMCIF	Special Mode Setting Register	CEADDMODE	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2541

Table 5.1 List of I/O Registers (Address Order) (18 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 857Ch	MMCIF	Version Register	CEVERSION	32	32	2, 3 PCLKB	2 ICLK	MMCIF	2541
0008 9000h	S12AD	A/D Control Register	ADCSR	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2623
0008 9004h	S12AD	A/D Channel Select Register A0	ADANSA0	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2627
0008 9008h	S12AD	A/D-Converted Value Addition/Average Mode Select Register 0	ADADS0	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2631
0008 900Ch	S12AD	A/D-Converted Value Addition/Average Count Select Register	ADADC	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2633
0008 900Eh	S12AD	A/D Control Extended Register	ADCER	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2634
0008 9010h	S12AD	A/D Start Trigger Select Register	ADSTRGR	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2636
0008 9014h	S12AD	A/D Channel Select Register B0	ADANSB0	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2629
0008 9018h	S12AD	A/D Data Duplication Register	ADDBLDR	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 901Eh	S12AD	A/D Self-Diagnosis Data Register	ADRD	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2619
0008 9020h	S12AD	A/D Data Register 0	ADDR0	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9022h	S12AD	A/D Data Register 1	ADDR1	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9024h	S12AD	A/D Data Register 2	ADDR2	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9026h	S12AD	A/D Data Register 3	ADDR3	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9028h	S12AD	A/D Data Register 4	ADDR4	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 902Ah	S12AD	A/D Data Register 5	ADDR5	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 902Ch	S12AD	A/D Data Register 6	ADDR6	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 902Eh	S12AD	A/D Data Register 7	ADDR7	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9060h	S12AD	A/D Sampling State Register 0	ADSSTR0	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 9066h	S12AD	A/D Sample-and-Hold Circuit Control Register	ADSHCR	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2642
0008 9073h	S12AD	A/D Sampling State Register 1	ADSSTR1	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 9074h	S12AD	A/D Sampling State Register 2	ADSSTR2	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 9075h	S12AD	A/D Sampling State Register 3	ADSSTR3	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 9076h	S12AD	A/D Sampling State Register 4	ADSSTR4	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 9077h	S12AD	A/D Sampling State Register 5	ADSSTR5	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 9078h	S12AD	A/D Sampling State Register 6	ADSSTR6	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 9079h	S12AD	A/D Sampling State Register 7	ADSSTR7	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 907Ah	S12AD	A/D Disconnection Detection Control Register	ADDISCR	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2644
0008 907Ch	S12AD	A/D Sample-and-Hold Operating Mode Select Register	ADSHMSR	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2643
0008 9080h	S12AD	A/D Group Scan Priority Control Register	ADGSPCR	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2645
0008 9084h	S12AD	A/D Data Duplication Register A	ADDBLDRA	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9086h	S12AD	A/D Data Duplication Register B	ADDBLDRB	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615

Table 5.1 List of I/O Registers (Address Order) (19 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 9090h	S12AD	A/D Compare Control Register	ADCMPCR	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2646
0008 9094h	S12AD	A/D Compare Channel Select Register 0	ADCMPANSR0	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2647
0008 9098h	S12AD	A/D Compare Level Register 0	ADCMPLR0	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2649
0008 909Ch	S12AD	A/D Compare Data Register 0	ADCMPDR0	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2652
0008 909Eh	S12AD	A/D Compare Data Register 1	ADCMPDR1	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2652
0008 90A0h	S12AD	A/D Compare Status Register 0	ADCMPSR0	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2655
0008 9100h	S12AD1	A/D Control Register	ADCSR	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2623
0008 9104h	S12AD1	A/D Channel Select Register A0	ADANSA0	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2627
0008 9106h	S12AD1	A/D Channel Select Register A1	ADANSA1	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2628
0008 9108h	S12AD1	A/D-Converted Value Addition/Average Mode Select Register 0	ADADS0	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2631
0008 910Ah	S12AD1	A/D-Converted Value Addition/Average Mode Select Register 1	ADADS1	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2632
0008 910Ch	S12AD1	A/D-Converted Value Addition/Average Count Select Register	ADADC	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2633
0008 910Eh	S12AD1	A/D Control Extended Register	ADCER	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2634
0008 9110h	S12AD1	A/D Start Trigger Select Register	ADSTRGR	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2636
0008 9112h	S12AD1	A/D Conversion Extended Input Control Register	ADEXICR	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2639
0008 9114h	S12AD1	A/D Channel Select Register B0	ADANSB0	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2629
0008 9116h	S12AD1	A/D Channel Select Register B1	ADANSB1	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2630
0008 9118h	S12AD1	A/D Data Duplication Register	ADDBLDR	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 911Ah	S12AD1	A/D Temperature Sensor Data Register	ADTSDR	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 911Ch	S12AD1	A/D Internal Reference Voltage Data Register	ADOCDR	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 911Eh	S12AD1	A/D Self-Diagnosis Data Register	ADRD	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2619
0008 9120h	S12AD1	A/D Data Register 0	ADDR0	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9122h	S12AD1	A/D Data Register 1	ADDR1	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9124h	S12AD1	A/D Data Register 2	ADDR2	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9126h	S12AD1	A/D Data Register 3	ADDR3	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9128h	S12AD1	A/D Data Register 4	ADDR4	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 912Ah	S12AD1	A/D Data Register 5	ADDR5	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 912Ch	S12AD1	A/D Data Register 6	ADDR6	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 912Eh	S12AD1	A/D Data Register 7	ADDR7	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9130h	S12AD1	A/D Data Register 8	ADDR8	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9132h	S12AD1	A/D Data Register 9	ADDR9	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615

Table 5.1 List of I/O Registers (Address Order) (20 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 9134h	S12AD1	A/D Data Register 10	ADDR10	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9136h	S12AD1	A/D Data Register 11	ADDR11	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9138h	S12AD1	A/D Data Register 12	ADDR12	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 913Ah	S12AD1	A/D Data Register 13	ADDR13	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 913Ch	S12AD1	A/D Data Register 14	ADDR14	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 913Eh	S12AD1	A/D Data Register 15	ADDR15	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9140h	S12AD1	A/D Data Register 16	ADDR16	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9142h	S12AD1	A/D Data Register 17	ADDR17	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9144h	S12AD1	A/D Data Register 18	ADDR18	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9146h	S12AD1	A/D Data Register 19	ADDR19	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9148h	S12AD1	A/D Data Register 20	ADDR20	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9160h	S12AD1	A/D Sampling State Register 0	ADSSTR0	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 9161h	S12AD1	A/D Sampling State Register L	ADSSTRL	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 9170h	S12AD1	A/D Sampling State Register T	ADSSTRT	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 9171h	S12AD1	A/D Sampling State Register O	ADSSTRO	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 9173h	S12AD1	A/D Sampling State Register 1	ADSSTR1	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 9174h	S12AD1	A/D Sampling State Register 2	ADSSTR2	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 9175h	S12AD1	A/D Sampling State Register 3	ADSSTR3	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 9176h	S12AD1	A/D Sampling State Register 4	ADSSTR4	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 9177h	S12AD1	A/D Sampling State Register 5	ADSSTR5	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 9178h	S12AD1	A/D Sampling State Register 6	ADSSTR6	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 9179h	S12AD1	A/D Sampling State Register 7	ADSSTR7	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2641
0008 917Ah	S12AD1	A/D Disconnection Detection Control Register	ADDISCR	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2644
0008 9180h	S12AD1	A/D Group Scan Priority Control Register	ADGSPCR	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2645
0008 9184h	S12AD1	A/D Data Duplication Register A	ADDBLDRA	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9186h	S12AD1	A/D Data Duplication Register B	ADDBLDRB	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2615
0008 9190h	S12AD1	A/D Compare Control Register	ADCMPCR	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2646
0008 9192h	S12AD1	A/D Compare Channel Select Extended Register	ADCMPANSER	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2648
0008 9193h	S12AD1	A/D Compare Level Extended Register	ADCMPLER	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2651
0008 9194h	S12AD1	A/D Compare Channel Select Register 0	ADCMPANSR0	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2647
0008 9196h	S12AD1	A/D Compare Channel Select Register 1	ADCMPANSR1	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2647

Table 5.1 List of I/O Registers (Address Order) (21 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 9198h	S12AD1	A/D Compare Level Register 0	ADCMPLR0	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2649
0008 919Ah	S12AD1	A/D Compare Level Register 1	ADCMPLR1	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2650
0008 919Ch	S12AD1	A/D Compare Data Register 0	ADCMPDR0	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2652
0008 919Eh	S12AD1	A/D Compare Data Register 1	ADCMPDR1	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2652
0008 91A0h	S12AD1	A/D Compare Status Register 0	ADCMPSR0	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2655
0008 91A2h	S12AD1	A/D Compare Status Register 1	ADCMPSR1	16	16	2, 3 PCLKB	2 ICLK	S12ADC	2656
0008 91A4h	S12AD1	A/D Compare Status Extended Register	ADCMPSER	8	8	2, 3 PCLKB	2 ICLK	S12ADC	2657
0008 9E00h	QSPI	QSPI Control Register	SPCR	8	8	4, 5 PCLKB	2, 3 ICLK	QSPI	2364
0008 9E01h	QSPI	QSPI Slave Select Polarity Register	SSLP	8	8	4, 5 PCLKB	2, 3 ICLK	QSPI	2365
0008 9E02h	QSPI	QSPI Pin Control Register	SPPCR	8	8	4, 5 PCLKB	2, 3 ICLK	QSPI	2366
0008 9E03h	QSPI	QSPI Status Register	SPSR	8	8	4, 5 PCLKB	2, 3 ICLK	QSPI	2367
0008 9E04h	QSPI	QSPI Data Register	SPDR	32	8, 16, 32	4, 5 PCLKB	2, 3 ICLK	QSPI	2369
0008 9E08h	QSPI	QSPI Sequence Control Register	SPSCR	8	8	4, 5 PCLKB	2, 3 ICLK	QSPI	2370
0008 9E09h	QSPI	QSPI Sequence Status Register	SPSSR	8	8	4, 5 PCLKB	2, 3 ICLK	QSPI	2370
0008 9E0Ah	QSPI	QSPI Bit Rate Register	SPBR	8	8	4, 5 PCLKB	2, 3 ICLK	QSPI	2371
0008 9E0Bh	QSPI	QSPI Data Control Register	SPDCR	8	8	4, 5 PCLKB	2, 3 ICLK	QSPI	2372
0008 9E0Ch	QSPI	QSPI Clock Delay Register	SPCKD	8	8	4, 5 PCLKB	2, 3 ICLK	QSPI	2373
0008 9E0Dh	QSPI	QSPI Slave Select Negation Delay Register	SSLND	8	8	4, 5 PCLKB	2, 3 ICLK	QSPI	2374
0008 9E0Eh	QSPI	QSPI Next-Access Delay Register	SPND	8	8	4, 5 PCLKB	2, 3 ICLK	QSPI	2375
0008 9E10h	QSPI	QSPI Command Register 0	SPCMD0	16	16	4, 5 PCLKB	2, 3 ICLK	QSPI	2376
0008 9E12h	QSPI	QSPI Command Register 1	SPCMD1	16	16	4, 5 PCLKB	2, 3 ICLK	QSPI	2376
0008 9E14h	QSPI	QSPI Command Register 2	SPCMD2	16	16	4, 5 PCLKB	2, 3 ICLK	QSPI	2376
0008 9E16h	QSPI	QSPI Command Register 3	SPCMD3	16	16	4, 5 PCLKB	2, 3 ICLK	QSPI	2376
0008 9E18h	QSPI	QSPI Buffer Control Register	SPBFCR	8	8	4, 5 PCLKB	2, 3 ICLK	QSPI	2380
0008 9E1Ah	QSPI	QSPI Buffer Data Count Set Register	SPBDCR	16	16	4, 5 PCLKB	2, 3 ICLK	QSPI	2381
0008 9E1Ch	QSPI	QSPI Transfer Data Length Multiplier Setting Register 0	SPBMUL0	32	32	4, 5 PCLKB	2, 3 ICLK	QSPI	2382
0008 9E20h	QSPI	QSPI Transfer Data Length Multiplier Setting Register 1	SPBMUL1	32	32	4, 5 PCLKB	2, 3 ICLK	QSPI	2382
0008 9E24h	QSPI	QSPI Transfer Data Length Multiplier Setting Register 2	SPBMUL2	32	32	4, 5 PCLKB	2, 3 ICLK	QSPI	2382
0008 9E28h	QSPI	QSPI Transfer Data Length Multiplier Setting Register 3	SPBMUL3	32	32	4, 5 PCLKB	2, 3 ICLK	QSPI	2382
0008 A000h	SCI0	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1969
0008 A001h	SCI0	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1985
0008 A002h	SCI0	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1973
0008 A003h	SCI0	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1967
0008 A004h	SCI0	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1978
0008 A005h	SCI0	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1965
0008 A006h	SMCI0	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1983
0008 A007h	SCI0	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1996
0008 A008h	SCI0	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1999

Table 5.1 List of I/O Registers (Address Order) (22 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 A009h	SCI0	I²C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2000
0008 A00Ah	SCI0	I²C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2001
0008 A00Bh	SCI0	I²C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2002
0008 A00Ch	SCI0	I²C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2004
0008 A00Dh	SCI0	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2005
0008 A00Eh	SCI0	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A00Fh	SCI0	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A00Eh	SCI0	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A010h	SCI0	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A011h	SCI0	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A010h	SCI0	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A012h	SCI0	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1995
0008 A020h	SCI1	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1969
0008 A021h	SCI1	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1985
0008 A022h	SCI1	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1973
0008 A023h	SCI1	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1967
0008 A024h	SCI1	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1978
0008 A025h	SCI1	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1965
0008 A026h	SMCI1	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1983
0008 A027h	SCI1	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1996
0008 A028h	SCI1	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1999
0008 A029h	SCI1	I²C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2000
0008 A02Ah	SCI1	I²C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2001
0008 A02Bh	SCI1	I²C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2002
0008 A02Ch	SCI1	I²C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2004
0008 A02Dh	SCI1	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2005
0008 A02Eh	SCI1	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A02Fh	SCI1	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A02Eh	SCI1	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A030h	SCI1	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A031h	SCI1	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1966

Table 5.1 List of I/O Registers (Address Order) (23 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 A030h	SCI1	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A032h	SCI1	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1995
0008 A040h	SCI2	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1969
0008 A041h	SCI2	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1985
0008 A042h	SCI2	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1973
0008 A043h	SCI2	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1967
0008 A044h	SCI2	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1978
0008 A045h	SCI2	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1965
0008 A046h	SMCI2	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1983
0008 A047h	SCI2	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1996
0008 A048h	SCI2	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1999
0008 A049h	SCI2	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2000
0008 A04Ah	SCI2	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2001
0008 A04Bh	SCI2	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2002
0008 A04Ch	SCI2	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2004
0008 A04Dh	SCI2	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2005
0008 A04Eh	SCI2	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A04Fh	SCI2	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A04Eh	SCI2	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A050h	SCI2	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A051h	SCI2	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A050h	SCI2	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A052h	SCI2	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1995
0008 A060h	SCI3	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1969
0008 A061h	SCI3	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1985
0008 A062h	SCI3	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1973
0008 A063h	SCI3	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1967
0008 A064h	SCI3	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1978
0008 A065h	SCI3	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1965
0008 A066h	SMCI3	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1983
0008 A067h	SCI3	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1996

Table 5.1 List of I/O Registers (Address Order) (24 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 A068h	SCI3	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1999
0008 A069h	SCI3	I2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2000
0008 A06Ah	SCI3	I2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2001
0008 A06Bh	SCI3	I2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2002
0008 A06Ch	SCI3	I2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2004
0008 A06Dh	SCI3	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2005
0008 A06Eh	SCI3	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A06Fh	SCI3	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A06Eh	SCI3	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A070h	SCI3	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A071h	SCI3	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A070h	SCI3	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A072h	SCI3	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1995
0008 A080h	SCI4	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1969
0008 A081h	SCI4	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1985
0008 A082h	SCI4	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1973
0008 A083h	SCI4	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1967
0008 A084h	SCI4	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1978
0008 A085h	SCI4	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1965
0008 A086h	SMCI4	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1983
0008 A087h	SCI4	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1996
0008 A088h	SCI4	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1999
0008 A089h	SCI4	I2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2000
0008 A08Ah	SCI4	I2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2001
0008 A08Bh	SCI4	I2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2002
0008 A08Ch	SCI4	I2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2004
0008 A08Dh	SCI4	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2005
0008 A08Eh	SCI4	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A08Fh	SCI4	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A08Eh	SCI4	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A090h	SCI4	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1966

Table 5.1 List of I/O Registers (Address Order) (25 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 A091h	SCI4	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A090h	SCI4	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A092h	SCI4	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1995
0008 A0A0h	SCI5	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1969
0008 A0A1h	SCI5	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1985
0008 A0A2h	SCI5	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1973
0008 A0A3h	SCI5	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1967
0008 A0A4h	SCI5	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1978
0008 A0A5h	SCI5	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1965
0008 A0A6h	SMCI5	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1983
0008 A0A7h	SCI5	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1996
0008 A0A8h	SCI5	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1999
0008 A0A9h	SCI5	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2000
0008 A0AAh	SCI5	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2001
0008 A0ABh	SCI5	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2002
0008 A0ACh	SCI5	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2004
0008 A0ADh	SCI5	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2005
0008 A0AEh	SCI5	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A0AFh	SCI5	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A0AEh	SCI5	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A0B0h	SCI5	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A0B1h	SCI5	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A0B0h	SCI5	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A0B2h	SCI5	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1995
0008 A0C0h	SCI6	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1969
0008 A0C1h	SCI6	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1985
0008 A0C2h	SCI6	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1973
0008 A0C3h	SCI6	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1967
0008 A0C4h	SCI6	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1978
0008 A0C5h	SCI6	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1965
0008 A0C6h	SMCI6	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1983

Table 5.1 List of I/O Registers (Address Order) (26 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 A0C7h	SCI6	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1996
0008 A0C8h	SCI6	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1999
0008 A0C9h	SCI6	I2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2000
0008 A0CAh	SCI6	I2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2001
0008 A0CBh	SCI6	I2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2002
0008 A0CCh	SCI6	I2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2004
0008 A0CDh	SCI6	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2005
0008 A0CEh	SCI6	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A0CFh	SCI6	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A0CEh	SCI6	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A0D0h	SCI6	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A0D1h	SCI6	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A0D0h	SCI6	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A0D2h	SCI6	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1995
0008 A0E0h	SCI7	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1969
0008 A0E1h	SCI7	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1985
0008 A0E2h	SCI7	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1973
0008 A0E3h	SCI7	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1967
0008 A0E4h	SCI7	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1978
0008 A0E5h	SCI7	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1965
0008 A0E6h	SMCI7	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1983
0008 A0E7h	SCI7	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1996
0008 A0E8h	SCI7	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1999
0008 A0E9h	SCI7	I2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2000
0008 A0EAh	SCI7	I2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2001
0008 A0EBh	SCI7	I2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2002
0008 A0ECh	SCI7	I2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2004
0008 A0EDh	SCI7	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	2005
0008 A0EEh	SCI7	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A0EFh	SCI7	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1968
0008 A0EEh	SCI7	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	SCIg, SCIh	1968

Table 5.1 List of I/O Registers (Address Order) (27 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 A0F0h	SCI7	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A0F1h	SCI7	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A0F0h	SCI7	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	SCIg, SCIh	1966
0008 A0F2h	SCI7	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	SCIg, SCIh	1995
0008 A500h	SSI0	Control Register	SSICR	32	32	2, 3 PCLKB	2 ICLK	SSI	2418
0008 A504h	SSI0	Status Register	SSISR	32	32	2, 3 PCLKB	2 ICLK	SSI	2422
0008 A510h	SSI0	FIFO Control Register	SSIFCR	32	32	2, 3 PCLKB	2 ICLK	SSI	2424
0008 A514h	SSI0	FIFO Status Register	SSIFSR	32	32	2, 3 PCLKB	2 ICLK	SSI	2426
0008 A518h	SSI0	Transmit FIFO Data Register	SSIFTDR	32	32	2, 3 PCLKB	2 ICLK	SSI	2428
0008 A51Ch	SSI0	Receive FIFO Data Register	SSIFRDR	32	32	2, 3 PCLKB	2 ICLK	SSI	2428
0008 A520h	SSI0	TDM Mode Register	SSITDMR	32	32	2, 3 PCLKB	2 ICLK	SSI	2429
0008 A540h	SSI1	Control Register	SSICR	32	32	2, 3 PCLKB	2 ICLK	SSI	2418
0008 A544h	SSI1	Status Register	SSISR	32	32	2, 3 PCLKB	2 ICLK	SSI	2422
0008 A550h	SSI1	FIFO Control Register	SSIFCR	32	32	2, 3 PCLKB	2 ICLK	SSI	2424
0008 A554h	SSI1	FIFO Status Register	SSIFSR	32	32	2, 3 PCLKB	2 ICLK	SSI	2426
0008 A558h	SSI1	Transmit FIFO Data Register	SSIFTDR	32	32	2, 3 PCLKB	2 ICLK	SSI	2428
0008 A55Ch	SSI1	Receive FIFO Data Register	SSIFRDR	32	32	2, 3 PCLKB	2 ICLK	SSI	2428
0008 A560h	SSI1	TDM Mode Register	SSITDMR	32	32	2, 3 PCLKB	2 ICLK	SSI	2429
0008 AC00h	SDHI	Command Register	SDCMD	32	32	2, 3 PCLKB	2 ICLK	SDHI	2465
0008 AC08h	SDHI	Argument Register	SDARG	32	32	2, 3 PCLKB	2 ICLK	SDHI	2467
0008 AC10h	SDHI	Data Stop Register	SDSTOP	32	32	2, 3 PCLKB	2 ICLK	SDHI	2467
0008 AC14h	SDHI	Block Count Register	SDBLKCNT	32	32	2, 3 PCLKB	2 ICLK	SDHI	2468
0008 AC18h	SDHI	Response Register 10	SDRSP10	32	32	2, 3 PCLKB	2 ICLK	SDHI	2469
0008 AC20h	SDHI	Response Register 32	SDRSP32	32	32	2, 3 PCLKB	2 ICLK	SDHI	2469
0008 AC28h	SDHI	Response Register 54	SDRSP54	32	32	2, 3 PCLKB	2 ICLK	SDHI	2469
0008 AC30h	SDHI	Response Register 76	SDRSP76	32	32	2, 3 PCLKB	2 ICLK	SDHI	2469
0008 AC38h	SDHI	SD Status Register 1	SDSTS1	32	32	2, 3 PCLKB	2 ICLK	SDHI	2470
0008 AC3Ch	SDHI	SD Status Register 2	SDSTS2	32	32	2, 3 PCLKB	2 ICLK	SDHI	2473
0008 AC40h	SDHI	SD Interrupt Mask Register 1	SDIMSK1	32	32	2, 3 PCLKB	2 ICLK	SDHI	2477
0008 AC44h	SDHI	SD Interrupt Mask Register 2	SDIMSK2	32	32	2, 3 PCLKB	2 ICLK	SDHI	2478
0008 AC48h	SDHI	SDHI Clock Control Register	SDCLKCR	32	32	2, 3 PCLKB	2 ICLK	SDHI	2479
0008 AC4Ch	SDHI	Transfer Data Size Register	SDSIZE	32	32	2, 3 PCLKB	2 ICLK	SDHI	2480
0008 AC50h	SDHI	Card Access Option Register	SDOPT	32	32	2, 3 PCLKB	2 ICLK	SDHI	2481
0008 AC58h	SDHI	SD Error Status Register 1	SDERSTS1	32	32	2, 3 PCLKB	2 ICLK	SDHI	2482
0008 AC5Ch	SDHI	SD Error Status Register 2	SDERSTS2	32	32	2, 3 PCLKB	2 ICLK	SDHI	2483
0008 AC60h	SDHI	SD Buffer Register	SDBUFR	32	32	2, 3 PCLKB	2 ICLK	SDHI	2484
0008 AC68h	SDHI	SDIO Mode Control Register	SDIOMD	32	32	2, 3 PCLKB	2 ICLK	SDHI	2484
0008 AC6Ch	SDHI	SDIO Status Register	SDIOSTS	32	32	2, 3 PCLKB	2 ICLK	SDHI	2486
0008 AC70h	SDHI	SDIO Interrupt Mask Register	SDIOIMSK	32	32	2, 3 PCLKB	2 ICLK	SDHI	2487
0008 ADB0h	SDHI	DMA Transfer Enable Register	SDDMAEN	32	32	2, 3 PCLKB	2 ICLK	SDHI	2488
0008 ADC0h	SDHI	SDHI Software Reset Register	SDRST	32	32	2, 3 PCLKB	2 ICLK	SDHI	2489
0008 ADC4h	SDHI	Version Register	SDVER	32	32	2, 3 PCLKB	2 ICLK	SDHI	2490
0008 ADE0h	SDHI	Swap Control Register	SDSWAP	32	32	2, 3 PCLKB	2 ICLK	SDHI	2491
0008 B000h	CAC	CAC Control Register 0	CACR0	8	8	2, 3 PCLKB	2 ICLK	CAC	336
0008 B001h	CAC	CAC Control Register 1	CACR1	8	8	2, 3 PCLKB	2 ICLK	CAC	337
0008 B002h	CAC	CAC Control Register 2	CACR2	8	8	2, 3 PCLKB	2 ICLK	CAC	338
0008 B003h	CAC	CAC Interrupt Request Enable Register	CAICR	8	8	2, 3 PCLKB	2 ICLK	CAC	339

Table 5.1 List of I/O Registers (Address Order) (28 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 B004h	CAC	CAC Status Register	CASTR	8	8	2, 3 PCLKB	2 ICLK	CAC	340
0008 B006h	CAC	CAC Upper-Limit Value Setting Register	CAULVR	16	16	2, 3 PCLKB	2 ICLK	CAC	341
0008 B008h	CAC	CAC Lower-Limit Value Setting Register	CALLVR	16	16	2, 3 PCLKB	2 ICLK	CAC	341
0008 B00Ah	CAC	CAC Counter Buffer Register	CACNTBR	16	16	2, 3 PCLKB	2 ICLK	CAC	341
0008 B080h	DOC	DOC Control Register	DOCR	8	8	2, 3 PCLKB	2 ICLK	DOC	2728
0008 B082h	DOC	DOC Data Input Register	DODIR	16	16	2, 3 PCLKB	2 ICLK	DOC	2729
0008 B084h	DOC	DOC Data Setting Register	DODSR	16	16	2, 3 PCLKB	2 ICLK	DOC	2729
0008 B100h	ELC	Event Link Control Register	ELCR	8	8	2, 3 PCLKB	2 ICLK	ELC	753
0008 B101h	ELC	Event Link Setting Register 0	ELSR0	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B104h	ELC	Event Link Setting Register 3	ELSR3	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B105h	ELC	Event Link Setting Register 4	ELSR4	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B108h	ELC	Event Link Setting Register 7	ELSR7	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B10Bh	ELC	Event Link Setting Register 10	ELSR10	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B10Ch	ELC	Event Link Setting Register 11	ELSR11	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B10Dh	ELC	Event Link Setting Register 12	ELSR12	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B10Eh	ELC	Event Link Setting Register 13	ELSR13	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B110h	ELC	Event Link Setting Register 15	ELSR15	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B111h	ELC	Event Link Setting Register 16	ELSR16	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B113h	ELC	Event Link Setting Register 18	ELSR18	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B114h	ELC	Event Link Setting Register 19	ELSR19	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B115h	ELC	Event Link Setting Register 20	ELSR20	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B116h	ELC	Event Link Setting Register 21	ELSR21	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B117h	ELC	Event Link Setting Register 22	ELSR22	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B118h	ELC	Event Link Setting Register 23	ELSR23	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B119h	ELC	Event Link Setting Register 24	ELSR24	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B11Ah	ELC	Event Link Setting Register 25	ELSR25	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B11Bh	ELC	Event Link Setting Register 26	ELSR26	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B11Ch	ELC	Event Link Setting Register 27	ELSR27	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B11Dh	ELC	Event Link Setting Register 28	ELSR28	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B11Fh	ELC	Event Link Option Setting Register A	ELOPA	8	8	2, 3 PCLKB	2 ICLK	ELC	760
0008 B120h	ELC	Event Link Option Setting Register B	ELOPB	8	8	2, 3 PCLKB	2 ICLK	ELC	761
0008 B121h	ELC	Event Link Option Setting Register C	ELOPC	8	8	2, 3 PCLKB	2 ICLK	ELC	761
0008 B122h	ELC	Event Link Option Setting Register D	ELOPD	8	8	2, 3 PCLKB	2 ICLK	ELC	762
0008 B123h	ELC	Port Group Setting Register 1	PGR1	8	8	2, 3 PCLKB	2 ICLK	ELC	767
0008 B124h	ELC	Port Group Setting Register 2	PGR2	8	8	2, 3 PCLKB	2 ICLK	ELC	767
0008 B125h	ELC	Port Group Control Register 1	PGC1	8	8	2, 3 PCLKB	2 ICLK	ELC	768
0008 B126h	ELC	Port Group Control Register 2	PGC2	8	8	2, 3 PCLKB	2 ICLK	ELC	768
0008 B127h	ELC	Port Buffer Register 1	PDBF1	8	8	2, 3 PCLKB	2 ICLK	ELC	769
0008 B128h	ELC	Port Buffer Register 2	PDBF2	8	8	2, 3 PCLKB	2 ICLK	ELC	769
0008 B129h	ELC	Event Link Port Setting Register 0	PEL0	8	8	2, 3 PCLKB	2 ICLK	ELC	770
0008 B12Ah	ELC	Event Link Port Setting Register 1	PEL1	8	8	2, 3 PCLKB	2 ICLK	ELC	770
0008 B12Bh	ELC	Event Link Port Setting Register 2	PEL2	8	8	2, 3 PCLKB	2 ICLK	ELC	770
0008 B12Ch	ELC	Event Link Port Setting Register 3	PEL3	8	8	2, 3 PCLKB	2 ICLK	ELC	770
0008 B12Dh	ELC	Event Link Software Event Generation Register	ELSEGR	8	8	2, 3 PCLKB	2 ICLK	ELC	771
0008 B131h	ELC	Event Link Setting Register 33	ELSR33	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B133h	ELC	Event Link Setting Register 35	ELSR35	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B134h	ELC	Event Link Setting Register 36	ELSR36	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B135h	ELC	Event Link Setting Register 37	ELSR37	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B136h	ELC	Event Link Setting Register 38	ELSR38	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B139h	ELC	Event Link Setting Register 41	ELSR41	8	8	2, 3 PCLKB	2 ICLK	ELC	754

Table 5.1 List of I/O Registers (Address Order) (29 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 B13Ah	ELC	Event Link Setting Register 42	ELSR42	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B13Bh	ELC	Event Link Setting Register 43	ELSR43	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B13Ch	ELC	Event Link Setting Register 44	ELSR44	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B13Dh	ELC	Event Link Setting Register 45	ELSR45	8	8	2, 3 PCLKB	2 ICLK	ELC	754
0008 B13Fh	ELC	Event Link Option Setting Register F	ELOPF	8	8	2, 3 PCLKB	2 ICLK	ELC	763
0008 B141h	ELC	Event Link Option Setting Register H	ELOPH	8	8	2, 3 PCLKB	2 ICLK	ELC	764
0008 B142h	ELC	Event Link Option Setting Register I	ELOPI	8	8	2, 3 PCLKB	2 ICLK	ELC	765
0008 B143h	ELC	Event Link Option Setting Register J	ELOPJ	8	8	2, 3 PCLKB	2 ICLK	ELC	766
0008 B300h	SCI12	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	SCIh	1969
0008 B301h	SCI12	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	SCIh	1985
0008 B302h	SCI12	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	SCIh	1973
0008 B303h	SCI12	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	SCIh	1967
0008 B304h	SCI12	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	SCIh	1978
0008 B305h	SCI12	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	SCIh	1965
0008 B306h	SMCI12	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	SCIh	1983
0008 B307h	SCI12	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	SCIh	1996
0008 B308h	SCI12	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	SCIh	1999
0008 B309h	SCI12	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	SCIh	2000
0008 B30Ah	SCI12	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	SCIh	2001
0008 B30Bh	SCI12	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	SCIh	2002
0008 B30Ch	SCI12	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	SCIh	2004
0008 B30Dh	SCI12	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	SCIh	2005
0008 B30Eh	SCI12	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	SCIh	1968
0008 B30Fh	SCI12	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	SCIh	1968
0008 B30Eh	SCI12	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	SCIh	1968
0008 B310h	SCI12	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	SCIh	1966
0008 B311h	SCI12	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	SCIh	1966
0008 B310h	SCI12	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	SCIh	1966
0008 B312h	SCI12	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	SCIh	1995
0008 B320h	SCI12	Extended Serial Module Enable Register	ESMER	8	8	2, 3 PCLKB	2 ICLK	SCIh	2006
0008 B321h	SCI12	Control Register 0	CR0	8	8	2, 3 PCLKB	2 ICLK	SCIh	2007
0008 B322h	SCI12	Control Register 1	CR1	8	8	2, 3 PCLKB	2 ICLK	SCIh	2007
0008 B323h	SCI12	Control Register 2	CR2	8	8	2, 3 PCLKB	2 ICLK	SCIh	2008
0008 B324h	SCI12	Control Register 3	CR3	8	8	2, 3 PCLKB	2 ICLK	SCIh	2009
0008 B325h	SCI12	Port Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	SCIh	2009
0008 B326h	SCI12	Interrupt Control Register	ICR	8	8	2, 3 PCLKB	2 ICLK	SCIh	2010
0008 B327h	SCI12	Status Register	STR	8	8	2, 3 PCLKB	2 ICLK	SCIh	2011
0008 B328h	SCI12	Status Clear Register	STCR	8	8	2, 3 PCLKB	2 ICLK	SCIh	2012
0008 B329h	SCI12	Control Field 0 Data Register	CF0DR	8	8	2, 3 PCLKB	2 ICLK	SCIh	2012
0008 B32Ah	SCI12	Control Field 0 Compare Enable Register	CF0CR	8	8	2, 3 PCLKB	2 ICLK	SCIh	2013
0008 B32Bh	SCI12	Control Field 0 Receive Data Register	CF0RR	8	8	2, 3 PCLKB	2 ICLK	SCIh	2013
0008 B32Ch	SCI12	Primary Control Field 1 Data Register	PCF1DR	8	8	2, 3 PCLKB	2 ICLK	SCIh	2013
0008 B32Dh	SCI12	Secondary Control Field 1 Data Register	SCF1DR	8	8	2, 3 PCLKB	2 ICLK	SCIh	2014
0008 B32Eh	SCI12	Control Field 1 Compare Enable Register	CF1CR	8	8	2, 3 PCLKB	2 ICLK	SCIh	2014
0008 B32Fh	SCI12	Control Field 1 Receive Data Register	CF1RR	8	8	2, 3 PCLKB	2 ICLK	SCIh	2014
0008 B330h	SCI12	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	SCIh	2015
0008 B331h	SCI12	Timer Mode Register	TMR	8	8	2, 3 PCLKB	2 ICLK	SCIh	2015
0008 B332h	SCI12	Timer Prescaler Register	TPRE	8	8	2, 3 PCLKB	2 ICLK	SCIh	2016
0008 B333h	SCI12	Timer Count Register	TCNT	8	8	2, 3 PCLKB	2 ICLK	SCIh	2016

Table 5.1 List of I/O Registers (Address Order) (30 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 C000h	PORT0	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	787
0008 C001h	PORT1	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	787
0008 C002h	PORT2	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	787
0008 C003h	PORT3	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	787
0008 C004h	PORT4	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	787
0008 C005h	PORT5	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	787
0008 C006h	PORT6	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	787
0008 C007h	PORT7	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	787
0008 C008h	PORT8	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	787
0008 C009h	PORT9	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	787
0008 C00Ah	PORTA	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	787
0008 C00Bh	PORTB	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	787
0008 C00Ch	PORTC	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	787
0008 C00Dh	PORTD	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	787
0008 C00Eh	PORTE	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	787
0008 C00Fh	PORTF	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	787
0008 C010h	PORTG	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	787
0008 C012h	PORTJ	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	787
0008 C020h	PORT0	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	788
0008 C021h	PORT1	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	788
0008 C022h	PORT2	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	788
0008 C023h	PORT3	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	788
0008 C024h	PORT4	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	788
0008 C025h	PORT5	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	788
0008 C026h	PORT6	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	788
0008 C027h	PORT7	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	788
0008 C028h	PORT8	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	788
0008 C029h	PORT9	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	788
0008 C02Ah	PORTA	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	788
0008 C02Bh	PORTB	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	788
0008 C02Ch	PORTC	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	788

Table 5.1 List of I/O Registers (Address Order) (31 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 C02Dh	PORTD	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	788
0008 C02Eh	PORTE	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	788
0008 C02Fh	PORTF	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	788
0008 C030h	PORTG	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	788
0008 C032h	PORTJ	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	788
0008 C040h	PORT0	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	789
0008 C041h	PORT1	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	789
0008 C042h	PORT2	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	789
0008 C043h	PORT3	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	789
0008 C044h	PORT4	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	789
0008 C045h	PORT5	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	789
0008 C046h	PORT6	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	789
0008 C047h	PORT7	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	789
0008 C048h	PORT8	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	789
0008 C049h	PORT9	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	789
0008 C04Ah	PORTA	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	789
0008 C04Bh	PORTB	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	789
0008 C04Ch	PORTC	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	789
0008 C04Dh	PORTD	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	789
0008 C04Eh	PORTE	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	789
0008 C04Fh	PORTF	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	789
0008 C050h	PORTG	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	789
0008 C052h	PORTJ	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	789
0008 C060h	PORT0	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	790
0008 C061h	PORT1	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	790
0008 C062h	PORT2	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	790
0008 C063h	PORT3	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	790
0008 C064h	PORT4	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	790
0008 C065h	PORT5	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	790
0008 C066h	PORT6	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	790
0008 C067h	PORT7	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	790

Table 5.1 List of I/O Registers (Address Order) (32 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 C068h	PORT8	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	790
0008 C069h	PORT9	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	790
0008 C06Ah	PORTA	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	790
0008 C06Bh	PORTB	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	790
0008 C06Ch	PORTC	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	790
0008 C06Dh	PORTD	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	790
0008 C06Eh	PORTE	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	790
0008 C06Fh	PORTF	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	790
0008 C070h	PORTG	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	790
0008 C072h	PORTJ	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	790
0008 C080h	PORT0	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	791
0008 C081h	PORT0	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	792
0008 C082h	PORT1	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	791
0008 C083h	PORT1	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	792
0008 C084h	PORT2	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	791
0008 C085h	PORT2	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	792
0008 C086h	PORT3	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	791
0008 C087h	PORT3	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	792
0008 C088h	PORT4	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	791
0008 C089h	PORT4	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	792
0008 C08Ah	PORT5	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	791
0008 C08Bh	PORT5	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	792
0008 C08Ch	PORT6	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	791
0008 C08Dh	PORT6	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	792
0008 C08Eh	PORT7	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	791
0008 C08Fh	PORT7	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	792
0008 C090h	PORT8	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	791
0008 C091h	PORT8	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	792
0008 C092h	PORT9	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	791
0008 C093h	PORT9	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	792
0008 C094h	PORTA	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	791

Table 5.1 List of I/O Registers (Address Order) (33 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 C095h	PORTA	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	792
0008 C096h	PORTB	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	791
0008 C097h	PORTB	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	792
0008 C098h	PORTC	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	791
0008 C099h	PORTC	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	792
0008 C09Ah	PORTD	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	791
0008 C09Bh	PORTD	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	792
0008 C09Ch	PORTE	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	791
0008 C09Dh	PORTE	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	792
0008 C09Eh	PORTF	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	791
0008 C09Fh	PORTF	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	792
0008 C0A0h	PORTG	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	791
0008 C0A1h	PORTG	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	792
0008 C0A4h	PORTJ	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	791
0008 C0A5h	PORTJ	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	792
0008 C0C0h	PORT0	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	793
0008 C0C1h	PORT1	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	793
0008 C0C2h	PORT2	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	793
0008 C0C3h	PORT3	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	793
0008 C0C4h	PORT4	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	793
0008 C0C5h	PORT5	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	793
0008 C0C6h	PORT6	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	793
0008 C0C7h	PORT7	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	793
0008 C0C8h	PORT8	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	793
0008 C0C9h	PORT9	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	793
0008 C0CAh	PORTA	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	793
0008 C0CBh	PORTB	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	793
0008 C0CCh	PORTC	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	793
0008 C0CDh	PORTD	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	793
0008 C0CEh	PORTE	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	793
0008 C0CFh	PORTF	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	793

Table 5.1 List of I/O Registers (Address Order) (34 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 C0D0h	PORTG	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	793
0008 C0D2h	PORTJ	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	793
0008 C0E0h	PORT0	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	794
0008 C0E2h	PORT2	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	794
0008 C0E5h	PORT5	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	794
0008 C0E9h	PORT9	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	794
0008 C0EAh	PORTA	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	794
0008 C0EBh	PORTB	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	794
0008 C0ECh	PORTC	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	794
0008 C0EDh	PORTD	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	794
0008 C0EEh	PORTE	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	794
0008 C0F0h	PORTG	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	794
0008 C100h	MPC	CS Output Enable Register	PFCSE	8	8	2, 3 PCLKB	2 ICLK	MPC	835
0008 C102h	MPC	CS Output Pin Select Register 0	PFCSS0	8	8	2, 3 PCLKB	2 ICLK	MPC	836
0008 C103h	MPC	CS Output Pin Select Register 1	PFCSS1	8	8	2, 3 PCLKB	2 ICLK	MPC	837
0008 C104h	MPC	Address Output Enable Register 0	PFAOE0	8	8	2, 3 PCLKB	2 ICLK	MPC	838
0008 C105h	MPC	Address Output Enable Register 1	PFAOE1	8	8	2, 3 PCLKB	2 ICLK	MPC	839
0008 C106h	MPC	External Bus Control Register 0	PFBCR0	8	8	2, 3 PCLKB	2 ICLK	MPC	840
0008 C107h	MPC	External Bus Control Register 1	PFBCR1	8	8	2, 3 PCLKB	2 ICLK	MPC	841
0008 C10Eh	MPC	Ethernet Control Register	PFENET	8	8	2, 3 PCLKB	2 ICLK	MPC	842
0008 C11Fh	MPC	Write-Protect Register	PWPR	8	8	2, 3 PCLKB	2 ICLK	MPC	814
0008 C140h	MPC	P00 Pin Function Control Register	P00PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	815
0008 C141h	MPC	P01 Pin Function Control Register	P01PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	815
0008 C142h	MPC	P02 Pin Function Control Register	P02PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	815
0008 C143h	MPC	P03 Pin Function Control Register	P03PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	815
0008 C145h	MPC	P05 Pin Function Control Register	P05PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	815
0008 C147h	MPC	P07 Pin Function Control Register	P07PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	815
0008 C148h	MPC	P10 Pin Function Control Register	P10PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	816
0008 C149h	MPC	P11 Pin Function Control Register	P11PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	816
0008 C14Ah	MPC	P12 Pin Function Control Register	P12PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	816
0008 C14Bh	MPC	P13 Pin Function Control Register	P13PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	816
0008 C14Ch	MPC	P14 Pin Function Control Register	P14PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	816
0008 C14Dh	MPC	P15 Pin Function Control Register	P15PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	816
0008 C14Eh	MPC	P16 Pin Function Control Register	P16PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	816
0008 C14Fh	MPC	P17 Pin Function Control Register	P17PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	816
0008 C150h	MPC	P20 Pin Function Control Register	P20PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	818
0008 C151h	MPC	P21 Pin Function Control Register	P21PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	818
0008 C152h	MPC	P22 Pin Function Control Register	P22PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	818
0008 C153h	MPC	P23 Pin Function Control Register	P23PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	818
0008 C154h	MPC	P24 Pin Function Control Register	P24PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	818
0008 C155h	MPC	P25 Pin Function Control Register	P25PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	818
0008 C156h	MPC	P26 Pin Function Control Register	P26PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	818
0008 C157h	MPC	P27 Pin Function Control Register	P27PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	818

Table 5.1 List of I/O Registers (Address Order) (35 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 C158h	MPC	P30 Pin Function Control Register	P30PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	819
0008 C159h	MPC	P31 Pin Function Control Register	P31PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	819
0008 C15Ah	MPC	P32 Pin Function Control Register	P32PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	819
0008 C15Bh	MPC	P33 Pin Function Control Register	P33PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	819
0008 C15Ch	MPC	P34 Pin Function Control Register	P34PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	819
0008 C160h	MPC	P40 Pin Function Control Register	P40PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	820
0008 C161h	MPC	P41 Pin Function Control Register	P41PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	820
0008 C162h	MPC	P42 Pin Function Control Register	P42PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	820
0008 C163h	MPC	P43 Pin Function Control Register	P43PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	820
0008 C164h	MPC	P44 Pin Function Control Register	P44PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	820
0008 C165h	MPC	P45 Pin Function Control Register	P45PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	820
0008 C166h	MPC	P46 Pin Function Control Register	P46PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	820
0008 C167h	MPC	P47 Pin Function Control Register	P47PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	820
0008 C168h	MPC	P50 Pin Function Control Register	P50PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	821
0008 C169h	MPC	P51 Pin Function Control Register	P51PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	821
0008 C16Ah	MPC	P52 Pin Function Control Register	P52PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	821
0008 C16Ch	MPC	P54 Pin Function Control Register	P54PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	821
0008 C16Dh	MPC	P55 Pin Function Control Register	P55PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	821
0008 C16Eh	MPC	P56 Pin Function Control Register	P56PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	821
0008 C170h	MPC	P60 Pin Function Control Register	P60PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	823
0008 C176h	MPC	P66 Pin Function Control Register	P66PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	823
0008 C177h	MPC	P67 Pin Function Control Register	P67PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	823
0008 C179h	MPC	P71 Pin Function Control Register	P71PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	824
0008 C17Ah	MPC	P72 Pin Function Control Register	P72PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	824
0008 C17Bh	MPC	P73 Pin Function Control Register	P73PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	824
0008 C17Ch	MPC	P74 Pin Function Control Register	P74PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	824
0008 C17Dh	MPC	P75 Pin Function Control Register	P75PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	824
0008 C17Eh	MPC	P76 Pin Function Control Register	P76PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	824
0008 C17Fh	MPC	P77 Pin Function Control Register	P77PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	824
0008 C180h	MPC	P80 Pin Function Control Register	P80PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	825
0008 C181h	MPC	P81 Pin Function Control Register	P81PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	825
0008 C182h	MPC	P82 Pin Function Control Register	P82PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	825
0008 C183h	MPC	P83 Pin Function Control Register	P83PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	825
0008 C186h	MPC	P86 Pin Function Control Register	P86PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	825
0008 C187h	MPC	P87 Pin Function Control Register	P87PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	825
0008 C188h	MPC	P90 Pin Function Control Register	P90PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	826
0008 C189h	MPC	P91 Pin Function Control Register	P91PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	826
0008 C18Ah	MPC	P92 Pin Function Control Register	P92PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	826
0008 C18Bh	MPC	P93 Pin Function Control Register	P93PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	826
0008 C18Ch	MPC	P94 Pin Function Control Register	P94PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	826
0008 C18Dh	MPC	P95 Pin Function Control Register	P95PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	826
0008 C18Eh	MPC	P96 Pin Function Control Register	P96PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	826
0008 C18Fh	MPC	P97 Pin Function Control Register	P97PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	826
0008 C190h	MPC	PA0 Pin Function Control Register	PA0PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	827
0008 C191h	MPC	PA1 Pin Function Control Register	PA1PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	827
0008 C192h	MPC	PA2 Pin Function Control Register	PA2PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	827
0008 C193h	MPC	PA3 Pin Function Control Register	PA3PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	827
0008 C194h	MPC	PA4 Pin Function Control Register	PA4PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	827
0008 C195h	MPC	PA5 Pin Function Control Register	PA5PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	827
0008 C196h	MPC	PA6 Pin Function Control Register	PA6PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	827

Table 5.1 List of I/O Registers (Address Order) (36 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 C197h	MPC	PA7 Pin Function Control Register	PA7PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	827
0008 C198h	MPC	PB0 Pin Function Control Register	PB0PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	828
0008 C199h	MPC	PB1 Pin Function Control Register	PB1PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	828
0008 C19Ah	MPC	PB2 Pin Function Control Register	PB2PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	828
0008 C19Bh	MPC	PB3 Pin Function Control Register	PB3PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	828
0008 C19Ch	MPC	PB4 Pin Function Control Register	PB4PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	828
0008 C19Dh	MPC	PB5 Pin Function Control Register	PB5PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	828
0008 C19Eh	MPC	PB6 Pin Function Control Register	PB6PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	828
0008 C19Fh	MPC	PB7 Pin Function Control Register	PB7PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	828
0008 C1A0h	MPC	PC0 Pin Function Control Register	PC0PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	830
0008 C1A1h	MPC	PC1 Pin Function Control Register	PC1PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	830
0008 C1A2h	MPC	PC2 Pin Function Control Register	PC2PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	830
0008 C1A3h	MPC	PC3 Pin Function Control Register	PC3PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	830
0008 C1A4h	MPC	PC4 Pin Function Control Register	PC4PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	830
0008 C1A5h	MPC	PC5 Pin Function Control Register	PC5PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	830
0008 C1A6h	MPC	PC6 Pin Function Control Register	PC6PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	830
0008 C1A7h	MPC	PC7 Pin Function Control Register	PC7PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	830
0008 C1A8h	MPC	PD0 Pin Function Control Register	PD0PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	831
0008 C1A9h	MPC	PD1 Pin Function Control Register	PD1PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	831
0008 C1AAh	MPC	PD2 Pin Function Control Register	PD2PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	831
0008 C1ABh	MPC	PD3 Pin Function Control Register	PD3PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	831
0008 C1ACh	MPC	PD4 Pin Function Control Register	PD4PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	831
0008 C1ADh	MPC	PD5 Pin Function Control Register	PD5PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	831
0008 C1AEh	MPC	PD6 Pin Function Control Register	PD6PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	831
0008 C1AFh	MPC	PD7 Pin Function Control Register	PD7PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	831
0008 C1B0h	MPC	PE0 Pin Function Control Register	PE0PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	832
0008 C1B1h	MPC	PE1 Pin Function Control Register	PE1PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	832
0008 C1B2h	MPC	PE2 Pin Function Control Register	PE2PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	832
0008 C1B3h	MPC	PE3 Pin Function Control Register	PE3PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	832
0008 C1B4h	MPC	PE4 Pin Function Control Register	PE4PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	832
0008 C1B5h	MPC	PE5 Pin Function Control Register	PE5PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	832
0008 C1B6h	MPC	PE6 Pin Function Control Register	PE6PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	832
0008 C1B7h	MPC	PE7 Pin Function Control Register	PE7PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	832
0008 C1B8h	MPC	PF0 Pin Function Control Register	PF0PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	833
0008 C1B9h	MPC	PF1 Pin Function Control Register	PF1PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	833
0008 C1BAh	MPC	PF2 Pin Function Control Register	PF2PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	833
0008 C1BDh	MPC	PF5 Pin Function Control Register	PF5PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	833
0008 C1C0h	MPC	PG0 Pin Function Control Register	PG0PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	833
0008 C1C1h	MPC	PG1 Pin Function Control Register	PG1PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	833
0008 C1C2h	MPC	PG2 Pin Function Control Register	PG2PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	833
0008 C1C3h	MPC	PG3 Pin Function Control Register	PG3PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	833
0008 C1C4h	MPC	PG4 Pin Function Control Register	PG4PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	833
0008 C1C5h	MPC	PG5 Pin Function Control Register	PG5PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	833
0008 C1C6h	MPC	PG6 Pin Function Control Register	PG6PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	833
0008 C1C7h	MPC	PG7 Pin Function Control Register	PG7PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	833
0008 C1D3h	MPC	PJ3 Pin Function Control Register	PJ3PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	834
0008 C1D5h	MPC	PJ5 Pin Function Control Register	PJ5PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	834
0008 C280h	SYSTEM	Deep Standby Control Register	DPSBYCR	8	8	4, 5 PCLKB	2, 3 ICLK	Low Power Consumption	361

Table 5.1 List of I/O Registers (Address Order) (37 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 C282h	SYSTEM	Deep Standby Interrupt Enable Register 0	DPSIER0	8	8	4, 5 PCLKB	2, 3 ICLK	Low Power Consumption	363
0008 C283h	SYSTEM	Deep Standby Interrupt Enable Register 1	DPSIER1	8	8	4, 5 PCLKB	2, 3 ICLK	Low Power Consumption	364
0008 C284h	SYSTEM	Deep Standby Interrupt Enable Register 2	DPSIER2	8	8	4, 5 PCLKB	2, 3 ICLK	Low Power Consumption	365
0008 C285h	SYSTEM	Deep Standby Interrupt Enable Register 3	DPSIER3	8	8	4, 5 PCLKB	2, 3 ICLK	Low Power Consumption	366
0008 C286h	SYSTEM	Deep Standby Interrupt Flag Register 0	DPSIFR0	8	8	4, 5 PCLKB	2, 3 ICLK	Low Power Consumption	367
0008 C287h	SYSTEM	Deep Standby Interrupt Flag Register 1	DPSIFR1	8	8	4, 5 PCLKB	2, 3 ICLK	Low Power Consumption	368
0008 C288h	SYSTEM	Deep Standby Interrupt Flag Register 2	DPSIFR2	8	8	4, 5 PCLKB	2, 3 ICLK	Low Power Consumption	369
0008 C289h	SYSTEM	Deep Standby Interrupt Flag Register 3	DPSIFR3	8	8	4, 5 PCLKB	2, 3 ICLK	Low Power Consumption	372
0008 C28Ah	SYSTEM	Deep Standby Interrupt Edge Register 0	DPSIEGR0	8	8	4, 5 PCLKB	2, 3 ICLK	Low Power Consumption	373
0008 C28Bh	SYSTEM	Deep Standby Interrupt Edge Register 1	DPSIEGR1	8	8	4, 5 PCLKB	2, 3 ICLK	Low Power Consumption	374
0008 C28Ch	SYSTEM	Deep Standby Interrupt Edge Register 2	DPSIEGR2	8	8	4, 5 PCLKB	2, 3 ICLK	Low Power Consumption	375
0008 C28Dh	SYSTEM	Deep Standby Interrupt Edge Register 3	DPSIEGR3	8	8	4, 5 PCLKB	2, 3 ICLK	Low Power Consumption	375
0008 C290h	SYSTEM	Reset Status Register 0	RSTSR0	8	8	4, 5 PCLKB	2, 3 ICLK	Resets	243
0008 C291h	SYSTEM	Reset Status Register 1	RSTSR1	8	8	4, 5 PCLKB	2, 3 ICLK	Resets	245
0008 C293h	SYSTEM	Main Clock Oscillator Forced Oscillation Control Register	MOFCR	8	8	4, 5 PCLKB	2, 3 ICLK	Clock Generation Circuit	315
0008 C294h	SYSTEM	High-Speed On-Chip Oscillator Power Supply Control Register	HOCOPCR	8	8	4, 5 PCLKB	2, 3 ICLK	Clock Generation Circuit	316
0008 C296h	FLASH	Flash P/E Protect Register	FWEPROR	8	8	2 ICLK		Flash	2751
0008 C297h	SYSTEM	Voltage Monitoring Circuit Control Register	LVCMPCR	8	8	4, 5 PCLKB	2, 3 ICLK	LVDA	277
0008 C298h	SYSTEM	Voltage Detection Level Select Register	LVDLVLR	8	8	4, 5 PCLKB	2, 3 ICLK	LVDA	278
0008 C29Ah	SYSTEM	Voltage Monitoring 1 Circuit Control Register 0	LVD1CR0	8	8	4, 5 PCLKB	2, 3 ICLK	LVDA	279
0008 C29Bh	SYSTEM	Voltage Monitoring 2 Circuit Control Register 0	LVD2CR0	8	8	4, 5 PCLKB	2, 3 ICLK	LVDA	280

Table 5.1 List of I/O Registers (Address Order) (38 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 C2A0h to 0008 C2BFh	SYSTEM	Deep Standby Backup Registers 0 to 31	DPSBKR0 to 31	8	8	4, 5 PCLKB	2, 3 ICLK	Low Power Consumption	376
0008 C400h	RTC	64-Hz Counter	R64CNT	8	8	2, 3 PCLKB	2 ICLK	RTCd	1438
0008 C402h	RTC	Second Counter	RSECCNT	8	8	2, 3 PCLKB	2 ICLK	RTCd	1439
0008 C402h	RTC	Binary Counter 0	BCNT0	8	8	2, 3 PCLKB	2 ICLK	RTCd	1439
0008 C404h	RTC	Minute Counter	RMINCNT	8	8	2, 3 PCLKB	2 ICLK	RTCd	1440
0008 C404h	RTC	Binary Counter 1	BCNT1	8	8	2, 3 PCLKB	2 ICLK	RTCd	1440
0008 C406h	RTC	Hour Counter	RHRCNT	8	8	2, 3 PCLKB	2 ICLK	RTCd	1441
0008 C406h	RTC	Binary Counter 2	BCNT2	8	8	2, 3 PCLKB	2 ICLK	RTCd	1441
0008 C408h	RTC	Day-of-Week Counter	RWKCNT	8	8	2, 3 PCLKB	2 ICLK	RTCd	1442
0008 C408h	RTC	Binary Counter 3	BCNT3	8	8	2, 3 PCLKB	2 ICLK	RTCd	1442
0008 C40Ah	RTC	Date Counter	RDAYCNT	8	8	2, 3 PCLKB	2 ICLK	RTCd	1443
0008 C40Ch	RTC	Month Counter	RMONCNT	8	8	2, 3 PCLKB	2 ICLK	RTCd	1444
0008 C40Eh	RTC	Year Counter	RYRCNT	16	16	2, 3 PCLKB	2 ICLK	RTCd	1445
0008 C410h	RTC	Second Alarm Register	RSECAR	8	8	2, 3 PCLKB	2 ICLK	RTCd	1446
0008 C410h	RTC	Binary Counter 0 Alarm Register	BCNT0AR	8	8	2, 3 PCLKB	2 ICLK	RTCd	1446
0008 C412h	RTC	Minute Alarm Register	RMINAR	8	8	2, 3 PCLKB	2 ICLK	RTCd	1447
0008 C412h	RTC	Binary Counter 1 Alarm Register	BCNT1AR	8	8	2, 3 PCLKB	2 ICLK	RTCd	1447
0008 C414h	RTC	Hour Alarm Register	RHRAR	8	8	2, 3 PCLKB	2 ICLK	RTCd	1448
0008 C414h	RTC	Binary Counter 2 Alarm Register	BCNT2AR	8	8	2, 3 PCLKB	2 ICLK	RTCd	1448
0008 C416h	RTC	Day-of-Week Alarm Register	RWKAR	8	8	2, 3 PCLKB	2 ICLK	RTCd	1449
0008 C416h	RTC	Binary Counter 3 Alarm Register	BCNT3AR	8	8	2, 3 PCLKB	2 ICLK	RTCd	1449
0008 C418h	RTC	Date Alarm Register	RDAYAR	8	8	2, 3 PCLKB	2 ICLK	RTCd	1450
0008 C418h	RTC	Binary Counter 0 Alarm Enable Register	BCNT0AER	8	8	2, 3 PCLKB	2 ICLK	RTCd	1450
0008 C41Ah	RTC	Month Alarm Register	RMONAR	8	8	2, 3 PCLKB	2 ICLK	RTCd	1451
0008 C41Ah	RTC	Binary Counter 1 Alarm Enable Register	BCNT1AER	8	8	2, 3 PCLKB	2 ICLK	RTCd	1451
0008 C41Ch	RTC	Year Alarm Register	RYRAR	16	16	2, 3 PCLKB	2 ICLK	RTCd	1452
0008 C41Ch	RTC	Binary Counter 2 Alarm Enable Register	BCNT2AER	16	16	2, 3 PCLKB	2 ICLK	RTCd	1452
0008 C41Eh	RTC	Year Alarm Enable Register	RYRAREN	8	8	2, 3 PCLKB	2 ICLK	RTCd	1453
0008 C41Eh	RTC	Binary Counter 3 Alarm Enable Register	BCNT3AER	8	8	2, 3 PCLKB	2 ICLK	RTCd	1453
0008 C422h	RTC	RTC Control Register 1	RCR1	8	8	2, 3 PCLKB	2 ICLK	RTCd	1454
0008 C424h	RTC	RTC Control Register 2	RCR2	8	8	2, 3 PCLKB	2 ICLK	RTCd	1456
0008 C426h	RTC	RTC Control Register 3	RCR3	8	8	2, 3 PCLKB	2 ICLK	RTCd	1458
0008 C428h	RTC	RTC Control Register 4	RCR4	8	8	2, 3 PCLKB	2 ICLK	RTCd	1460
0008 C42Ah	RTC	Frequency Register H	RFRH	16	16	2, 3 PCLKB	2 ICLK	RTCd	1460
0008 C42Ch	RTC	Frequency Register L	RFRL	16	16	2, 3 PCLKB	2 ICLK	RTCd	1460
0008 C42Eh	RTC	Time Error Adjustment Register	RADJ	8	8	2, 3 PCLKB	2 ICLK	RTCd	1462
0008 C440h	RTC	Time Capture Control Register 0	RTCCR0	8	8	2, 3 PCLKB	2 ICLK	RTCd	1463
0008 C442h	RTC	Time Capture Control Register 1	RTCCR1	8	8	2, 3 PCLKB	2 ICLK	RTCd	1463
0008 C444h	RTC	Time Capture Control Register 2	RTCCR2	8	8	2, 3 PCLKB	2 ICLK	RTCd	1463
0008 C452h	RTC	Second Capture Register 0	RSECCP0	8	8	2, 3 PCLKB	2 ICLK	RTCd	1465
0008 C452h	RTC	BCNT0 Capture Register 0	BCNT0CP0	8	8	2, 3 PCLKB	2 ICLK	RTCd	1465
0008 C454h	RTC	Minute Capture Register 0	RMINCP0	8	8	2, 3 PCLKB	2 ICLK	RTCd	1466
0008 C454h	RTC	BCNT1 Capture Register 0	BCNT1CP0	8	8	2, 3 PCLKB	2 ICLK	RTCd	1466
0008 C456h	RTC	Hour Capture Register 0	RHRCP0	8	8	2, 3 PCLKB	2 ICLK	RTCd	1467
0008 C456h	RTC	BCNT2 Capture Register 0	BCNT2CP0	8	8	2, 3 PCLKB	2 ICLK	RTCd	1467
0008 C45Ah	RTC	Date Capture Register 0	RDAYCP0	8	8	2, 3 PCLKB	2 ICLK	RTCd	1468
0008 C45Ah	RTC	BCNT3 Capture Register 0	BCNT3CP0	8	8	2, 3 PCLKB	2 ICLK	RTCd	1468
0008 C45Ch	RTC	Month Capture Register 0	RMONCP0	8	8	2, 3 PCLKB	2 ICLK	RTCd	1469

Table 5.1 List of I/O Registers (Address Order) (39 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 C462h	RTC	Second Capture Register 1	RSECCP1	8	8	2, 3 PCLKB	2 ICLK	RTCd	1465
0008 C462h	RTC	BCNT0 Capture Register 1	BCNT0CP1	8	8	2, 3 PCLKB	2 ICLK	RTCd	1465
0008 C464h	RTC	Minute Capture Register 1	RMINCP1	8	8	2, 3 PCLKB	2 ICLK	RTCd	1466
0008 C464h	RTC	BCNT1 Capture Register 1	BCNT1CP1	8	8	2, 3 PCLKB	2 ICLK	RTCd	1466
0008 C466h	RTC	Hour Capture Register 1	RHRCP1	8	8	2, 3 PCLKB	2 ICLK	RTCd	1467
0008 C466h	RTC	BCNT2 Capture Register 1	BCNT2CP1	8	8	2, 3 PCLKB	2 ICLK	RTCd	1467
0008 C46Ah	RTC	Date Capture Register 1	RDAYCP1	8	8	2, 3 PCLKB	2 ICLK	RTCd	1468
0008 C46Ah	RTC	BCNT3 Capture Register 1	BCNT3CP1	8	8	2, 3 PCLKB	2 ICLK	RTCd	1468
0008 C46Ch	RTC	Month Capture Register 1	RMONCP1	8	8	2, 3 PCLKB	2 ICLK	RTCd	1469
0008 C472h	RTC	Second Capture Register 2	RSECCP2	8	8	2, 3 PCLKB	2 ICLK	RTCd	1465
0008 C472h	RTC	BCNT0 Capture Register 2	BCNT0CP2	8	8	2, 3 PCLKB	2 ICLK	RTCd	1465
0008 C474h	RTC	Minute Capture Register 2	RMINCP2	8	8	2, 3 PCLKB	2 ICLK	RTCd	1466
0008 C474h	RTC	BCNT1 Capture Register 2	BCNT1CP2	8	8	2, 3 PCLKB	2 ICLK	RTCd	1466
0008 C476h	RTC	Hour Capture Register 2	RHRCP2	8	8	2, 3 PCLKB	2 ICLK	RTCd	1467
0008 C476h	RTC	BCNT2 Capture Register 2	BCNT2CP2	8	8	2, 3 PCLKB	2 ICLK	RTCd	1467
0008 C47Ah	RTC	Date Capture Register 2	RDAYCP2	8	8	2, 3 PCLKB	2 ICLK	RTCd	1468
0008 C47Ah	RTC	BCNT3 Capture Register 2	BCNT3CP2	8	8	2, 3 PCLKB	2 ICLK	RTCd	1468
0008 C47Ch	RTC	Month Capture Register 2	RMONCP2	8	8	2, 3 PCLKB	2 ICLK	RTCd	1469
0008 C4C0h	POE3	Input Level Control/Status Register 1	ICSR1	16	16	2, 3 PCLKB	2 ICLK	POE3	1089
0008 C4C2h	POE3	Output Level Control/Status Register 1	OCSR1	16	16	2, 3 PCLKB	2 ICLK	POE3	1095
0008 C4C4h	POE3	Input Level Control/Status Register 2	ICSR2	16	16	2, 3 PCLKB	2 ICLK	POE3	1090
0008 C4C6h	POE3	Output Level Control/Status Register 2	OCSR2	16	16	2, 3 PCLKB	2 ICLK	POE3	1096
0008 C4C8h	POE3	Input Level Control/Status Register 3	ICSR3	16	16	2, 3 PCLKB	2 ICLK	POE3	1091
0008 C4CAh	POE3	Software Port Output Enable Register	SPOER	8	8	2, 3 PCLKB	2 ICLK	POE3	1099
0008 C4CBh	POE3	Port Output Enable Control Register 1	POECR1	8	8	2, 3 PCLKB	2 ICLK	POE3	1101
0008 C4CCh	POE3	Port Output Enable Control Register 2	POECR2	16	16	2, 3 PCLKB	2 ICLK	POE3	1102
0008 C4CEh	POE3	Port Output Enable Control Register 3	POECR3	16	16	2, 3 PCLKB	2 ICLK	POE3	1104
0008 C4D0h	POE3	Port Output Enable Control Register 4	POECR4	16	16	2, 3 PCLKB	2 ICLK	POE3	1105
0008 C4D2h	POE3	Port Output Enable Control Register 5	POECR5	16	16	2, 3 PCLKB	2 ICLK	POE3	1107
0008 C4D4h	POE3	Port Output Enable Control Register 6	POECR6	16	16	2, 3 PCLKB	2 ICLK	POE3	1108
0008 C4D6h	POE3	Input Level Control/Status Register 4	ICSR4	16	16	2, 3 PCLKB	2 ICLK	POE3	1092
0008 C4D8h	POE3	Input Level Control/Status Register 5	ICSR5	16	16	2, 3 PCLKB	2 ICLK	POE3	1093
0008 C4DAh	POE3	Active Level Setting Register 1	ALR1	16	16	2, 3 PCLKB	2 ICLK	POE3	1097
0008 C4DCh	POE3	Input Level Control/Status Register 6	ICSR6	16	16	2, 3 PCLKB	2 ICLK	POE3	1094
0008 C4E0h	POE3	GPT0 Pin Select Register	G0SELR	8	8	2, 3 PCLKB	2 ICLK	POE3	1110
0008 C4E1h	POE3	GPT1 Pin Select Register	G1SELR	8	8	2, 3 PCLKB	2 ICLK	POE3	1111
0008 C4E2h	POE3	GPT2 Pin Select Register	G2SELR	8	8	2, 3 PCLKB	2 ICLK	POE3	1112
0008 C4E3h	POE3	GPT3 Pin Select Register	G3SELR	8	8	2, 3 PCLKB	2 ICLK	POE3	1113
0008 C4E4h	POE3	MTU0 Pin Select Register 1	M0SELR1	8	8	2, 3 PCLKB	2 ICLK	POE3	1114
0008 C4E5h	POE3	MTU0 Pin Select Register 2	M0SELR2	8	8	2, 3 PCLKB	2 ICLK	POE3	1115
0008 C4E6h	POE3	MTU3 Pin Select Register	M3SELR	8	8	2, 3 PCLKB	2 ICLK	POE3	1116
0008 C4E7h	POE3	MTU4 Pin Select Register 1	M4SELR1	8	8	2, 3 PCLKB	2 ICLK	POE3	1117
0008 C4E8h	POE3	MTU4 Pin Select Register 2	M4SELR2	8	8	2, 3 PCLKB	2 ICLK	POE3	1118
0008 C4E9h	POE3	MTU/GPT Pin Select Register	MGSELR	8	8	2, 3 PCLKB	2 ICLK	POE3	1119
0008 C500h	TEMPS	Temperature Sensor Control Register	TSCR	8	8	2, 3 PCLKB	2 ICLK	TEMPS	2721
0008 C5C0h	DA	D/A A/D Synchronous Unit Select Register	DAADUSR	8	8	2, 3 PCLKB	2 ICLK	R12DA	2714
0009 0200h to 0009 03FFh	CAN0	Mailbox Registers 0 to 31	MB0 to 31	128	8, 16, 32*6	2, 3 PCLKB	2 ICLK	CAN	2236
0009 0400h to 0009 041Fh	CAN0	Mask Registers 0 to 7	MKR0 to 7	32	8, 16, 32	2, 3 PCLKB	2 ICLK	CAN	2233

Table 5.1 List of I/O Registers (Address Order) (40 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0009 0420h	CAN0	FIFO Received ID Compare Register 0	FIDCR0	32	8, 16, 32	2, 3 PCLKB	2 ICLK	CAN	2234
0009 0424h	CAN0	FIFO Received ID Compare Register 1	FIDCR1	32	8, 16, 32	2, 3 PCLKB	2 ICLK	CAN	2234
0009 0428h	CAN0	Mask Invalid Register	MKIVLR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	CAN	2235
0009 042Ch	CAN0	Mailbox Interrupt Enable Register	MIER	32	8, 16, 32	2, 3 PCLKB	2 ICLK	CAN	2240
0009 0820h to 0009 083Fh	CAN0	Message Control Registers 0 to 31	MCTL0 to 31	8	8	2, 3 PCLKB	2 ICLK	CAN	2241
0009 0840h	CAN0	Control Register	CTLR	16	8, 16	2, 3 PCLKB	2 ICLK	CAN	2228
0009 0842h	CAN0	Status Register	STR	16	8, 16	2, 3 PCLKB	2 ICLK	CAN	2250
0009 0844h	CAN0	Bit Configuration Register	BCR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	CAN	2231
0009 0848h	CAN0	Receive FIFO Control Register	RFCR	8	8	2, 3 PCLKB	2 ICLK	CAN	2244
0009 0849h	CAN0	Receive FIFO Pointer Control Register	RFPCR	8	8	2, 3 PCLKB	2 ICLK	CAN	2247
0009 084Ah	CAN0	Transmit FIFO Control Register	TFCR	8	8	2, 3 PCLKB	2 ICLK	CAN	2247
0009 084Bh	CAN0	Transmit FIFO Pointer Control Register	TFPCR	8	8	2, 3 PCLKB	2 ICLK	CAN	2249
0009 084Ch	CAN0	Error Interrupt Enable Register	EIER	8	8	2, 3 PCLKB	2 ICLK	CAN	2256
0009 084Dh	CAN0	Error Interrupt Factor Judge Register	EIFR	8	8	2, 3 PCLKB	2 ICLK	CAN	2257
0009 084Eh	CAN0	Receive Error Count Register	RECR	8	8	2, 3 PCLKB	2 ICLK	CAN	2259
0009 084Fh	CAN0	Transmit Error Count Register	TECR	8	8	2, 3 PCLKB	2 ICLK	CAN	2260
0009 0850h	CAN0	Error Code Store Register	ECSR	8	8	2, 3 PCLKB	2 ICLK	CAN	2260
0009 0851h	CAN0	Channel Search Support Register	CSSR	8	8	2, 3 PCLKB	2 ICLK	CAN	2254
0009 0852h	CAN0	Mailbox Search Status Register	MSSR	8	8	2, 3 PCLKB	2 ICLK	CAN	2253
0009 0853h	CAN0	Mailbox Search Mode Register	MSMR	8	8	2, 3 PCLKB	2 ICLK	CAN	2252
0009 0854h	CAN0	Time Stamp Register	TSR	16	16	2, 3 PCLKB	2 ICLK	CAN	2261
0009 0856h	CAN0	Acceptance Filter Support Register	AFSR	16	8, 16	2, 3 PCLKB	2 ICLK	CAN	2255
0009 0858h	CAN0	Test Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	CAN	2262
0009 1200h to 0009 13FFh	CAN1	Mailbox Registers 0 to 31	MB0 to 31	128	8, 16, 32*6	2, 3 PCLKB	2 ICLK	CAN	2236
0009 1400h to 0009 141Fh	CAN1	Mask Registers 0 to 7	MKR0 to 7	32	8, 16, 32	2, 3 PCLKB	2 ICLK	CAN	2233
0009 1420h	CAN1	FIFO Received ID Compare Register 0	FIDCR0	32	8, 16, 32	2, 3 PCLKB	2 ICLK	CAN	2234
0009 1424h	CAN1	FIFO Received ID Compare Register 1	FIDCR1	32	8, 16, 32	2, 3 PCLKB	2 ICLK	CAN	2234
0009 1428h	CAN1	Mask Invalid Register	MKIVLR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	CAN	2235
0009 142Ch	CAN1	Mailbox Interrupt Enable Register	MIER	32	8, 16, 32	2, 3 PCLKB	2 ICLK	CAN	2240
0009 1820h to 0009 183Fh	CAN1	Message Control Registers 0 to 31	MCTL0 to 31	8	8	2, 3 PCLKB	2 ICLK	CAN	2241
0009 1840h	CAN1	Control Register	CTLR	16	8, 16	2, 3 PCLKB	2 ICLK	CAN	2228
0009 1842h	CAN1	Status Register	STR	16	8, 16	2, 3 PCLKB	2 ICLK	CAN	2250
0009 1844h	CAN1	Bit Configuration Register	BCR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	CAN	2231
0009 1848h	CAN1	Receive FIFO Control Register	RFCR	8	8	2, 3 PCLKB	2 ICLK	CAN	2244
0009 1849h	CAN1	Receive FIFO Pointer Control Register	RFPCR	8	8	2, 3 PCLKB	2 ICLK	CAN	2247
0009 184Ah	CAN1	Transmit FIFO Control Register	TFCR	8	8	2, 3 PCLKB	2 ICLK	CAN	2247
0009 184Bh	CAN1	Transmit FIFO Pointer Control Register	TFPCR	8	8	2, 3 PCLKB	2 ICLK	CAN	2249
0009 184Ch	CAN1	Error Interrupt Enable Register	EIER	8	8	2, 3 PCLKB	2 ICLK	CAN	2256
0009 184Dh	CAN1	Error Interrupt Factor Judge Register	EIFR	8	8	2, 3 PCLKB	2 ICLK	CAN	2257
0009 184Eh	CAN1	Receive Error Count Register	RECR	8	8	2, 3 PCLKB	2 ICLK	CAN	2259
0009 184Fh	CAN1	Transmit Error Count Register	TECR	8	8	2, 3 PCLKB	2 ICLK	CAN	2260
0009 1850h	CAN1	Error Code Store Register	ECSR	8	8	2, 3 PCLKB	2 ICLK	CAN	2260
0009 1851h	CAN1	Channel Search Support Register	CSSR	8	8	2, 3 PCLKB	2 ICLK	CAN	2254

Table 5.1 List of I/O Registers (Address Order) (41 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0009 1852h	CAN1	Mailbox Search Status Register	MSSR	8	8	2, 3 PCLKB	2 ICLK	CAN	2253
0009 1853h	CAN1	Mailbox Search Mode Register	MSMR	8	8	2, 3 PCLKB	2 ICLK	CAN	2252
0009 1854h	CAN1	Time Stamp Register	TSR	16	16	2, 3 PCLKB	2 ICLK	CAN	2261
0009 1856h	CAN1	Acceptance Filter Support Register	AFSR	16	8, 16	2, 3 PCLKB	2 ICLK	CAN	2255
0009 1858h	CAN1	Test Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	CAN	2262
0009 2200h to 0009 23FFh	CAN2	Mailbox Registers 0 to 31	MB0 to 31	128	8, 16, 32*6	2, 3 PCLKB	2 ICLK	CAN	2236
0009 2400h to 0009 241Fh	CAN2	Mask Registers 0 to 7	MKR0 to 7	32	8, 16, 32	2, 3 PCLKB	2 ICLK	CAN	2233
0009 2420h	CAN2	FIFO Received ID Compare Register 0	FIDCR0	32	8, 16, 32	2, 3 PCLKB	2 ICLK	CAN	2234
0009 2424h	CAN2	FIFO Received ID Compare Register 1	FIDCR1	32	8, 16, 32	2, 3 PCLKB	2 ICLK	CAN	2234
0009 2428h	CAN2	Mask Invalid Register	MKIVLR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	CAN	2235
0009 242Ch	CAN2	Mailbox Interrupt Enable Register	MIER	32	8, 16, 32	2, 3 PCLKB	2 ICLK	CAN	2240
0009 2820h to 0009 283Fh	CAN2	Message Control Registers 0 to 31	MCTL0 to 31	8	8	2, 3 PCLKB	2 ICLK	CAN	2241
0009 2840h	CAN2	Control Register	CTLR	16	8, 16	2, 3 PCLKB	2 ICLK	CAN	2228
0009 2842h	CAN2	Status Register	STR	16	8, 16	2, 3 PCLKB	2 ICLK	CAN	2250
0009 2844h	CAN2	Bit Configuration Register	BCR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	CAN	2231
0009 2848h	CAN2	Receive FIFO Control Register	RFCR	8	8	2, 3 PCLKB	2 ICLK	CAN	2247
0009 2849h	CAN2	Receive FIFO Pointer Control Register	RFPCR	8	8	2, 3 PCLKB	2 ICLK	CAN	2247
0009 284Ah	CAN2	Transmit FIFO Control Register	TFCR	8	8	2, 3 PCLKB	2 ICLK	CAN	2249
0009 284Bh	CAN2	Transmit FIFO Pointer Control Register	TFPCR	8	8	2, 3 PCLKB	2 ICLK	CAN	2247
0009 284Ch	CAN2	Error Interrupt Enable Register	EIER	8	8	2, 3 PCLKB	2 ICLK	CAN	2256
0009 284Dh	CAN2	Error Interrupt Factor Judge Register	EIFR	8	8	2, 3 PCLKB	2 ICLK	CAN	2257
0009 284Eh	CAN2	Receive Error Count Register	RECR	8	8	2, 3 PCLKB	2 ICLK	CAN	2259
0009 284Fh	CAN2	Transmit Error Count Register	TECR	8	8	2, 3 PCLKB	2 ICLK	CAN	2260
0009 2850h	CAN2	Error Code Store Register	ECSR	8	8	2, 3 PCLKB	2 ICLK	CAN	2260
0009 2851h	CAN2	Channel Search Support Register	CSSR	8	8	2, 3 PCLKB	2 ICLK	CAN	2254
0009 2852h	CAN2	Mailbox Search Status Register	MSSR	8	8	2, 3 PCLKB	2 ICLK	CAN	2253
0009 2853h	CAN2	Mailbox Search Mode Register	MSMR	8	8	2, 3 PCLKB	2 ICLK	CAN	2252
0009 2854h	CAN2	Time Stamp Register	TSR	16	8, 16	2, 3 PCLKB	2 ICLK	CAN	2261
0009 2856h	CAN2	Acceptance Filter Support Register	AFSR	16	8, 16	2, 3 PCLKB	2 ICLK	CAN	2255
0009 2858h	CAN2	Test Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	CAN	2262
0009 4200h	CMTW0	Timer Start Register	CMWSTR	16	16	2, 3 PCLKB	2 ICLK	CMTW	1412
0009 4204h	CMTW0	Timer Control Register	CMWCR	16	16	2, 3 PCLKB	2 ICLK	CMTW	1413
0009 4208h	CMTW0	Timer I/O Control Register	CMWIOR	16	16	2, 3 PCLKB	2 ICLK	CMTW	1415
0009 4210h	CMTW0	Timer Counter	CMWCNT	32	32	2, 3 PCLKB	2 ICLK	CMTW	1416
0009 4214h	CMTW0	Compare Match Constant Register	CMWCOR	32	32	2, 3 PCLKB	2 ICLK	CMTW	1416
0009 4218h	CMTW0	Input Capture Register 0	CMWICR0	32	32	2, 3 PCLKB	2 ICLK	CMTW	1417
0009 421Ch	CMTW0	Input Capture Register 1	CMWICR1	32	32	2, 3 PCLKB	2 ICLK	CMTW	1417
0009 4220h	CMTW0	Output Compare Register 0	CMWOCR0	32	32	2, 3 PCLKB	2 ICLK	CMTW	1417
0009 4224h	CMTW0	Output Compare Register 1	CMWOCR1	32	32	2, 3 PCLKB	2 ICLK	CMTW	1417
0009 4280h	CMTW1	Timer Start Register	CMWSTR	16	16	2, 3 PCLKB	2 ICLK	CMTW	1412
0009 4284h	CMTW1	Timer Control Register	CMWCR	16	16	2, 3 PCLKB	2 ICLK	CMTW	1413
0009 4288h	CMTW1	Timer I/O Control Register	CMWIOR	16	16	2, 3 PCLKB	2 ICLK	CMTW	1415
0009 4290h	CMTW1	Timer Counter	CMWCNT	32	32	2, 3 PCLKB	2 ICLK	CMTW	1416
0009 4294h	CMTW1	Compare Match Constant Register	CMWCOR	32	32	2, 3 PCLKB	2 ICLK	CMTW	1416
0009 4298h	CMTW1	Input Capture Register 0	CMWICR0	32	32	2, 3 PCLKB	2 ICLK	CMTW	1417
0009 429Ch	CMTW1	Input Capture Register 1	CMWICR1	32	32	2, 3 PCLKB	2 ICLK	CMTW	1417

Table 5.1 List of I/O Registers (Address Order) (42 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0009 42A0h	CMTW1	Output Compare Register 0	CMWOCR0	32	32	2, 3 PCLKB	2 ICLK	CMTW	1417
0009 42A4h	CMTW1	Output Compare Register 1	CMWOCR1	32	32	2, 3 PCLKB	2 ICLK	CMTW	1417
0009 8000h to 0009 D6BFh	SRC	Filter Coefficient Table	SRCFCTR0 to 5551	32	32	4, 5 PCLKB	2, 3 ICLK	SRC	2457
0009 DFF0h	SRC	Input Data Register	SRCID	32	32	5, 6 PCLKB	2, 3 ICLK	SRC	2448
0009 DFF4h	SRC	Output Data Register	SRCOD	32	32	5, 6 PCLKB	2, 3 ICLK	SRC	2449
0009 DFF8h	SRC	Input Data Control Register	SRCIDCTRL	16	16	4, 5 PCLKB	2, 3 ICLK	SRC	2450
0009 DFFAh	SRC	Output Data Control Register	SRCODCTRL	16	16	4, 5 PCLKB	2, 3 ICLK	SRC	2451
0009 DFFCh	SRC	Control Register	SRCCTRL	16	16	4, 5 PCLKB	2, 3 ICLK	SRC	2452
0009 DFFEh	SRC	Status Register	SRCSTAT	16	16	4, 5 PCLKB	2, 3 ICLK	SRC	2455
000A 0000h	USB0	System Configuration Control Register	SYSCFG	16	16	3, 4 PCLKB	2 ICLK	USBb	1731
000A 0004h	USB0	System Configuration Status Register 0	SYSSTS0	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9 × (frequency ratio of ICLK/ PCLKB)*5	USBb	1733
000A 0008h	USB0	Device State Control Register 0	DVSTCTR0	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9 × (frequency ratio of ICLK/ PCLKB)*5	USBb	1735
000A 0014h	USB0	CFIFO Port Register	CFIFO	16	8, 16	3, 4 PCLKB	2 ICLK	USBb	1738
000A 0018h	USB0	D0FIFO Port Register	D0FIFO	16	8, 16	3, 4 PCLKB	2 ICLK	USBb	1738
000A 001Ch	USB0	D1FIFO Port Register	D1FIFO	16	8, 16	3, 4 PCLKB	2 ICLK	USBb	1738
000A 0020h	USB0	CFIFO Port Select Register	CFIFOSEL	16	16	3, 4 PCLKB	2 ICLK	USBb	1740
000A 0022h	USB0	CFIFO Port Control Register	CFIFOCTR	16	16	3, 4 PCLKB	2 ICLK	USBb	1744
000A 0028h	USB0	D0FIFO Port Select Register	D0FIFOSEL	16	16	3, 4 PCLKB	2 ICLK	USBb	1742
000A 002Ah	USB0	D0FIFO Port Control Register	D0FIFOCTR	16	16	3, 4 PCLKB	2 ICLK	USBb	1744
000A 002Ch	USB0	D1FIFO Port Select Register	D1FIFOSEL	16	16	3, 4 PCLKB	2 ICLK	USBb	1742
000A 002Eh	USB0	D1FIFO Port Control Register	D1FIFOCTR	16	16	3, 4 PCLKB	2 ICLK	USBb	1744
000A 0030h	USB0	Interrupt Enable Register 0	INTENB0	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*5	USBb	1746
000A 0032h	USB0	Interrupt Enable Register 1	INTENB1	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*5	USBb	1747
000A 0036h	USB0	BRDY Interrupt Enable Register	BRDYENB	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*5	USBb	1748
000A 0038h	USB0	NRDY Interrupt Enable Register	NRDYENB	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*5	USBb	1749
000A 003Ah	USB0	BEMP Interrupt Enable Register	BEMPENB	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*5	USBb	1750
000A 003Ch	USB0	SOF Output Configuration Register	SOFCFG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*5	USBb	1751
000A 0040h	USB0	Interrupt Status Register 0	INTSTS0	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*5	USBb	1752
000A 0042h	USB0	Interrupt Status Register 1	INTSTS1	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*5	USBb	1755

Table 5.1 List of I/O Registers (Address Order) (43 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000A 0046h	USB0	BRDY Interrupt Status Register	BRDYSTS	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1758
000A 0048h	USB0	NRDY Interrupt Status Register	NRDYSTS	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1759
000A 004Ah	USB0	BEMP Interrupt Status Register	BEMPSTS	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1760
000A 004Ch	USB0	Frame Number Register	FRMNUM	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1761
000A 004Eh	USB0	Device State Change Register	DVCHGR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1762
000A 0050h	USB0	USB Address Register	USBADDR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1763
000A 0054h	USB0	USB Request Type Register	USBREQ	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1764
000A 0056h	USB0	USB Request Value Register	USBVAL	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1765
000A 0058h	USB0	USB Request Index Register	USBINDX	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1766
000A 005Ah	USB0	USB Request Length Register	USBLENG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1767
000A 005Ch	USB0	DCP Configuration Register	DCPCFG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1768
000A 005Eh	USB0	DCP Maximum Packet Size Register	DCPMAXP	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1769
000A 0060h	USB0	DCP Control Register	DCPCTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1770
000A 0064h	USB0	Pipe Window Select Register	PIPESEL	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1773
000A 0068h	USB0	Pipe Configuration Register	PIPECFG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1774
000A 006Ch	USB0	Pipe Maximum Packet Size Register	PIPEMAXP	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1776
000A 006Eh	USB0	Pipe Cycle Control Register	PIPEPERI	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1777

Table 5.1 List of I/O Registers (Address Order) (44 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000A 0070h	USB0	PIPE1 Control Register	PIPE1CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1778
000A 0072h	USB0	PIPE2 Control Register	PIPE2CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1778
000A 0074h	USB0	PIPE3 Control Register	PIPE3CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1778
000A 0076h	USB0	PIPE4 Control Register	PIPE4CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1778
000A 0078h	USB0	PIPE5 Control Register	PIPE5CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1778
000A 007Ah	USB0	PIPE6 Control Register	PIPE6CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1783
000A 007Ch	USB0	PIPE7 Control Register	PIPE7CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1783
000A 007Eh	USB0	PIPE8 Control Register	PIPE8CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1783
000A 0080h	USB0	PIPE9 Control Register	PIPE9CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1783
000A 0090h	USB0	Pipe1 Transaction Counter Enable Register	PIPE1TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1786
000A 0092h	USB0	Pipe1 Transaction Counter Register	PIPE1TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1787
000A 0094h	USB0	Pipe2 Transaction Counter Enable Register	PIPE2TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1786
000A 0096h	USB0	Pipe2 Transaction Counter Register	PIPE2TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1787
000A 0098h	USB0	Pipe3 Transaction Counter Enable Register	PIPE3TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1786
000A 009Ah	USB0	Pipe3 Transaction Counter Register	PIPE3TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1787
000A 009Ch	USB0	Pipe4 Transaction Counter Enable Register	PIPE4TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1786
000A 009Eh	USB0	Pipe4 Transaction Counter Register	PIPE4TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1787

Table 5.1 List of I/O Registers (Address Order) (45 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000A 00A0h	USB0	Pipe5 Transaction Counter Enable Register	PIPE5TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1786
000A 00A2h	USB0	Pipe5 Transaction Counter Register	PIPE5TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1787
000A 00D0h	USB0	Device Address 0 Configuration Register	DEVADD0	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1788
000A 00D2h	USB0	Device Address 1 Configuration Register	DEVADD1	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1788
000A 00D4h	USB0	Device Address 2 Configuration Register	DEVADD2	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1788
000A 00D6h	USB0	Device Address 3 Configuration Register	DEVADD3	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1788
000A 00D8h	USB0	Device Address 4 Configuration Register	DEVADD4	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1788
000A 00DAh	USB0	Device Address 5 Configuration Register	DEVADD5	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1788
000A 00F0h	USB0	PHY Cross Point Adjustment Register	PHYSLEW	32	32	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1788
000A 0400h	USB	Deep Standby USB Transceiver Control/Pin Monitoring Register	DPUSR0R	32	32	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1790
000A 0404h	USB	Deep Standby USB Suspend/Resume Interrupt Register	DPUSR1R	32	32	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/PCLKB)*5	USBb	1791
000A 0500h	PDC	PDC Control Register 0	PCCR0	32	32	2, 3 PCLKB	2 ICLK	PDC	2563
000A 0504h	PDC	PDC Control Register 1	PCCR1	32	32	2, 3 PCLKB	2 ICLK	PDC	2565
000A 0508h	PDC	PDC Status Register	PCSR	32	32	2, 3 PCLKB	2 ICLK	PDC	2566
000A 050Ch	PDC	PDC Pin Monitor Register	PCMONR	32	32	2, 3 PCLKB	2 ICLK	PDC	2568
000A 0510h	PDC	PDC Receive Data Register	PCDR	32	32	2, 3 PCLKB	2 ICLK	PDC	2568
000A 0514h	PDC	Vertical Capture Register	VCR	32	32	2, 3 PCLKB	2 ICLK	PDC	2570
000A 0518h	PDC	Horizontal Capture Register	HCR	32	32	2, 3 PCLKB	2 ICLK	PDC	2571
000C 0000h	EDMAC 0	EDMAC Mode Register	EDMR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMAC a	1689
000C 0008h	EDMAC 0	EDMAC Transmit Request Register	EDTRR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMAC a	1690
000C 0010h	EDMAC 0	EDMAC Receive Request Register	EDRRR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMAC a	1691
000C 0018h	EDMAC 0	Transmit Descriptor List Start Address Register	TDLAR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMAC a	1692
000C 0020h	EDMAC 0	Receive Descriptor List Start Address Register	RDLAR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMAC a	1693
000C 0028h	EDMAC 0	ETHERC/EDMAC Status Register	EESR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMAC a	1694
000C 0030h	EDMAC 0	ETHERC/EDMAC Status Interrupt Enable Register	EESIPR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMAC a	1701

Table 5.1 List of I/O Registers (Address Order) (46 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000C 0038h	EDMAC0	ETHERC/EDMAC Transmit/Receive Status Copy Enable Register	TRSCER	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1701
000C 0040h	EDMAC0	Missed-Frame Counter Register	RMFCR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1706
000C 0048h	EDMAC0	Transmit FIFO Threshold Register	TFTR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1707
000C 0050h	EDMAC0	FIFO Depth Register	FDR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1708
000C 0058h	EDMAC0	Receive Method Control Register	RMCR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1709
000C 0064h	EDMAC0	Transmit FIFO Underflow Counter	TFUCR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1710
000C 0068h	EDMAC0	Receive FIFO Overflow Counter	RFOCR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1710
000C 006Ch	EDMAC0	Independent Output Signal Setting Register	IOSR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1711
000C 0070h	EDMAC0	Flow Control Start FIFO Threshold Setting Register	FCFTR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1712
000C 0078h	EDMAC0	Receive Data Padding Insert Register	RPADIR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1713
000C 007Ch	EDMAC0	Transmit Interrupt Setting Register	TRIMD	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1714
000C 00C8h	EDMAC0	Receive Buffer Write Address Register	RBWAR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1714
000C 00CCh	EDMAC0	Receive Descriptor Fetch Address Register	RDFAR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1715
000C 00D4h	EDMAC0	Transmit Buffer Read Address Register	TBRAR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1715
000C 00D8h	EDMAC0	Transmit Descriptor Fetch Address Register	TDFAR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1716
000C 0100h	ETHERC0	ETHERC Mode Register	ECMR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1522
000C 0108h	ETHERC0	Receive Frame Maximum Length Register	RFLR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1524
000C 0110h	ETHERC0	ETHERC Status Register	ECSR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1525
000C 0118h	ETHERC0	ETHERC Interrupt Enable Register	ECSIPR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1526
000C 0120h	ETHERC0	PHY Interface Register	PIR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1527
000C 0128h	ETHERC0	PHY Status Register	PSR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1528
000C 0140h	ETHERC0	Random Number Generation Counter Limit Setting Register	RDMLR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1528
000C 0150h	ETHERC0	Interpacket Gap Register	IPGR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1529
000C 0154h	ETHERC0	Automatic PAUSE Frame Register	APR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1529
000C 0158h	ETHERC0	Manual PAUSE Frame Register	MPR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1530
000C 0160h	ETHERC0	Received PAUSE Frame Counter	RFCF	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1530
000C 0164h	ETHERC0	PAUSE Frame Retransmit Count Setting Register	TPAUSER	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1531
000C 0168h	ETHERC0	PAUSE Frame Retransmit Counter	TPAUSECR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1531
000C 016Ch	ETHERC0	Broadcast Frame Receive Count Setting Register	BCFRR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1532
000C 01C0h	ETHERC0	MAC Address Upper Bit Register	MAHR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1532
000C 01C8h	ETHERC0	MAC Address Lower Bit Register	MALR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1533

Table 5.1 List of I/O Registers (Address Order) (47 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000C 01D0h	ETHERC0	Transmit Retry Over Counter Register	TROCR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1533
000C 01D4h	ETHERC0	Late Collision Detect Counter Register	CDCR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1534
000C 01D8h	ETHERC0	Lost Carrier Counter Register	LCCR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1534
000C 01DCh	ETHERC0	Carrier Not Detect Counter Register	CNDCR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1535
000C 01E4h	ETHERC0	CRC Error Frame Receive Counter Register	CEFCR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1535
000C 01E8h	ETHERC0	Frame Receive Error Counter Register	FRECR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1536
000C 01ECh	ETHERC0	Too-Short Frame Receive Counter Register	TSFRCR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1536
000C 01F0h	ETHERC0	Too-Long Frame Receive Counter Register	TLFRCR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1537
000C 01F4h	ETHERC0	Received Alignment Error Frame Counter Register	RFCR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1537
000C 01F8h	ETHERC0	Multicast Address Frame Receive Counter Register	MAFCR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1538
000C 0200h	EDMAC1	EDMAC Mode Register	EDMR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1689
000C 0208h	EDMAC1	EDMAC Transmit Request Register	EDTRR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1690
000C 0210h	EDMAC1	EDMAC Receive Request Register	EDRRR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1691
000C 0218h	EDMAC1	Transmit Descriptor List Start Address Register	TDLAR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1692
000C 0220h	EDMAC1	Receive Descriptor List Start Address Register	RDLAR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1693
000C 0228h	EDMAC1	ETHERC/EDMAC Status Register	EESR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1694
000C 0230h	EDMAC1	ETHERC/EDMAC Status Interrupt Enable Register	EESIPR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1701
000C 0238h	EDMAC1	ETHERC/EDMAC Transmit/Receive Status Copy Enable Register	TRSCER	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1705
000C 0240h	EDMAC1	Missed-Frame Counter Register	RMFCR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1706
000C 0248h	EDMAC1	Transmit FIFO Threshold Register	TFTR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1707
000C 0250h	EDMAC1	FIFO Depth Register	FDR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1708
000C 0258h	EDMAC1	Receive Method Control Register	RMCR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1709
000C 0264h	EDMAC1	Transmit FIFO Underflow Counter	TFUCR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1710
000C 0268h	EDMAC1	Receive FIFO Overflow Counter	RFOCR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1710
000C 026Ch	EDMAC1	Independent Output Signal Setting Register	IOSR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1711
000C 0270h	EDMAC1	Flow Control Start FIFO Threshold Setting Register	FCFTR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1712
000C 0278h	EDMAC1	Receive Data Padding Insert Register	RPADIR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1713
000C 027Ch	EDMAC1	Transmit Interrupt Setting Register	TRIMD	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1714
000C 02C8h	EDMAC1	Receive Buffer Write Address Register	RBWAR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1714
000C 02CCh	EDMAC1	Receive Descriptor Fetch Address Register	RDFAR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1715
000C 02D4h	EDMAC1	Transmit Buffer Read Address Register	TBRAR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1715

Table 5.1 List of I/O Registers (Address Order) (48 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000C 02D8h	EDMAC1	Transmit Descriptor Fetch Address Register	TDFAR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1716
000C 0300h	ETHERC1	ETHERC Mode Register	ECMR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1522
000C 0308h	ETHERC1	Receive Frame Length Register	RFLR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1524
000C 0310h	ETHERC1	ETHERC Status Register	ECSR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1525
000C 0318h	ETHERC1	ETHERC Interrupt Enable Register	ECSIPR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1526
000C 0320h	ETHERC1	PHY Interface Register	PIR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1527
000C 0328h	ETHERC1	PHY Status Register	PSR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1528
000C 0340h	ETHERC1	Random Number Generation Counter Upper Limit Setting Register	RDMLR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1528
000C 0350h	ETHERC1	Interpacket Gap Register	IPGR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1529
000C 0354h	ETHERC1	Automatic PAUSE Frame Register	APR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1529
000C 0358h	ETHERC1	Manual PAUSE Frame Register	MPR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1530
000C 0360h	ETHERC1	Received PAUSE Frame Counter	RFCF	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1530
000C 0364h	ETHERC1	PAUSE Frame Retransmit Count Setting Register	TPAUSER	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1531
000C 0368h	ETHERC1	PAUSE Frame Retransmit Counter Register	TPAUSECR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1531
000C 036Ch	ETHERC1	Broadcast Frame Receive Count Setting Register	BCFRR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1532
000C 03C0h	ETHERC1	MAC Address Upper Bit Register	MAHR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1532
000C 03C8h	ETHERC1	MAC Address Lower Bit Register	MALR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1533
000C 03D0h	ETHERC1	Transmit Retry Over Counter Register	TROCR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1533
000C 03D4h	ETHERC1	Late Collision Detect Counter Register	CDCR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1534
000C 03D8h	ETHERC1	Lost Carrier Counter Register	LCCR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1534
000C 03DCh	ETHERC1	Carrier Not Detect Counter Register	CNDCR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1535
000C 03E4h	ETHERC1	CRC Error Frame Receive Counter Register	CEFCR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1535
000C 03E8h	ETHERC1	Frame Receive Error Counter Register	FRECR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1536
000C 03ECh	ETHERC1	Too-Short Frame Receive Counter Register	TSFRCR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1536
000C 03F0h	ETHERC1	Too-Long Frame Receive Counter Register	TLFRCR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1537
000C 03F4h	ETHERC1	Received Alignment Error Frame Counter Register	RFCR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1537
000C 03F8h	ETHERC1	Multicast Address Frame Receive Counter Register	MAFCR	32	32	13, 14 PCLKA	2 to 7 ICLK	ETHERC	1538
000C 0400h	PTPEDMAC	EDMAC Mode Register	EDMR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1689
000C 0408h	PTPEDMAC	EDMAC Transmit Request Register	EDTRR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1690
000C 0410h	PTPEDMAC	EDMAC Receive Request Register	EDRRR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1691
000C 0418h	PTPEDMAC	Transmit Descriptor List Start Address Register	TDLAR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1692

Table 5.1 List of I/O Registers (Address Order) (49 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000C 0420h	PTPEDMAC	Receive Descriptor List Start Address Register	RDLAR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1693
000C 0428h	PTPEDMAC	PTP/EDMAC Status Register	EESR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1698
000C 0430h	PTPEDMAC	PTP/EDMAC Status Interrupt Enable Register	EESIPR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1701
000C 0440h	PTPEDMAC	Missed-Frame Counter Register	RMFCR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1706
000C 0448h	PTPEDMAC	Transmit FIFO Threshold Register	TFTR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1707
000C 0450h	PTPEDMAC	FIFO Depth Register	FDR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1708
000C 0458h	PTPEDMAC	Receive Method Control Register	RMCR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1709
000C 0464h	PTPEDMAC	Transmit FIFO Underflow Counter	TFUCR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1710
000C 0468h	PTPEDMAC	Receive FIFO Overflow Counter	RFOCR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1710
000C 0470h	PTPEDMAC	Flow Control Start FIFO Threshold Setting Register	FCFTR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1712
000C 0478h	PTPEDMAC	Receive Data Padding Insert Register	RPADIR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1713
000C 047Ch	PTPEDMAC	Transmit Interrupt Setting Register	TRIMD	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1714
000C 04C8h	PTPEDMAC	Receive Buffer Write Address Register	RBWAR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1714
000C 04CCh	PTPEDMAC	Receive Descriptor Fetch Address Register	RDFAR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1715
000C 04D4h	PTPEDMAC	Transmit Buffer Read Address Register	TBRAR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1715
000C 04D8h	PTPEDMAC	Transmit Descriptor Fetch Address Register	TDFAR	32	32	4, 5 PCLKA	2, 3 ICLK	EDMACa	1716
000C 0500h	EPTPC	PTP Reset Register	PTRSTR	32	32	3, 4 PCLKA	2, 3 ICLK	EPTPC	1643
000C 0504h	EPTPC	STCA Clock Select Register	STCSELR	32	32	3, 4 PCLKA	2, 3 ICLK	EPTPC	1643
000C 1200h	MTU3	Timer Control Register	TCR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	854
000C 1201h	MTU4	Timer Control Register	TCR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	854
000C 1202h	MTU3	Timer Mode Register 1	TMDR1	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	860
000C 1203h	MTU4	Timer Mode Register 1	TMDR1	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	860
000C 1204h	MTU3	Timer I/O Control Register H	TIORH	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	864
000C 1205h	MTU3	Timer I/O Control Register L	TIORL	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	865
000C 1206h	MTU4	Timer I/O Control Register H	TIORH	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	864
000C 1207h	MTU4	Timer I/O Control Register L	TIORL	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	865
000C 1208h	MTU3	Timer Interrupt Enable Register	TIER	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	882
000C 1209h	MTU4	Timer Interrupt Enable Register	TIER	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	882
000C 120Ah	MTU	Timer Output Master Enable Register A	TOERA	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	900
000C 120Dh	MTU	Timer Gate Control Register A	TGCRA	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	908
000C 120Eh	MTU	Timer Output Control Register 1A	TOCR1A	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	902
000C 120Fh	MTU	Timer Output Control Register 2A	TOCR2A	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	904
000C 1210h	MTU3	Timer Counter	TCNT	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	889
000C 1212h	MTU4	Timer Counter	TCNT	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	889
000C 1214h	MTU	Timer Cycle Data Register A	TCDRA	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	909
000C 1216h	MTU	Timer Dead Time Data Register A	TDDRA	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	910
000C 1218h	MTU3	Timer General Register A	TGRA	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 121Ah	MTU3	Timer General Register B	TGRB	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 121Ch	MTU4	Timer General Register A	TGRA	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 121Eh	MTU4	Timer General Register B	TGRB	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1220h	MTU	Timer Subcounter A	TCNTSA	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	909

Table 5.1 List of I/O Registers (Address Order) (50 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000C 1222h	MTU	Timer Cycle Buffer Register A	TCBRA	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	910
000C 1224h	MTU3	Timer General Register C	TGRC	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1226h	MTU3	Timer General Register D	TGRD	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1228h	MTU4	Timer General Register C	TGRC	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 122Ah	MTU4	Timer General Register D	TGRD	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 122Ch	MTU3	Timer Status Register	TSR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	885
000C 122Dh	MTU4	Timer Status Register	TSR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	885
000C 1230h	MTU	Timer Interrupt Skipping Set Register 1A	TITCR1A	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	924
000C 1231h	MTU	Timer Interrupt Skipping Counter 1A	TITCNT1A	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	926
000C 1232h	MTU	Timer Buffer Transfer Set Register A	TBTERA	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	912
000C 1234h	MTU	Timer Dead Time Enable Register A	TDERA	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	911
000C 1236h	MTU	Timer Output Level Buffer Register A	TOLBRA	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	907
000C 1238h	MTU3	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	886
000C 1239h	MTU4	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	886
000C 123Ah	MTU	Timer Interrupt Skipping Mode Register A	TITMRA	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	923
000C 123Bh	MTU	Timer Interrupt Skipping Set Register 2A	TITCR2A	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	924
000C 123Ch	MTU	Timer Interrupt Skipping Counter 2A	TITCNT2A	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	926
000C 1240h	MTU4	Timer A/D Converter Start Request Control Register	TADCR	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	918
000C 1244h	MTU4	Timer A/D Converter Start Request Cycle Set Register A	TADCORA	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	922
000C 1246h	MTU4	Timer A/D Converter Start Request Cycle Set Register B	TADCORB	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	922
000C 1248h	MTU4	Timer A/D Converter Start Request Cycle Set Buffer Register A	TADCOBRA	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	922
000C 124Ah	MTU4	Timer A/D Converter Start Request Cycle Set Buffer Register B	TADCOBRB	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	922
000C 124Ch	MTU3	Timer Control Register 2	TCR2	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	856
000C 124Dh	MTU4	Timer Control Register 2	TCR2	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	856
000C 1260h	MTU	Timer Waveform Control Register A	TWCRA	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	913
000C 1270h	MTU	Timer Mode Register 2A	TMDR2A	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	862
000C 1272h	MTU3	Timer General Register E	TGRE	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1274h	MTU4	Timer General Register E	TGRE	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1276h	MTU4	Timer General Register F	TGRF	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1280h	MTU	Timer Start Register A	TSTRA	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	893
000C 1281h	MTU	Timer Synchronous Register A	TSYRA	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	895
000C 1282h	MTU	Timer Counter Synchronous Start Register	TCSYSTR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	897
000C 1284h	MTU	Timer Read/Write Enable Register A	TRWERA	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	899
000C 1290h	MTU0	Noise Filter Control Register 0	NFCR0	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	915
000C 1291h	MTU1	Noise Filter Control Register 1	NFCR1	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	915
000C 1292h	MTU2	Noise Filter Control Register 2	NFCR2	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	915
000C 1293h	MTU3	Noise Filter Control Register 3	NFCR3	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	915
000C 1294h	MTU4	Noise Filter Control Register 4	NFCR4	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	915
000C 1298h	MTU8	Noise Filter Control Register 8	NFCR8	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	915
000C 1299h	MTU0	Noise Filter Control Register C	NFCRC	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	916
000C 1300h	MTU0	Timer Control Register	TCR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	854
000C 1301h	MTU0	Timer Mode Register 1	TMDR1	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	860
000C 1302h	MTU0	Timer I/O Control Register H	TIORH	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	864
000C 1303h	MTU0	Timer I/O Control Register L	TIORL	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	864
000C 1304h	MTU0	Timer Interrupt Enable Register	TIER	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	882
000C 1306h	MTU0	Timer Counter	TCNT	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	889
000C 1308h	MTU0	Timer General Register A	TGRA	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891

Table 5.1 List of I/O Registers (Address Order) (51 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000C 130Ah	MTU0	Timer General Register B	TGRB	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 130Ch	MTU0	Timer General Register C	TGRC	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 130Eh	MTU0	Timer General Register D	TGRD	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1320h	MTU0	Timer General Register E	TGRE	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1322h	MTU0	Timer General Register F	TGRF	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1324h	MTU0	Timer Interrupt Enable Register 2	TIER2	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	882
000C 1326h	MTU0	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	886
000C 1328h	MTU0	Timer Control Register 2	TCR2	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	856
000C 1380h	MTU1	Timer Control Register	TCR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	854
000C 1381h	MTU1	Timer Mode Register 1	TMDR1	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	860
000C 1382h	MTU1	Timer I/O Control Register	TIOR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	864
000C 1384h	MTU1	Timer Interrupt Enable Register	TIER	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	882
000C 1385h	MTU1	Timer Status Register	TSR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	885
000C 1386h	MTU1	Timer Counter	TCNT	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	889
000C 1388h	MTU1	Timer General Register A	TGRA	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 138Ah	MTU1	Timer General Register B	TGRB	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1390h	MTU1	Timer Input Capture Control Register	TICCR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	887
000C 1391h	MTU1	Timer Mode Register 3	TMDR3	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	863
000C 1394h	MTU1	Timer Control Register 2	TCR2	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	856
000C 13A0h	MTU1	Timer Longword Counter	TCNTLW	32	32	5, 6 PCLKA	2, 3 ICLK	MTU3a	890
000C 13A4h	MTU1	Timer Longword General Register	TGRALW	32	32	5, 6 PCLKA	2, 3 ICLK	MTU3a	892
000C 13A8h	MTU1	Timer Longword General Register	TGRBLW	32	32	5, 6 PCLKA	2, 3 ICLK	MTU3a	892
000C 1400h	MTU2	Timer Control Register	TCR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	854
000C 1401h	MTU2	Timer Mode Register 1	TMDR1	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	860
000C 1402h	MTU2	Timer I/O Control Register	TIOR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	864
000C 1404h	MTU2	Timer Interrupt Enable Register	TIER	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	882
000C 1405h	MTU2	Timer Status Register	TSR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	885
000C 1406h	MTU2	Timer Counter	TCNT	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	889
000C 1408h	MTU2	Timer General Register A	TGRA	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 140Ah	MTU2	Timer General Register B	TGRB	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 140Ch	MTU2	Timer Control Register 2	TCR2	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	856
000C 1600h	MTU8	Timer Control Register	TCR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	854
000C 1601h	MTU8	Timer Mode Register 1	TMDR1	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	860
000C 1602h	MTU8	Timer I/O Control Register H	TIORH	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	864
000C 1603h	MTU8	Timer I/O Control Register L	TIORL	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	864
000C 1604h	MTU8	Timer Interrupt Enable Register	TIER	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	882
000C 1606h	MTU8	Timer Control Register 2	TCR2	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	856
000C 1608h	MTU8	Timer Counter	TCNT	32	32	5, 6 PCLKA	2, 3 ICLK	MTU3a	889
000C 160Ch	MTU8	Timer General Register A	TGRA	32	32	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1610h	MTU8	Timer General Register B	TGRB	32	32	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1614h	MTU8	Timer General Register C	TGRC	32	32	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1618h	MTU8	Timer General Register D	TGRD	32	32	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1A00h	MTU6	Timer Control Register	TCR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	854
000C 1A01h	MTU7	Timer Control Register	TCR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	854
000C 1A02h	MTU6	Timer Mode Register 1	TMDR1	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	860
000C 1A03h	MTU7	Timer Mode Register 1	TMDR1	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	860
000C 1A04h	MTU6	Timer I/O Control Register H	TIORH	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	864
000C 1A05h	MTU6	Timer I/O Control Register L	TIORL	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	864
000C 1A06h	MTU7	Timer I/O Control Register H	TIORH	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	864
000C 1A07h	MTU7	Timer I/O Control Register L	TIORL	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	864

Table 5.1 List of I/O Registers (Address Order) (52 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000C 1A08h	MTU6	Timer Interrupt Enable Register	TIER	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	882
000C 1A09h	MTU7	Timer Interrupt Enable Register	TIER	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	882
000C 1A0Ah	MTU	Timer Output Master Enable Register B	TOERB	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	900
000C 1A0Eh	MTU	Timer Output Control Register 1B	TOCR1B	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	902
000C 1A0Fh	MTU	Timer Output Control Register 2B	TOCR2B	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	904
000C 1A10h	MTU6	Timer Counter	TCNT	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	889
000C 1A12h	MTU7	Timer Counter	TCNT	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	889
000C 1A14h	MTU	Timer Cycle Data Register B	TCDRB	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	909
000C 1A16h	MTU	Timer Dead Time Data Register B	TDDRB	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	910
000C 1A18h	MTU6	Timer General Register A	TGRA	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1A1Ah	MTU6	Timer General Register B	TGRB	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1A1Ch	MTU7	Timer General Register A	TGRA	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1A1Eh	MTU7	Timer General Register B	TGRB	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1A20h	MTU	Timer Subcounter B	TCNTSB	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	909
000C 1A22h	MTU	Timer Cycle Buffer Register B	TCBRB	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	910
000C 1A24h	MTU6	Timer General Register C	TGRC	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1A26h	MTU6	Timer General Register D	TGRD	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1A28h	MTU7	Timer General Register C	TGRC	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1A2Ah	MTU7	Timer General Register D	TGRD	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1A2Ch	MTU6	Timer Status Register	TSR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	885
000C 1A2Dh	MTU7	Timer Status Register	TSR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	885
000C 1A30h	MTU	Timer Interrupt Skipping Set Register 1B	TITCR1B	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	924
000C 1A31h	MTU	Timer Interrupt Skipping Counter 1B	TITCNT1B	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	926
000C 1A32h	MTU	Timer Buffer Transfer Set Register B	TBTERB	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	912
000C 1A34h	MTU	Timer Dead Time Enable Register B	TDERB	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	911
000C 1A36h	MTU	Timer Output Level Buffer Register B	TOLBRB	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	907
000C 1A38h	MTU6	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	886
000C 1A39h	MTU7	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	886
000C 1A3Ah	MTU	Timer Interrupt Skipping Mode Register B	TITMRB	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	923
000C 1A3Bh	MTU	Timer Interrupt Skipping Set Register 2B	TITCR2B	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	924
000C 1A3Ch	MTU	Timer Interrupt Skipping Counter 2B	TITCNT2B	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	926
000C 1A40h	MTU7	Timer A/D Converter Start Request Control Register	TADCR	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	918
000C 1A44h	MTU7	Timer A/D Converter Start Request Cycle Set Register A	TADCORA	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	922
000C 1A46h	MTU7	Timer A/D Converter Start Request Cycle Set Register B	TADCORB	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	922
000C 1A48h	MTU7	Timer A/D Converter Start Request Cycle Set Buffer Register A	TADCOBRA	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	922
000C 1A4Ah	MTU7	Timer A/D Converter Start Request Cycle Set Buffer Register B	TADCOBRB	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	922
000C 1A4Ch	MTU6	Timer Control Register 2	TCR2	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	856
000C 1A4Dh	MTU7	Timer Control Register 2	TCR2	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	856
000C 1A50h	MTU6	Timer Synchronous Clear Register	TSYCR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	888
000C 1A60h	MTU	Timer Waveform Control Register B	TWCRB	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	913
000C 1A70h	MTU	Timer Mode Register 2B	TMDR2B	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	862
000C 1A72h	MTU6	Timer General Register E	TGRE	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1A74h	MTU7	Timer General Register E	TGRE	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1A76h	MTU7	Timer General Register F	TGRF	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1A80h	MTU	Timer Start Register B	TSTRB	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	893
000C 1A81h	MTU	Timer Synchronous Register B	TSYRB	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	895
000C 1A84h	MTU	Timer Read/Write Enable Register B	TRWERB	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	899

Table 5.1 List of I/O Registers (Address Order) (53 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000C 1A93h	MTU6	Noise Filter Control Register 6	NFCR6	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	915
000C 1A94h	MTU7	Noise Filter Control Register 7	NFCR7	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	915
000C 1A95h	MTU5	Noise Filter Control Register 5	NFCR5	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	917
000C 1C80h	MTU5	Timer Counter U	TCNTU	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	889
000C 1C82h	MTU5	Timer General Register U	TGRU	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1C84h	MTU5	Timer Control Register U	TCRU	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	854
000C 1C85h	MTU5	Timer Control Register 2	TCR2U	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	857
000C 1C86h	MTU5	Timer I/O Control Register U	TIORU	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	864
000C 1C90h	MTU5	Timer Counter V	TCNTV	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	889
000C 1C92h	MTU5	Timer General Register V	TGRV	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1C94h	MTU5	Timer Control Register V	TCRV	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	854
000C 1C95h	MTU5	Timer Control Register 2	TCR2V	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	857
000C 1C96h	MTU5	Timer I/O Control Register V	TIORV	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	864
000C 1CA0h	MTU5	Timer Counter W	TCNTW	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	889
000C 1CA2h	MTU5	Timer General Register W	TGRW	16	16	5, 6 PCLKA	2, 3 ICLK	MTU3a	891
000C 1CA4h	MTU5	Timer Control Register W	TCRW	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	854
000C 1CA5h	MTU5	Timer Control Register 2	TCR2W	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	857
000C 1CA6h	MTU5	Timer I/O Control Register W	TIORW	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	864
000C 1CB2h	MTU5	Timer Interrupt Enable Register	TIER	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	882
000C 1CB4h	MTU5	Timer Start Register	TSTR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	893
000C 1CB6h	MTU5	Timer Compare Match Clear Register	TCNTCMPCLR	8	8	5, 6 PCLKA	2, 3 ICLK	MTU3a	881
000C 2000h	GPT	General PWM Timer Software Start Register	GTSTR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1138
000C 2002h	GPT	Noise Filter Control Register	NFCR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1139
000C 2004h	GPT	General PWM Timer Hardware Source Start/Stop Control Register	GTHSCR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1141
000C 2006h	GPT	General PWM Timer Hardware Source Clear Control Register	GTHCCR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1143
000C 2008h	GPT	General PWM Timer Hardware Start Source Select Register	GTHSSR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1145
000C 200Ah	GPT	General PWM Timer Hardware Stop/Clear Source Select Register	GTHPSR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1146
000C 200Ch	GPT	General PWM Timer Write-Protection Register	GTWP	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1147
000C 200Eh	GPT	General PWM Timer Sync Register	GTSYNC	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1148
000C 2010h	GPT	General PWM Timer External Trigger Input Interrupt Register	GTETINT	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1149
000C 2014h	GPT	General PWM Timer Buffer Operation Disable Register	GTBDR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1150
000C 2018h	GPT	General PWM Timer Start Write-Protection Register	GTSWP	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1151
000C 2100h	GPT0	General PWM Timer I/O Control Register	GTIOR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1152
000C 2102h	GPT0	General PWM Timer Interrupt Output Setting Register	GTINTAD	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1156
000C 2104h	GPT0	General PWM Timer Control Register	GTCR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1158
000C 2106h	GPT0	General PWM Timer Buffer Enable Register	GTBER	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1160
000C 2108h	GPT0	General PWM Timer Count Direction Register	GTUDC	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1162
000C 210Ah	GPT0	General PWM Timer Interrupt and A/D Converter Start Request Skipping Setting Register	GTITC	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1163
000C 210Ch	GPT0	General PWM Timer Status Register	GTST	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1165
000C 210Eh	GPT0	General PWM Timer Counter	GTCNT	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2110h	GPT0	General PWM Timer Compare Capture Register A	GTCCRA	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2112h	GPT0	General PWM Timer Compare Capture Register B	GTCCRB	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2114h	GPT0	General PWM Timer Compare Capture Register C	GTCCRC	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2116h	GPT0	General PWM Timer Compare Capture Register D	GTCCRD	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2118h	GPT0	General PWM Timer Compare Capture Register E	GTCCRE	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166

Table 5.1 List of I/O Registers (Address Order) (54 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000C 211Ah	GPT0	General PWM Timer Compare Capture Register F	GTCCRF	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 211Ch	GPT0	General PWM Timer Cycle Setting Register	GTPR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1167
000C 211Eh	GPT0	General PWM Timer Cycle Setting Buffer Register	GTPBR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1167
000C 2120h	GPT0	General PWM Timer Cycle Setting Double-Buffer Register	GTPDBR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1167
000C 2124h	GPT0	A/D Converter Start Request Timing Register A	GTADTRA	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 2126h	GPT0	A/D Converter Start Request Timing Buffer Register A	GTADTBRA	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 2128h	GPT0	A/D Converter Start Request Timing Double-Buffer Register A	GTADTDBRA	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 212Ch	GPT0	A/D Converter Start Request Timing Register B	GTADTRB	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 212Eh	GPT0	A/D Converter Start Request Timing Buffer Register B	GTADTBRB	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 2130h	GPT0	A/D Converter Start Request Timing Double-Buffer Register B	GTADTDBRB	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 2134h	GPT0	General PWM Timer Output Negate Control Register	GTONCR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1169
000C 2136h	GPT0	General PWM Timer Dead Time Control Register	GTDTCR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1171
000C 2138h	GPT0	General PWM Timer Dead Time Value Register U	GTDVU	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1172
000C 213Ah	GPT0	General PWM Timer Dead Time Value Register D	GTDVD	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1172
000C 213Ch	GPT0	General PWM Timer Dead Time Buffer Register U	GTDBU	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1172
000C 213Eh	GPT0	General PWM Timer Dead Time Buffer Register D	GTDBD	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1172
000C 2140h	GPT0	General PWM Timer Output Protection Function Status Register	GTSOS	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1173
000C 2142h	GPT0	General PWM Timer Output Protection Function Temporary Release Register	GTSOTR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1174
000C 2180h	GPT1	General PWM Timer I/O Control Register	GTIOR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1152
000C 2182h	GPT1	General PWM Timer Interrupt Output Setting Register	GTINTAD	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1156
000C 2184h	GPT1	General PWM Timer Control Register	GTCR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1158
000C 2186h	GPT1	General PWM Timer Buffer Enable Register	GTBER	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1160
000C 2188h	GPT1	General PWM Timer Count Direction Register	GTUDC	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1162
000C 218Ah	GPT1	General PWM Timer Interrupt and A/D Converter Start Request Skipping Setting Register	GTITC	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1163
000C 218Ch	GPT1	General PWM Timer Status Register	GTST	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1165
000C 218Eh	GPT1	General PWM Timer Counter	GTCNT	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2190h	GPT1	General PWM Timer Compare Capture Register A	GTCCRA	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2192h	GPT1	General PWM Timer Compare Capture Register B	GTCCRB	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2194h	GPT1	General PWM Timer Compare Capture Register C	GTCCRC	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2196h	GPT1	General PWM Timer Compare Capture Register D	GTCCRD	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2198h	GPT1	General PWM Timer Compare Capture Register E	GTCCRE	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 219Ah	GPT1	General PWM Timer Compare Capture Register F	GTCCRF	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 219Ch	GPT1	General PWM Timer Cycle Setting Register	GTPR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1167
000C 219Eh	GPT1	General PWM Timer Cycle Setting Buffer Register	GTPBR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1167
000C 21A0h	GPT1	General PWM Timer Cycle Setting Double-Buffer Register	GTPDBR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1167
000C 21A4h	GPT1	A/D Converter Start Request Timing Register A	GTADTRA	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 21A6h	GPT1	A/D Converter Start Request Timing Buffer Register A	GTADTBRA	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 21A8h	GPT1	A/D Converter Start Request Timing Double-Buffer Register A	GTADTDBRA	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 21ACh	GPT1	A/D Converter Start Request Timing Register B	GTADTRB	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 21AEh	GPT1	A/D Converter Start Request Timing Buffer Register B	GTADTBRB	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 21B0h	GPT1	A/D Converter Start Request Timing Double-Buffer Register B	GTADTDBRB	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 21B4h	GPT1	General PWM Timer Output Negate Control Register	GTONCR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1169
000C 21B6h	GPT1	General PWM Timer Dead Time Control Register	GTDTCR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1171
000C 21B8h	GPT1	General PWM Timer Dead Time Value Register U	GTDVU	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1172
000C 21BAh	GPT1	General PWM Timer Dead Time Value Register D	GTDVD	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1172

Table 5.1 List of I/O Registers (Address Order) (55 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000C 21BCh	GPT1	General PWM Timer Dead Time Buffer Register U	GTDBU	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1172
000C 21BEh	GPT1	General PWM Timer Dead Time Buffer Register D	GTDBD	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1172
000C 21C0h	GPT1	General PWM Timer Output Protection Function Status Register	GTSOS	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1173
000C 21C2h	GPT1	General PWM Timer Output Protection Function Temporary Release Register	GTSOTR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1174
000C 2200h	GPT2	General PWM Timer I/O Control Register	GTIOR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1152
000C 2202h	GPT2	General PWM Timer Interrupt Output Setting Register	GTINTAD	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1156
000C 2204h	GPT2	General PWM Timer Control Register	GTCR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1158
000C 2206h	GPT2	General PWM Timer Buffer Enable Register	GTBER	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1160
000C 2208h	GPT2	General PWM Timer Count Direction Register	GTUDC	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1162
000C 220Ah	GPT2	General PWM Timer Interrupt and A/D Converter Start Request Skipping Setting Register	GTITC	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1163
000C 220Ch	GPT2	General PWM Timer Status Register	GTST	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1165
000C 220Eh	GPT2	General PWM Timer Counter	GTCNT	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2210h	GPT2	General PWM Timer Compare Capture Register A	GTCCRA	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2212h	GPT2	General PWM Timer Compare Capture Register B	GTCCRB	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2214h	GPT2	General PWM Timer Compare Capture Register C	GTCCRC	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2216h	GPT2	General PWM Timer Compare Capture Register D	GTCCRD	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2218h	GPT2	General PWM Timer Compare Capture Register E	GTCCRE	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 221Ah	GPT2	General PWM Timer Compare Capture Register F	GTCCRF	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 221Ch	GPT2	General PWM Timer Cycle Setting Register	GTPR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1167
000C 221Eh	GPT2	General PWM Timer Cycle Setting Buffer Register	GTPBR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1167
000C 2220h	GPT2	General PWM Timer Cycle Setting Double-Buffer Register	GTPDBR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1167
000C 2224h	GPT2	A/D Converter Start Request Timing Register A	GTADTRA	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 2226h	GPT2	A/D Converter Start Request Timing Buffer Register A	GTADTBRA	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 2228h	GPT2	A/D Converter Start Request Timing Double-Buffer Register A	GTADTDBRA	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 222Ch	GPT2	A/D Converter Start Request Timing Register B	GTADTRB	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 222Eh	GPT2	A/D Converter Start Request Timing Buffer Register B	GTADTBRB	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 2230h	GPT2	A/D Converter Start Request Timing Double-Buffer Register B	GTADTDBRB	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 2234h	GPT2	General PWM Timer Output Negate Control Register	GTONCR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1169
000C 2236h	GPT2	General PWM Timer Dead Time Control Register	GTDTCR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1171
000C 2238h	GPT2	General PWM Timer Dead Time Value Register U	GTDVU	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1172
000C 223Ah	GPT2	General PWM Timer Dead Time Value Register D	GTDVD	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1172
000C 223Ch	GPT2	General PWM Timer Dead Time Buffer Register U	GTDBU	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1172
000C 223Eh	GPT2	General PWM Timer Dead Time Buffer Register D	GTDBD	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1172
000C 2240h	GPT2	General PWM Timer Output Protection Function Status Register	GTSOS	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1173
000C 2242h	GPT2	General PWM Timer Output Protection Function Temporary Release Register	GTSOTR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1174
000C 2280h	GPT3	General PWM Timer I/O Control Register	GTIOR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1152
000C 2282h	GPT3	General PWM Timer Interrupt Output Setting Register	GTINTAD	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1156
000C 2284h	GPT3	General PWM Timer Control Register	GTCR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1158
000C 2286h	GPT3	General PWM Timer Buffer Enable Register	GTBER	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1160
000C 2288h	GPT3	General PWM Timer Count Direction Register	GTUDC	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1162
000C 228Ah	GPT3	General PWM Timer Interrupt and A/D Converter Start Request Skipping Setting Register	GTITC	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1163
000C 228Ch	GPT3	General PWM Timer Status Register	GTST	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1165
000C 228Eh	GPT3	General PWM Timer Counter	GTCNT	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2290h	GPT3	General PWM Timer Compare Capture Register A	GTCCRA	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2292h	GPT3	General PWM Timer Compare Capture Register B	GTCCRB	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166

Table 5.1 List of I/O Registers (Address Order) (56 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000C 2294h	GPT3	General PWM Timer Compare Capture Register C	GTCCRC	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2296h	GPT3	General PWM Timer Compare Capture Register D	GTCCRD	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 2298h	GPT3	General PWM Timer Compare Capture Register E	GTCCRE	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 229Ah	GPT3	General PWM Timer Compare Capture Register F	GTCCRF	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1166
000C 229Ch	GPT3	General PWM Timer Cycle Setting Register	GTPR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1167
000C 229Eh	GPT3	General PWM Timer Cycle Setting Buffer Register	GTPBR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1167
000C 22A0h	GPT3	General PWM Timer Cycle Setting Double-Buffer Register	GTPDBR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1167
000C 22A4h	GPT3	A/D Converter Start Request Timing Register A	GTADTRA	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 22A6h	GPT3	A/D Converter Start Request Timing Buffer Register A	GTADTBRA	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 22A8h	GPT3	A/D Converter Start Request Timing Double-Buffer Register A	GTADTDBRA	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 22ACh	GPT3	A/D Converter Start Request Timing Register B	GTADTRB	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 22AEh	GPT3	A/D Converter Start Request Timing Buffer Register B	GTADTBRB	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 22B0h	GPT3	A/D Converter Start Request Timing Double-Buffer Register B	GTADTDBRB	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1168
000C 22B4h	GPT3	General PWM Timer Output Negate Control Register	GTONCR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1169
000C 22B6h	GPT3	General PWM Timer Dead Time Control Register	GTDTCR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1171
000C 22B8h	GPT3	General PWM Timer Dead Time Value Register U	GTDVU	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1172
000C 22BAh	GPT3	General PWM Timer Dead Time Value Register D	GTDVD	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1172
000C 22BCh	GPT3	General PWM Timer Dead Time Buffer Register U	GTDBU	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1172
000C 22BEh	GPT3	General PWM Timer Dead Time Buffer Register D	GTDBD	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1172
000C 22C0h	GPT3	General PWM Timer Output Protection Function Status Register	GTSOS	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1173
000C 22C2h	GPT3	General PWM Timer Output Protection Function Temporary Release Register	GTSOTR	16	16	4, 5 PCLKA	2, 3 ICLK	GPTA	1174
000C 4000h	EPTPC	MINT Interrupt Source Status Register	MIESR	32	32	5, 6 PCLKA	2, 3 ICLK	EPTPC	1555
000C 4004h	EPTPC	MINT Interrupt Request Permission Register	MIEIPR	32	32	5, 6 PCLKA	2, 3 ICLK	EPTPC	1557
000C 4010h	EPTPC	ELC Output/IPLS Interrupt Request Permission Register	ELIPPR	32	32	5, 6 PCLKA	2, 3 ICLK	EPTPC	1558
000C 4014h	EPTPC	ELC Output/IPLS Interrupt Permission Automatic Clearing Register	ELIPACR	32	32	5, 6 PCLKA	2, 3 ICLK	EPTPC	1560
000C 4040h	EPTPC	STCA Status Register	STSR	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1562
000C 4044h	EPTPC	STCA Status Notification Permission Register	STIPR	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1563
000C 4050h	EPTPC	STCA Clock Frequency Setting Register	STCFR	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1564
000C 4054h	EPTPC	STCA Operating Mode Register	STMR	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1565
000C 4058h	EPTPC	Sync Message Reception Timeout Register	SYNTOR	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1567
000C 4060h	EPTPC	IPLS Interrupt Request Timer Select Register	IPTSELR	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1568
000C 4064h	EPTPC	MINT Interrupt Request Timer Select Register	MITSELR	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1569
000C 4068h	EPTPC	ELC Output Timer Select Register	ELTSELR	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1570
000C 406Ch	EPTPC	Time Synchronization Channel Select Register	STCHSELR	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1571
000C 4080h	EPTPC	Slave Time Synchronization Start Register	SYNSTARTR	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1572
000C 4084h	EPTPC	Local Time Counter Initial Value Load Directive Register	LCIVLDR	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1572
000C 4090h	EPTPC	Synchronization Loss Detection Threshold Register	SYNTDARU	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1573
000C 4094h	EPTPC	Synchronization Loss Detection Threshold Register	SYNTDARL	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1573
000C 4098h	EPTPC	Synchronization Detection Threshold Register	SYNTDBRU	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1574
000C 409Ch	EPTPC	Synchronization Detection Threshold Register	SYNTDBRL	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1574
000C 40B0h	EPTPC	Local Time Counter Initial Value Register	LCIVRU	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1575
000C 40B4h	EPTPC	Local Time Counter Initial Value Register	LCIVRM	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1575
000C 40B8h	EPTPC	Local Time Counter Initial Value Register	LCIVRL	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1575
000C 4124h	EPTPC	Worst 10 Acquisition Directive Register	GETW10R	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1576
000C 4128h	EPTPC	Positive Gradient Limit Register	PLIMITRU	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1577

Table 5.1 List of I/O Registers (Address Order) (57 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000C 412Ch	EPTPC	Positive Gradient Limit Register	PLIMITRM	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1577
000C 4130h	EPTPC	Positive Gradient Limit Register	PLIMITRL	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1577
000C 4134h	EPTPC	Negative Gradient Limit Register	MLIMITRU	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1579
000C 4138h	EPTPC	Negative Gradient Limit Register	MLIMITRM	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1579
000C 413Ch	EPTPC	Negative Gradient Limit Register	MLIMITRL	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1579
000C 4140h	EPTPC	Statistical Information Retention Control Register	GETINFOR	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1580
000C 4170h	EPTPC	Local Time Counter	LCCVRU	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1581
000C 4174h	EPTPC	Local Time Counter	LCCVRM	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1581
000C 4178h	EPTPC	Local Time Counter	LCCVRL	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1581
000C 4210h	EPTPC	Positive Gradient Worst 10 Value Register	PW10VRU	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1583
000C 4214h	EPTPC	Positive Gradient Worst 10 Value Register	PW10VRM	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1583
000C 4218h	EPTPC	Positive Gradient Worst 10 Value Register	PW10VRL	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1583
000C 42D0h	EPTPC	Negative Gradient Worst 10 Value Register	MW10RU	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1585
000C 42D4h	EPTPC	Negative Gradient Worst 10 Value Register	MW10RM	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1585
000C 42D8h	EPTPC	Negative Gradient Worst 10 Value Register	MW10RL	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1585
000C 4300h	EPTPC	Timer Start Time Setting Register	TMSTTRU0	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1587
000C 4304h	EPTPC	Timer Start Time Setting Register	TMSTTRL0	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1587
000C 4308h	EPTPC	Timer Cycle Setting Register 0	TMCYCR0	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1588
000C 430Ch	EPTPC	Timer Pulse Width Setting Register 0	TMPLSR0	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1589
000C 4310h	EPTPC	Timer Start Time Setting Register	TMSTTRU1	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1587
000C 4314h	EPTPC	Timer Start Time Setting Register	TMSTTRL1	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1587
000C 4318h	EPTPC	Timer Cycle Setting Register 1	TMCYCR1	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1588
000C 431Ch	EPTPC	Timer Pulse Width Setting Register 1	TMPLSR1	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1589
000C 4320h	EPTPC	Timer Start Time Setting Register	TMSTTRU2	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1587
000C 4324h	EPTPC	Timer Start Time Setting Register	TMSTTRL2	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1587
000C 4328h	EPTPC	Timer Cycle Setting Register 2	TMCYCR2	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1588
000C 432Ch	EPTPC	Timer Pulse Width Setting Register 2	TMPLSR2	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1589
000C 4330h	EPTPC	Timer Start Time Setting Register	TMSTTRU3	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1587
000C 4334h	EPTPC	Timer Start Time Setting Register	TMSTTRL3	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1587
000C 4338h	EPTPC	Timer Cycle Setting Register 3	TMCYCR3	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1588
000C 433Ch	EPTPC	Timer Pulse Width Setting Register 3	TMPLSR3	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1589
000C 4340h	EPTPC	Timer Start Time Setting Register	TMSTTRU4	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1587
000C 4344h	EPTPC	Timer Start Time Setting Register	TMSTTRL4	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1587
000C 4348h	EPTPC	Timer Cycle Setting Register 4	TMCYCR4	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1588
000C 434Ch	EPTPC	Timer Pulse Width Setting Register 4	TMPLSR4	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1589
000C 4350h	EPTPC	Timer Start Time Setting Register	TMSTTRU5	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1587
000C 4354h	EPTPC	Timer Start Time Setting Register	TMSTTRL5	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1587
000C 4358h	EPTPC	Timer Cycle Setting Register 5	TMCYCR5	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1588
000C 435Ch	EPTPC	Timer Pulse Width Setting Register 5	TMPLSR5	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1589
000C 437Ch	EPTPC	Timer Start Register	TMSTARTR	32	32	8 to 43 PCLKA	2 to 22 ICLK	EPTPC	1590
000C 4400h	EPTPC	PRC-TC Status Register	PRSR	32	32	9, 10 PCLKA	2 to 5 ICLK	EPTPC	1591
000C 4404h	EPTPC	PRC-TC Status Notification Permission Register	PRIPR	32	32	9, 10 PCLKA	2 to 5 ICLK	EPTPC	1593
000C 4410h	EPTPC	Channel 0 Local MAC Address Register	PRMACRU0	32	32	9, 10 PCLKA	2 to 5 ICLK	EPTPC	1594
000C 4414h	EPTPC	Channel 0 Local MAC Address Register	PRMACRL0	32	32	9, 10 PCLKA	2 to 5 ICLK	EPTPC	1594
000C 4418h	EPTPC	Channel 1 Local MAC Address Register	PRMACRU1	32	32	9, 10 PCLKA	2 to 5 ICLK	EPTPC	1595
000C 441Ch	EPTPC	Channel 1 Local MAC Address Register	PRMACRL1	32	32	9, 10 PCLKA	2 to 5 ICLK	EPTPC	1595
000C 4420h	EPTPC	Packet Transmission Control Register	TRNDISR	32	32	9, 10 PCLKA	2 to 5 ICLK	EPTPC	1596
000C 4430h	EPTPC	Relay Mode Register	TRNMR	32	32	9, 10 PCLKA	2 to 5 ICLK	EPTPC	1597
000C 4434h	EPTPC	Cut-Through Transfer Start Threshold Register	TRNCTTDR	32	32	9, 10 PCLKA	2 to 5 ICLK	EPTPC	1598

Table 5.1 List of I/O Registers (Address Order) (58 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000C 4800h	EPTPC 0	SYNFP Status Register	SYSR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1599
000C 4804h	EPTPC 0	SYNFP Status Notification Permission Register	SYIPR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1601
000C 4810h	EPTPC 0	SYNFP MAC Address Register	SYMACRU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1602
000C 4814h	EPTPC 0	SYNFP MAC Address Register	SYMACRL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1602
000C 481Ch	EPTPC 0	SYNFP Local IP Address Register	SYIPADDRR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1603
000C 4840h	EPTPC 0	SYNFP Specification Version Setting Register	SYSPVRR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1604
000C 4844h	EPTPC 0	SYNFP Domain Number Setting Register	SYDOMR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1605
000C 4850h	EPTPC 0	Announce Message Flag Field Setting Register	ANFR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1606
000C 4854h	EPTPC 0	Sync Message Flag Field Setting Register	SYNFR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1607
000C 4858h	EPTPC 0	Delay_Req Message Flag Field Setting Register	DYRQFR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1608
000C 485Ch	EPTPC 0	Delay_Resp Message Flag Field Setting Register	DYRPFR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1609
000C 4860h	EPTPC 0	SYNFP Local Clock ID Registers	SYCIDRU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1610
000C 4864h	EPTPC 0	SYNFP Local Clock ID Registers	SYCIDRL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1610
000C 4868h	EPTPC 0	SYNFP Local Port Number Register	SYPNUMR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1611
000C 4880h	EPTPC 0	SYNFP Register Value Load Directive Register	SYRVLDR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1612
000C 4890h	EPTPC 0	SYNFP Reception Filter Register 1	SYRFL1R	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1614
000C 4894h	EPTPC 0	SYNFP Reception Filter Register 2	SYRFL2R	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1616
000C 4898h	EPTPC 0	SYNFP Transmission Enable Register	SYTRENR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1617
000C 48A0h	EPTPC 0	Master Clock ID Register	MTCIDU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1618
000C 48A4h	EPTPC 0	Master Clock ID Register	MTCIDL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1618
000C 48A8h	EPTPC 0	Master Clock Port Number Register	MTPID	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1619
000C 48C0h	EPTPC 0	SYNFP Transmission Interval Setting Register	SYTLIR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1620
000C 48C4h	EPTPC 0	SYNFP Received logMessageInterval Value Indication Register	SYRLIR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1621
000C 48C8h	EPTPC 0	offsetFromMaster Value Register	OFMRU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1622
000C 48CCh	EPTPC 0	offsetFromMaster Value Register	OFMRL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1622
000C 48D0h	EPTPC 0	meanPathDelay Value Register	MPDRU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1623
000C 48D4h	EPTPC 0	meanPathDelay Value Register	MPDRL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1623
000C 48E0h	EPTPC 0	grandmasterPriority Field Setting Register	GMPR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1624
000C 48E4h	EPTPC 0	grandmasterClockQuality Field Setting Register	GMCQR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1625
000C 48E8h	EPTPC 0	grandmasterIdentity Field Setting Registers	GMIDRU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1626
000C 48ECh	EPTPC 0	grandmasterIdentity Field Setting Registers	GMIDRL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1626

Table 5.1 List of I/O Registers (Address Order) (59 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000C 48F0h	EPTPC 0	currentUtcOffset/timeSource Field Setting Register	CUOTSR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1627
000C 48F4h	EPTPC 0	stepsRemoved Field Setting Register	SRR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1627
000C 4900h	EPTPC 0	PTP-primary Message Destination MAC Address Setting Registers	PPMACRU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1628
000C 4904h	EPTPC 0	PTP-primary Message Destination MAC Address Setting Registers	PPMACRL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1628
000C 4908h	EPTPC 0	PTP-pdelay Message MAC Address Setting Registers	PDMACRU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1629
000C 490Ch	EPTPC 0	PTP-pdelay Message MAC Address Setting Registers	PDMACRL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1629
000C 4910h	EPTPC 0	PTP Message EtherType Setting Register	PETYPER	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1630
000C 4920h	EPTPC 0	PTP-primary Message Destination IP Address Setting Register	PPIPR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1631
000C 4924h	EPTPC 0	PTP-pdelay Message Destination IP Address Setting Register	PDIPR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1631
000C 4928h	EPTPC 0	PTP event Message TOS Setting Register	PETOSR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1632
000C 492Ch	EPTPC 0	PTP general Message TOS Setting Register	PGTOSR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1633
000C 4930h	EPTPC 0	PTP-primary Message TTL Setting Register	PPTTLR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1633
000C 4934h	EPTPC 0	PTP-pdelay Message TTL Setting Register	PDTTLR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1634
000C 4938h	EPTPC 0	PTP event Message UDP Destination Port Number Setting Register	PEUDPR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1634
000C 493Ch	EPTPC 0	PTP general Message UDP Destination Port Number Setting Register	PGUDPR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1635
000C 4940h	EPTPC 0	Frame Reception Filter Setting Register	FFLTR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1636
000C 4960h	EPTPC 0	Frame Reception Filter MAC Address 0 Setting Registers	FMAC0RU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1637
000C 4964h	EPTPC 0	Frame Reception Filter MAC Address 0 Setting Registers	FMAC0RL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1637
000C 4968h	EPTPC 0	Frame Reception Filter MAC Address 1 Setting Registers	FMAC1RU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1638
000C 496Ch	EPTPC 0	Frame Reception Filter MAC Address 1 Setting Registers	FMAC1RL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1638
000C 49C0h	EPTPC 0	Asymmetric Delay Setting Register	DASYMRU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1639
000C 49C4h	EPTPC 0	Asymmetric Delay Setting Register	DASYMRL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1639
000C 49C8h	EPTPC 0	Timestamp Latency Setting Register	TSLATR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1640
000C 49CCh	EPTPC 0	SYNFP Operation Setting Register	SYCONFR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1641
000C 49D0h	EPTPC 0	SYNFP Frame Format Setting Register	SYFORMR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1642
000C 49D4h	EPTPC 0	Response Message Reception Timeout Register	RSTOUTR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1643
000C 4C00h	EPTPC 1	SYNFP Status Register	SYSR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1599
000C 4C04h	EPTPC 1	SYNFP Status Notification Permission Register	SYIPR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1601
000C 4C10h	EPTPC 1	SYNFP MAC Address Registers	SYMACRU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1602
000C 4C14h	EPTPC 1	SYNFP MAC Address Registers	SYMACRL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1602
000C 4C1Ch	EPTPC 1	SYNFP Local IP Address Register	SYIPADDRR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1603

Table 5.1 List of I/O Registers (Address Order) (60 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000C 4C40h	EPTPC 1	SYNFP Specification Version Setting Register	SYSPVRR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1604
000C 4C44h	EPTPC 1	SYNFP Domain Number Setting Register	SYDOMR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1605
000C 4C50h	EPTPC 1	Announce Message Flag Field Setting Register	ANFR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1606
000C 4C54h	EPTPC 1	Sync Message Flag Field Setting Register	SYNFR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1607
000C 4C58h	EPTPC 1	Delay_Req Message Flag Field Setting Register	DYRQFR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1608
000C 4C5Ch	EPTPC 1	Delay_Resp Message Flag Field Setting Register	DYRPFR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1609
000C 4C60h	EPTPC 1	SYNFP Local Clock ID Registers	SYCIDRU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1610
000C 4C64h	EPTPC 1	SYNFP Local Clock ID Registers	SYCIDRL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1610
000C 4C68h	EPTPC 1	SYNFP Local Port Number Register	SYPNUMR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1611
000C 4C80h	EPTPC 1	SYNFP Register Value Load Directive Register	SYRVLDR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1612
000C 4C90h	EPTPC 1	SYNFP Reception Filter Register 1	SYRFL1R	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1614
000C 4C94h	EPTPC 1	SYNFP Reception Filter Register 2	SYRFL2R	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1616
000C 4C98h	EPTPC 1	SYNFP Transmission Enable Register	SYTRENR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1617
000C 4CA0h	EPTPC 1	Master Clock ID Register	MTCIDU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1618
000C 4CA4h	EPTPC 1	Master Clock ID Register	MTCIDL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1618
000C 4CA8h	EPTPC 1	Master Clock Port Number Register	MTPID	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1619
000C 4CC0h	EPTPC 1	SYNFP Transmission Interval Setting Register	SYTLIR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1620
000C 4CC4h	EPTPC 1	SYNFP Received logMessageInterval Value Indication Register	SYRLIR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1621
000C 4CC8h	EPTPC 1	offsetFromMaster Value Register	OFMRU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1622
000C 4CCCh	EPTPC 1	offsetFromMaster Value Register	OFMRL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1622
000C 4CD0h	EPTPC 1	meanPathDelay Value Register	MPDRU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1623
000C 4CD4h	EPTPC 1	meanPathDelay Value Register	MPDRL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1623
000C 4CE0h	EPTPC 1	grandmasterPriority Field Setting Register	GMPR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1624
000C 4CE4h	EPTPC 1	grandmasterClockQuality Field Setting Register	GMCQR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1625
000C 4CE8h	EPTPC 1	grandmasterIdentity Field Setting Registers	GMIDRU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1626
000C 4CECh	EPTPC 1	grandmasterIdentity Field Setting Registers	GMIDRL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1626
000C 4CF0h	EPTPC 1	currentUtcOffset/timeSource Field Setting Register	CUOTSR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1627
000C 4CF4h	EPTPC 1	stepsRemoved Field Setting Register	SRR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1627
000C 4D00h	EPTPC 1	PTP-primary Message Destination MAC Address Setting Registers	PPMACRU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1628
000C 4D04h	EPTPC 1	PTP-primary Message Destination MAC Address Setting Registers	PPMACRL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1628
000C 4D08h	EPTPC 1	PTP-pdelay Message MAC Address Setting Registers	PDMACRU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1629

Table 5.1 List of I/O Registers (Address Order) (61 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000C 4D0Ch	EPTPC1	PTP-pdelay Message MAC Address Setting Registers	PDMACRL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1629
000C 4D10h	EPTPC1	PTP Message EtherType Setting Register	PETYPER	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1630
000C 4D20h	EPTPC1	PTP-primary Message Destination IP Address Setting Register	PPIPR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1631
000C 4D24h	EPTPC1	PTP-pdelay Message Destination IP Address Setting Register	PDIPR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1631
000C 4D28h	EPTPC1	PTP event Message TOS Setting Register	PETOSR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1632
000C 4D2Ch	EPTPC1	PTP general Message TOS Setting Register	PGTOSR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1633
000C 4D30h	EPTPC1	PTP-primary Message TTL Setting Register	PPTTLR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1633
000C 4D34h	EPTPC1	PTP-pdelay Message TTL Setting Register	PDTTLR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1634
000C 4D38h	EPTPC1	PTP event Message UDP Destination Port Number Setting Register	PEUDPR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1634
000C 4D3Ch	EPTPC1	PTP general Message UDP Destination Port Number Setting Register	PGUDPR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1635
000C 4D40h	EPTPC1	Frame Reception Filter Setting Register	FFLTR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1636
000C 4D60h	EPTPC1	Frame Reception Filter MAC Address 0 Setting Registers	FMAC0RU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1637
000C 4D64h	EPTPC1	Frame Reception Filter MAC Address 0 Setting Registers	FMAC0RL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1637
000C 4D68h	EPTPC1	Frame Reception Filter MAC Address 1 Setting Registers	FMAC1RU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1638
000C 4D6Ch	EPTPC1	Frame Reception Filter MAC Address 1 Setting Registers	FMAC1RL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1638
000C 4DC0h	EPTPC1	Asymmetric Delay Setting Register	DASYMRU	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1639
000C 4DC4h	EPTPC1	Asymmetric Delay Setting Register	DASYMRL	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1639
000C 4DC8h	EPTPC1	Timestamp Latency Setting Register	TSLATR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1640
000C 4DCCh	EPTPC1	SYNFP Operation Setting Register	SYCONFR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1641
000C 4DD0h	EPTPC1	SYNFP Frame Format Setting Register	SYFORMR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1642
000C 4DD4h	EPTPC1	Response Message Reception Timeout Register	RSTOUTR	32	32	9 to 211 PCLKA	2 to 106 ICLK	EPTPC	1643
000D 0000h	SCIFA8	Serial Mode Register	SMR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2106
000D 0002h	SCIFA8	Bit Rate Register	BRR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2112
000D 0002h	SCIFA8	Modulation Duty Register	MDDR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2118
000D 0004h	SCIFA8	Serial Control Register	SCR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2107
000D 0006h	SCIFA8	Transmit FIFO Data Register	FTDR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2105
000D 0008h	SCIFA8	Serial Status Register	FSR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2109
000D 000Ah	SCIFA8	Receive FIFO Data Register	FRDR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2105
000D 000Ch	SCIFA8	FIFO Control Register	FCR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2119
000D 000Eh	SCIFA8	FIFO Data Count Register	FDR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2120
000D 0010h	SCIFA8	Serial Port Register	SPTR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2121
000D 0012h	SCIFA8	Line Status Register	LSR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2123
000D 0014h	SCIFA8	Serial Extended Mode Register	SEMR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2125
000D 0016h	SCIFA8	FIFO Trigger Control Register	FTCR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2124
000D 0020h	SCIFA9	Serial Mode Register	SMR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2106
000D 0022h	SCIFA9	Bit Rate Register	BRR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2112
000D 0022h	SCIFA9	Modulation Duty Register	MDDR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2118
000D 0024h	SCIFA9	Serial Control Register	SCR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2107

Table 5.1 List of I/O Registers (Address Order) (62 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000D 0026h	SCIFA9	Transmit FIFO Data Register	FTDR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2105
000D 0028h	SCIFA9	Serial Status Register	FSR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2109
000D 002Ah	SCIFA9	Receive FIFO Data Register	FRDR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2105
000D 002Ch	SCIFA9	FIFO Control Register	FCR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2119
000D 002Eh	SCIFA9	FIFO Data Count Register	FDR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2120
000D 0030h	SCIFA9	Serial Port Register	SPTR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2121
000D 0032h	SCIFA9	Line Status Register	LSR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2123
000D 0034h	SCIFA9	Serial Extended Mode Register	SEMR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2125
000D 0036h	SCIFA9	FIFO Trigger Control Register	FTCR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2124
000D 0040h	SCIFA10	Serial Mode Register	SMR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2106
000D 0042h	SCIFA10	Bit Rate Register	BRR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2112
000D 0042h	SCIFA10	Modulation Duty Register	MDDR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2118
000D 0044h	SCIFA10	Serial Control Register	SCR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2107
000D 0046h	SCIFA10	Transmit FIFO Data Register	FTDR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2105
000D 0048h	SCIFA10	Serial Status Register	FSR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2109
000D 004Ah	SCIFA10	Receive FIFO Data Register	FRDR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2105
000D 004Ch	SCIFA10	FIFO Control Register	FCR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2119
000D 004Eh	SCIFA10	FIFO Data Count Register	FDR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2120
000D 0050h	SCIFA10	Serial Port Register	SPTR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2121
000D 0052h	SCIFA10	Line Status Register	LSR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2123
000D 0054h	SCIFA10	Serial Extended Mode Register	SEMR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2125
000D 0056h	SCIFA10	FIFO Trigger Control Register	FTCR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2124
000D 0060h	SCIFA11	Serial Mode Register	SMR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2106
000D 0062h	SCIFA11	Bit Rate Register	BRR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2112
000D 0062h	SCIFA11	Modulation Duty Register	MDDR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2118
000D 0064h	SCIFA11	Serial Control Register	SCR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2107
000D 0066h	SCIFA11	Transmit FIFO Data Register	FTDR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2105
000D 0068h	SCIFA11	Serial Status Register	FSR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2109
000D 006Ah	SCIFA11	Receive FIFO Data Register	FRDR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2105
000D 006Ch	SCIFA11	FIFO Control Register	FCR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2119
000D 006Eh	SCIFA11	FIFO Data Count Register	FDR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2120
000D 0070h	SCIFA11	Serial Port Register	SPTR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2121
000D 0072h	SCIFA11	Line Status Register	LSR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2123
000D 0074h	SCIFA11	Serial Extended Mode Register	SEMR	8	8	3, 4 PCLKB	2 ICLK	SCIFA	2125
000D 0076h	SCIFA11	FIFO Trigger Control Register	FTCR	16	16	3, 4 PCLKB	2 ICLK	SCIFA	2124

Table 5.1 List of I/O Registers (Address Order) (63 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000D 0100h	RSPI0	RSPI Control Register	SPCR	8	8	3, 4 PCLKA	2 ICLK	RSPIa	2284
000D 0101h	RSPI0	RSPI Slave Select Polarity Register	SSLP	8	8	3, 4 PCLKA	2 ICLK	RSPIa	2286
000D 0102h	RSPI0	RSPI Pin Control Register	SPPCR	8	8	3, 4 PCLKA	2 ICLK	RSPIa	2287
000D 0103h	RSPI0	RSPI Status Register	SPSR	8	8	3, 4 PCLKA	2 ICLK	RSPIa	2288
000D 0104h	RSPI0	RSPI Data Register	SPDR	32	16, 32	3, 4 PCLKA	2 ICLK	RSPIa	2291
000D 0108h	RSPI0	RSPI Sequence Control Register	SPSCR	8	8	3, 4 PCLKA	2 ICLK	RSPIa	2294
000D 0109h	RSPI0	RSPI Sequence Status Register	SPSSR	8	8	3, 4 PCLKA	2 ICLK	RSPIa	2295
000D 010Ah	RSPI0	RSPI Bit Rate Register	SPBR	8	8	3, 4 PCLKA	2 ICLK	RSPIa	2296
000D 010Bh	RSPI0	RSPI Data Control Register	SPDCR	8	8	3, 4 PCLKA	2 ICLK	RSPIa	2297
000D 010Ch	RSPI0	RSPI Clock Delay Register	SPCKD	8	8	3, 4 PCLKA	2 ICLK	RSPIa	2299
000D 010Dh	RSPI0	RSPI Slave Select Negation Delay Register	SSLND	8	8	3, 4 PCLKA	2 ICLK	RSPIa	2300
000D 010Eh	RSPI0	RSPI Next-Access Delay Register	SPND	8	8	3, 4 PCLKA	2 ICLK	RSPIa	2301
000D 010Fh	RSPI0	RSPI Control Register 2	SPCR2	8	8	3, 4 PCLKA	2 ICLK	RSPIa	2302
000D 0110h	RSPI0	RSPI Command Register 0	SPCMD0	16	16	3, 4 PCLKA	2 ICLK	RSPIa	2304
000D 0112h	RSPI0	RSPI Command Register 1	SPCMD1	16	16	3, 4 PCLKA	2 ICLK	RSPIa	2304
000D 0114h	RSPI0	RSPI Command Register 2	SPCMD2	16	16	3, 4 PCLKA	2 ICLK	RSPIa	2304
000D 0116h	RSPI0	RSPI Command Register 3	SPCMD3	16	16	3, 4 PCLKA	2 ICLK	RSPIa	2304
000D 0118h	RSPI0	RSPI Command Register 4	SPCMD4	16	16	3, 4 PCLKA	2 ICLK	RSPIa	2304
000D 011Ah	RSPI0	RSPI Command Register 5	SPCMD5	16	16	3, 4 PCLKA	2 ICLK	RSPIa	2304
000D 011Ch	RSPI0	RSPI Command Register 6	SPCMD6	16	16	3, 4 PCLKA	2 ICLK	RSPIa	2304
000D 011Eh	RSPI0	RSPI Command Register 7	SPCMD7	16	16	3, 4 PCLKA	2 ICLK	RSPIa	2304
000D 0400h	USBA	System Configuration Control Register	SYSCFG	16	16	3, 4 PCLKB	2 ICLK	USBA	1841
000D 0402h	USBA	CPU Bus Wait Register	BUSWAIT	16	16	3, 4 PCLKB	2 ICLK	USBA	1843
000D 0404h	USBA	System Configuration Status Register	SYSSTS0	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1844
000D 0406h	USBA	PLL Status Register	PLLSTA	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1845
000D 0408h	USBA	Device State Control Register 0	DVSTCTR0	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1846
000D 0414h	USBA	CFIFO Port Register	CFIFO	32	8,16,32	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1849
000D 0418h	USBA	D0FIFO Port Register	D0FIFO	32	8,16,32	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1854
000D 041Ch	USBA	D1FIFO Port Register	D1FIFO	32	8,16,32	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1854

Table 5.1 List of I/O Registers (Address Order) (64 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000D 0420h	USBA	CFIFO Port Select Register	CFIFOSEL	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1852
000D 0422h	USBA	CFIFO Port Control Register	CFIFOCTR	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1856
000D 0428h	USBA	D0FIFO Port Select Register	D0FIFOSEL	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1854
000D 042Ah	USBA	D0FIFO Port Control Register	D0FIFOCTR	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1856
000D 042Ch	USBA	D1FIFO Port Select Register	D1FIFOSEL	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1854
000D 042Eh	USBA	D1FIFO Port Control Register	D1FIFOCTR	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1856
000D 0430h	USBA	Interrupt Enable Register 0	INTENB0	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1858
000D 0432h	USBA	Interrupt Enable Register 1	INTENB1	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1859
000D 0436h	USBA	BRDY Interrupt Enable Register	BRDYENB	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1860
000D 0438h	USBA	NRDY Interrupt Enable Register	NRDYENB	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1860
000D 043Ah	USBA	BEMP Interrupt Enable Register	BEMPENB	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1861
000D 043Ch	USBA	SOF Output Configuration Register	SOFCFG	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1862

Table 5.1 List of I/O Registers (Address Order) (65 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000D 043Eh	USBA	PHY Setting Register	PHYSET	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1863
000D 0440h	USBA	Interrupt Status Register 0	INTSTS0	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1864
000D 0442h	USBA	Interrupt Status Register 1	INTSTS1	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1866
000D 0446h	USBA	BRDY Interrupt Status Register	BRDYSTS	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1869
000D 0448h	USBA	NRDY Interrupt Status Register	NRDYSTS	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1869
000D 044Ah	USBA	BEMP Interrupt Status Register	BEMPSTS	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1870
000D 044Ch	USBA	Frame Number Register	FRMNUM	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1871
000D 044Eh	USBA	µFrame Number Register	UFRMNUM	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1871
000D 0450h	USBA	USB Address Register	USBADDR	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1872
000D 0454h	USBA	USB Request Type Register	USBREQ	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1872
000D 0456h	USBA	USB Request Value Register	USBVAL	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1873
000D 0458h	USBA	USB Request Index Register	USBINDX	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1873

Table 5.1 List of I/O Registers (Address Order) (66 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000D 045Ah	USBA	USB Request Length Register	USBLENG	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1874
000D 045Ch	USBA	DCP Configuration Register	DCPCFG	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1875
000D 045Eh	USBA	DCP Maximum Packet Size Register	DCPMAXP	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1876
000D 0460h	USBA	DCP Control Register	DCPCTR	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1877
000D 0464h	USBA	Pipe Window Select Register	PIPESEL	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1880
000D 0468h	USBA	Pipe Configuration Register	PIPECFG	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1881
000D 046Ah	USBA	Pipe Buffer Register	PIPEBUF	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1884
000D 046Ch	USBA	Pipe Maximum Packet Size Register	PIPEMAXP	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1885
000D 046Eh	USBA	Pipe Cycle Control Register	PIPEPERI	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1886
000D 0470h	USBA	Pipe1 Control Register	PIPE1CTR	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1887
000D 0472h	USBA	Pipe2 Control Register	PIPE2CTR	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1887
000D 0474h	USBA	Pipe3 Control Register	PIPE3CTR	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1887

Table 5.1 List of I/O Registers (Address Order) (67 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000D 0476h	USBA	Pipe4 Control Register	PIPE4CTR	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1887
000D 0478h	USBA	Pipe5 Control Register	PIPE5CTR	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1887
000D 047Ah	USBA	Pipe6 Control Register	PIPE6CTR	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1887
000D 047Ch	USBA	Pipe7 Control Register	PIPE7CTR	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1887
000D 047Eh	USBA	Pipe8 Control Register	PIPE8CTR	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1887
000D 0480h	USBA	Pipe9 Control Register	PIPE9CTR	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1887
000D 0490h	USBA	Pipe1 Transaction Counter Enable Register	PIPE1TRE	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1892
000D 0492h	USBA	Pipe1 Transaction Counter Register	PIPE1TRN	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1893
000D 0494h	USBA	Pipe2 Transaction Counter Enable Register	PIPE2TRE	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1892
000D 0496h	USBA	Pipe2 Transaction Counter Register	PIPE2TRN	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1893
000D 0498h	USBA	Pipe3 Transaction Counter Enable Register	PIPE3TRE	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1892
000D 049Ah	USBA	Pipe3 Transaction Counter Register	PIPE3TRN	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1893

Table 5.1 List of I/O Registers (Address Order) (68 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000D 049Ch	USBA	Pipe4 Transaction Counter Enable Register	PIPE4TRE	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1892
000D 049Eh	USBA	Pipe4 Transaction Counter Register	PIPE4TRN	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1893
000D 04A0h	USBA	Pipe5 Transaction Counter Enable Register	PIPE5TRE	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1892
000D 04A2h	USBA	Pipe5 Transaction Counter Register	PIPE5TRN	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1893
000D 04D0h	USBA	Device Address 0 Configuration Register	DEVADD0	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1894
000D 04D2h	USBA	Device Address 1 Configuration Register	DEVADD1	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1894
000D 04D4h	USBA	Device Address 2 Configuration Register	DEVADD2	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1894
000D 04D6h	USBA	Device Address 3 Configuration Register	DEVADD3	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1894
000D 04D8h	USBA	Device Address 4 Configuration Register	DEVADD4	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1894
000D 04DAh	USBA	Device Address 5 Configuration Register	DEVADD5	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1894
000D 0500h	USBA	Low Power Control Register	LPCTRL	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1895
000D 0502h	USBA	Low Power Status Register	LPSTS	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1896

Table 5.1 List of I/O Registers (Address Order) (69 / 69)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000D 0540h	USBA	Battery Charging Control Register	BCCTRL	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1897
000D 0544h	USBA	Function L1 Control Register 1	PL1CTRL1	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1899
000D 0546h	USBA	Function L1 Control Register 2	PL1CTRL2	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1901
000D 0548h	USBA	Host L1 Control Register 1	HL1CTRL1	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1901
000D 054Ah	USBA	Host L1 Control Register 2	HL1CTRL2	16	16	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1902
000D 0560h	USBA	Deep Standby USB Transceiver Control/Pin Monitor Register	DPUSR0R	32	32	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1903
000D 0564h	USBA	Deep Standby USB Suspend/Resume Interrupt Register	DPUSR1R	32	32	(3 + BUSWAIT) PCLKA or more	Rounded up to the nearest integer greater than 1 + (3 + BUSWAIT) × (frequency ratio of ICLK/ PCLKB)*5	USBA	1904

Note 1. When the same output trigger is specified for pulse output groups 2 and 3 by the PPG0.PCR setting, the PPG0.NDRH address is 0008 81ECh. When different output triggers are specified, the PPG0.NDRH addresses for pulse output groups 2 and 3 are 0008 81EEh and 0008 81ECh, respectively.

Note 2. When the same output trigger is specified for pulse output groups 0 and 1 by the PPG0.PCR setting, the PPG0.NDRL address is 0008 81EDh. When different output triggers are specified, the PPG0.NDRL addresses for pulse output groups 0 and 1 are 0008 81EFh and 0008 81EDh, respectively.

Note 3. When the same output trigger is specified for pulse output groups 6 and 7 by the PPG1.PCR setting, the PPG1.NDRH address is 0008 81FCh. When different output triggers are specified, the PPG1.NDRH addresses for pulse output groups 6 and 7 are 0008 81FEh and 0008 81FCh, respectively.

Note 4. When the same output trigger is specified for pulse output groups 4 and 5 by the PPG1.PCR setting, the PPG1.NDRL address is 0008 81FDh. When different output triggers are specified, the PPG1.NDRL addresses for pulse output groups 4 and 5 are 0008 81FFh and 0008 81FDh, respectively.

Note 5. When the register is accessed while the USB is operating, a delay may be generated in accessing.

Note 6. The address must end with 0h, 4h, 8h, or Ch when access is made in 32-bit units. The address must end with 0h, 2h, 4h, 6h, 8h, Ah, Ch, or Eh when access is made in 16-bit units.

6. Resets

6.1 Overview

There are nine types of resets: RES# pin reset, power-on reset, voltage-monitoring 0 reset, voltage-monitoring 1 reset, voltage-monitoring 2 reset, deep software standby reset, independent watchdog timer reset, watchdog timer reset, and software reset.

Table 6.1 lists the reset names and sources.

Table 6.1 Reset Names and Sources

Reset Name	Source
RES# pin reset	Voltage input to the RES# pin is driven low.
Power-on reset	VCC rises (voltage detection: VPOR)*1
Voltage-monitoring 0 reset	VCC falls (voltage detection: Vdet0)*1
Voltage-monitoring 1 reset	VCC falls (voltage detection: Vdet1)*1
Voltage-monitoring 2 reset	VCC falls (voltage detection: Vdet2)*1
Deep software standby reset	Deep software standby mode is canceled by an interrupt.
Independent watchdog timer reset	The independent watchdog timer underflows, or a refresh error occurs.
Watchdog timer reset	The watchdog timer underflows, or a refresh error occurs.
Software reset	Register setting

Note 1. For details on the voltages to be monitored (VPOR, Vdet0, Vdet1, and Vdet2), see section 8, Voltage Detection Circuit (LVDA) and section 64, Electrical Characteristics.

The internal state and pins are initialized by a reset.

Table 6.2 lists the reset targets to be initialized.

Table 6.2 Targets to be Initialized by Each Reset Source

Targets to be Initialized	Reset Source								
	RES# Pin Reset	Power-On Reset	Voltage-Monitoring 0 Reset	Independent Watchdog Timer Reset	Watchdog Timer Reset	Voltage-Monitoring 1 Reset	Voltage-Monitoring 2 Reset	Deep Software Standby Reset	Software Reset
Power-on reset detect flag (RSTSR0.PORF)	○	—	—	—	—	—	—	—	—
Cold start/warm start determination flag (RSTSR1.CWSF)	—	○	—	—	—	—	—	—	—
Voltage-monitoring 0 reset detect flag (RSTSR0.LVD0RF)	○	○	—	—	—	—	—	—	—
Independent watchdog timer reset detect flag (RSTSR2.IWDTRF)	○	○	○	—	—	—	—	○	—
Independent watchdog timer (IWDTRR, IWDTCR, IWDTSR, IWDTRCR, IWDTCSTPR, ILOCOCR)	○	○	○	—	—	—	—	○	—
Watchdog timer reset detect flag (RSTSR2.WDTRF)	○	○	○	○	—	—	—	○	—
Registers related to the watchdog timer (WDTRR, WDTCR, WDTSR, WDTRCR)	○	○	○	○	—	—	—	○	—
Voltage-monitoring 1 reset detect flag (RSTSR0.LVD1RF)	○	○	○	○	○	—	—	—	—
Registers related to the voltage monitor function 1 (LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL[3:0])	○	○	○	○	○	—	—	—	—
(LVD1CR1, LVD1SR)	○	○	○	○	○	—	—	○	—
Voltage-monitoring 2 reset detect flag (RSTSR0.LVD2RF)	○	○	○	○	○	○	—	—	—
Registers related to the voltage monitor function 2 (LVD2CR0, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL[3:0])	○	○	○	○	○	○	—	—	—
(LVD2CR1, LVD2SR)	○	○	○	○	○	○	—	○	—
Deep software standby reset detect flag (RSTSR0.DPSRSTF)	○	○	○	○	○	○	○	—	—
Software reset detect flag (RSTSR2.SWRF)	○	○	○	○	○	○	○	○	—
Register related to the realtime clock*1	—	—	—	—	—	—	—	—	—
Register related to high-speed on-chip oscillator (HOCOPCR.HOCOPCNT)	○	○	○	○	○	○	○	—	○
Register related to main clock oscillator (MOFCR)	○	○	○	○	○	○	○	—	○
Pin state	○	○	○	○	○	○	○	—	○
Registers related to the low power-consumption function (DPSBYCR, DPSIER0 to DPSIER3, DPSIFR0 to DPSIFR3, DPSIEGR0 to DPSIEGR3, DPUSR0R, DPUSR1R)*2	○	○	○	○	○	○	○	—	○
Registers other than the above, CPU, and internal state	○	○	○	○	○	○	○	○	○

○: Targets to be initialized, —: No change occurs.
Note 1. Some control bits (RCR1.CIE, RCR2.RTCOE, ADJ30, and RESET) are initialized by all types of resets. For details on the target bits, refer to section 32, Realtime Clock (RTCd).
Note 2. Of the registers related to the low-power-consumption function, the DPSBKRy register is not initialized by any reset. For details, see section 11, Low Power Consumption.

When a reset is canceled, the reset exception handling starts. For details on the reset exception handling, see section 14, Exception Handling.
Table 6.3 lists the pin related to the reset.

Table 6.3 Pin Related to Reset

Pin Name	I/O	Function
RES#	Input	Reset pin

6.2 Register Descriptions

6.2.1 Reset Status Register 0 (RSTSR0)

Address(es): 0008 C290h

	b7	b6	b5	b4	b3	b2	b1	b0
	DPSRSTF	—	—	—	LVD2RF	LVD1RF	LVD0RF	PORF
Value after reset:	0*1	0	0	0	0*1	0*1	0*1	0*1

Bit	Symbol	Bit Name	Description	R/W
b0	PORF	Power-On Reset Detect Flag	0: Power-on reset not detected. 1: Power-on reset detected.	R(/W) *2
b1	LVD0RF	Voltage-Monitoring 0 Reset Detect Flag	0: Voltage-monitoring 0 reset not detected. 1: Voltage-monitoring 0 reset detected.	R(/W) *2
b2	LVD1RF	Voltage-Monitoring 1 Reset Detect Flag	0: Voltage-monitoring 1 reset not detected. 1: Voltage-monitoring 1 reset detected.	R(/W) *2
b3	LVD2RF	Voltage-Monitoring 2 Reset Detect Flag	0: Voltage-monitoring 2 reset not detected. 1: Voltage-monitoring 2 reset detected.	R(/W) *2
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DPSRSTF	Deep Software Standby Reset Flag	0: Deep software standby mode cancelation not requested by an interrupt. 1: Deep software standby mode cancelation requested by an interrupt.	R(/W) *2

Note 1. The value after reset depends on the reset source.
Note 2. Only 0 can be written to clear the flag.

PORF Flag (Power-On Reset Detect Flag)

The PORF flag indicates that a power-on reset has occurred.
[Setting condition]

- When a power-on reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When PORF is read as 1 and then 0 is written to PORF.

LVD0RF Flag (Voltage-Monitoring 0 Reset Detect Flag)

The LVD0RF flag indicates that a voltage-monitoring 0 reset has occurred due to the VCC voltage falling below Vdet0.
[Setting condition]

- When a voltage-monitoring 0 reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When LVD0RF is read as 1 and then 0 is written to LVD0RF.

LVD1RF Flag (Voltage-Monitoring 1 Reset Detect Flag)

The LVD1RF flag indicates that a voltage-monitoring 1 reset has occurred due to the VCC voltage falling below Vdet1.
[Setting condition]

- When a voltage-monitoring 1 reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When LVD1RF is read as 1 and then 0 is written to LVD1RF.

LVD2RF Flag (Voltage-Monitoring 2 Reset Detect Flag)

The LVD2RF flag indicates that a voltage-monitoring 2 reset has occurred due to the VCC voltage falling below Vdet2.
[Setting condition]

- When a voltage-monitoring 2 reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When LVD2RF is read as 1 and then 0 is written to LVD2RF.

DPSRSTF Flag (Deep Software Standby Reset Flag)

The DPSRSTF flag indicates that deep software standby mode has been canceled by an interrupt and that an internal reset (deep software standby reset) occurred.
[Setting condition]

- When deep software standby mode is cancelled by an interrupt.
  For details, see section 11, Low Power Consumption.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When DPSRSTF is read as 1 and then 0 is written to DPSRSTF.

6.2.2 Reset Status Register 1 (RSTSR1)

Address(es): 0008 C291h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CWSF
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Bit Name	Description	R/W
b0	CWSF	Cold/Warm Start Determination Flag	0: Cold start 1: Warm start	R(/W) *2
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The value after reset depends on the reset source.
Note 2. Only 1 can be written to set the flag.

RSTSR1 determines whether a power-on reset has caused the reset processing (cold start) or a reset signal input during operation has caused the reset processing (warm start).

CWSF Flag (Cold/Warm Start Determination Flag)

The CWSF flag indicates the type of reset processing: cold start or warm start.
The CWSF flag is initialized by a power-on reset. It is not initialized by a reset signal generated by the RES# pin.
[Setting condition]

- When 1 is written through programming; it is not set to 0 even when 0 is written.

[Clearing condition]

- When a reset listed in Table 6.2 occurs.

6.2.3 Reset Status Register 2 (RSTSR2)

Address(es): 0008 00C0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SWRF	WDTRF	IWDTRF
Value after reset:	0	0	0	0	0	0*1	0*1	0*1

Bit	Symbol	Bit Name	Description	R/W
b0	IWDTRF	Independent Watchdog Timer Reset Detect Flag	0: Independent watchdog timer reset not detected. 1: Independent watchdog timer reset detected.	R(/W) *2
b1	WDTRF	Watchdog Timer Reset Detect Flag	0: Watchdog timer reset not detected. 1: Watchdog timer reset detected.	R(/W) *2
b2	SWRF	Software Reset Detect Flag	0: Software reset not detected. 1: Software reset detected.	R(/W) *2
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The value after reset depends on the reset source.
Note 2. Only 0 can be written to clear the flag.

IWDTRF Flag (Independent Watchdog Timer Reset Detect Flag)

The IWDTRF flag indicates that an independent watchdog timer reset has occurred.
[Setting condition]

- When an independent watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When IWDTRF is read as 1 and then 0 is written to IWDTRF.

WDTRF Flag (Watchdog Timer Reset Detect Flag)

The WDTRF flag indicates that a watchdog timer reset has occurred.
[Setting condition]

- When a watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When WDTRF is read as 1 and then 0 is written to WDTRF.

SWRF Flag (Software Reset Detect Flag)

The SWRF flag indicates that a software reset has occurred.
[Setting condition]

- When a software reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When SWRF is read as 1 and then 0 is written to SWRF.

6.2.4 Software Reset Register (SWRR)

Address(es): 0008 00C2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SWRR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	SWRR[15:0]	Software Reset	Writing A501h resets the LSI. These bits are read as 0000h.	R/W

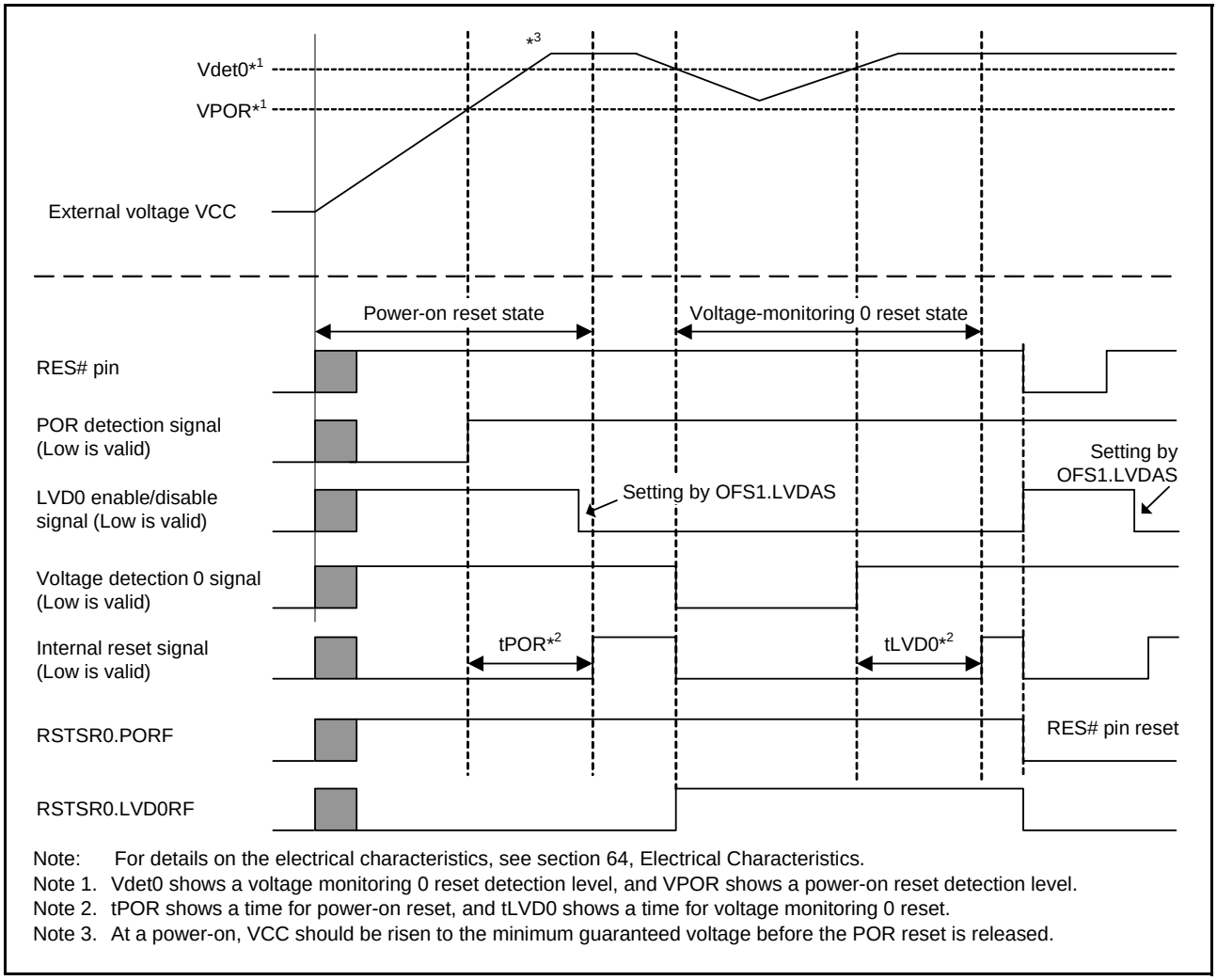
6.3 Operation

6.3.1 RES# Pin Reset

This is a reset generated by the RES# pin.
When the RES# pin is driven low, all the processing in progress is aborted and the LSI enters a reset state.
In order to unfailingly reset the LSI, the RES# pin should be held low for the specified power supply stabilization time at a power-on.
When the RES# pin is driven high from low, the internal reset is canceled after the post-RES# cancelation wait time (tRESWT) has elapsed, and then the CPU starts the reset exception handling.
For details, see section 64, Electrical Characteristics.

6.3.2 Power-On Reset and Voltage Monitoring 0 Reset

The power-on reset is an internal reset generated by the power-on reset circuit.
If the RES# pin is in a high level state when power is supplied, a power-on reset is generated. In addition, if the RES# pin is in a high level state when power falls (including the case when VCC falls below VPOR), a power-on reset is generated.
After VCC has exceeded VPOR and the specified period (power-on reset time) has elapsed, the internal reset is canceled and the CPU starts the reset exception handling. The power-on reset time is used for the stabilization of the power supply and the LSI circuit.
After a power-on reset has been generated, the PORF flag in the RSTSR0 is set to 1. The PORF flag is initialized by the RES# pin reset.
The voltage monitoring 0 reset is an internal reset generated by the voltage monitoring circuit. If the voltage detection 0 circuit start (LVDAS) bit in the option function select register 1 (OFS1) is 0 (voltage monitoring 0 reset is enabled after a reset) and VCC falls below Vdet0, the RSTSR0.LVD0RF flag becomes 1 and the voltage detection circuit generates voltage monitoring 0 reset. Clear the OFS1.LVDAS bit to 0 if the voltage monitoring 0 reset is to be used.
After VCC has exceeded Vdet0 and the voltage-monitoring 0 reset time (tLVD0) has elapsed, the internal reset is canceled and the CPU starts the reset exception handling. The Vdet0 voltage detection level can be changed by the setting of the VDSEL[1:0] bits in the option function select register 1 (OFS1).
Figure 6.1 shows operations during a power-on reset and voltage monitoring 0 reset.
For details on voltage monitoring 0 reset, refer to section 8, Voltage Detection Circuit (LVDA).



Note: For details on the electrical characteristics, see section 64, Electrical Characteristics.
Note 1. Vdet0 shows a voltage monitoring 0 reset detection level, and VPOR shows a power-on reset detection level.
Note 2. tPOR shows a time for power-on reset, and tLVD0 shows a time for voltage monitoring 0 reset.
Note 3. At a power-on, VCC should be risen to the minimum guaranteed voltage before the POR reset is released.

Figure 6.1 Operation Examples During a Power-On Reset and Voltage Monitoring 0 Reset

6.3.3 Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset

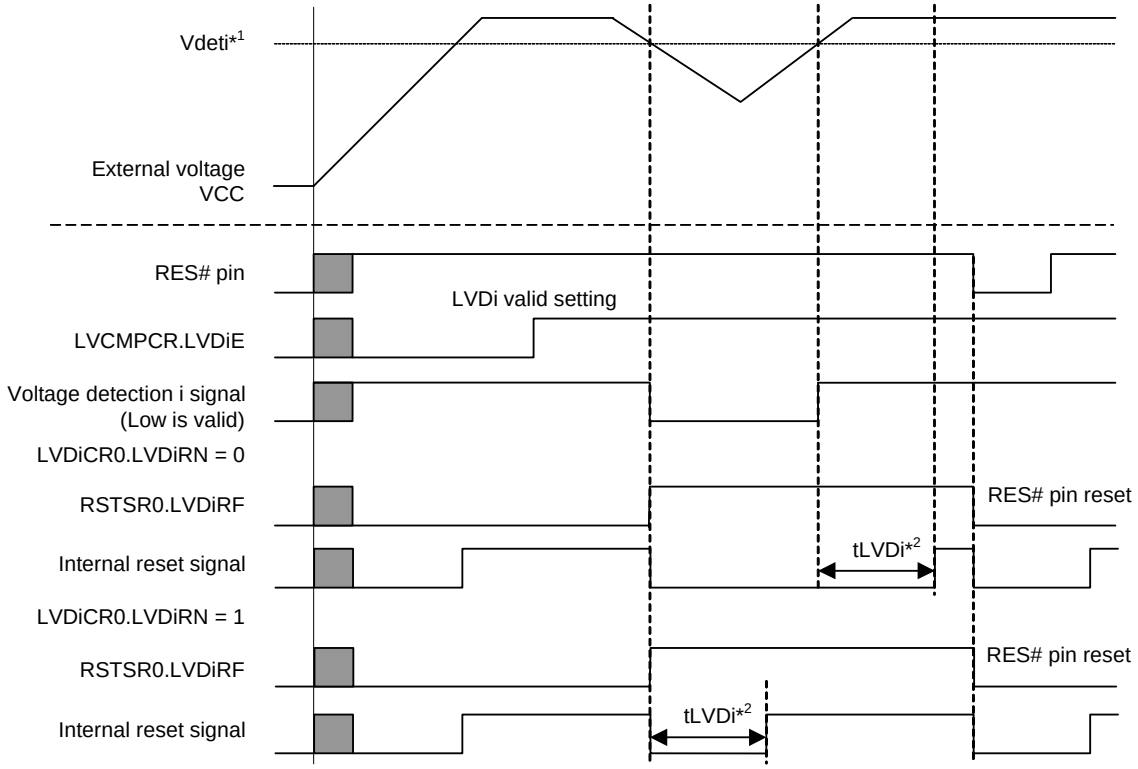
The voltage monitoring 1 reset and voltage monitoring 2 reset are internal resets generated by the voltage monitoring circuit.

When the voltage monitoring 1 interrupt/reset enable bit (LVD1RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the voltage monitoring 1 circuit mode select bit (LVD1RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in the voltage monitoring 1 circuit control register 0 (LVD1CR0), the RSTSR0.LVD1RF flag is set to 1 and the voltage-detection circuit generates a voltage monitoring 1 reset if VCC falls to or below Vdet1.

Likewise, when the voltage monitoring 2 interrupt/reset enable bit (LVD2RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the voltage monitoring 2 circuit mode select bit (LVD2RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in voltage monitoring 2 circuit control register 0 (LVD2CR0), the RSTSR0.LVD2RF flag is set to 1 and the voltage detection circuit generates a voltage monitoring 2 reset if VCC falls to or below Vdet2.

Timing for release from the voltage monitoring 1 reset state is selectable with the voltage monitoring 1 reset negate select bit (LVD1RN) in the LVD1CR0. When the LVD1CR0.LVD1RN bit is 0 and VCC has fallen to or below Vdet1, the CPU is released from the internal reset state and starts reset exception handling once the voltage-monitoring 1 reset time (tLVD1) has elapsed after VCC has risen above Vdet1. When the LVD1CR0.LVD1RN bit is 1 and VCC has fallen to or below Vdet1, the CPU is released from the internal reset state and starts reset exception handling once the voltage-

monitoring 1 reset time (tLVD1) has elapsed.
Likewise, timing for release from the voltage monitoring 2 reset state is selectable by setting the voltage monitoring 2 reset negate select bit (LVD2RN) in LVD2CR0 register.
Detection levels Vdet1 and Vdet2 can be changed by settings in the voltage detection select register (LVDLVLR).
Figure 6.2 shows examples of operations during voltage monitoring 1 and 2 resets.
For details on the voltage monitoring 1 reset and voltage monitoring 2 reset, refer to section 8, Voltage Detection Circuit (LVDA).



Note: For details on the electrical characteristics, see section 64, Electrical Characteristics.
Note 1. Vdeti shows a detection level of voltage monitoring 1 reset and voltage monitoring 2 reset.
Note 2. tLVDi shows a time for voltage monitoring 1 reset and voltage monitoring 2 reset. (i = 1, 2)

Figure 6.2 Operation Examples During Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset

6.3.4 Deep Software Standby Reset

This is an internal reset generated when deep software standby mode is canceled by an interrupt.
When a deep software standby mode cancelation source is generated, a deep software standby reset is generated. The deep software standby reset is canceled after tDSBY (recovery time after cancellation of deep software standby mode) has elapsed. At the same time, deep software standby mode is also canceled.
When tDSBYWT (wait time after cancellation of deep software standby mode) has elapsed after deep software standby mode has been canceled, the internal reset is canceled and the CPU starts the reset exception handling.
For details of the deep software standby reset, see section 11, Low Power Consumption.

6.3.5 Independent Watchdog Timer Reset

Independent watchdog timer reset is an internal reset generated by the independent watchdog timer.
Output of the independent watchdog timer reset from the independent watchdog timer can be selected by settings in the IWDT reset control register (IWDTRCR) or option function select register 0 (OFS0).
When output of the independent watchdog timer reset is selected, an independent watchdog timer reset is generated if the independent watchdog timer underflows, or if data is written when refresh operation is disabled. When the internal reset time (tRESW2) has elapsed after the independent watchdog timer reset has been generated, the internal reset is canceled and the CPU starts the reset exception handling.
For details on the independent watchdog timer reset, see section 34, Independent Watchdog Timer (IWDTa).

6.3.6 Watchdog Timer Reset

The watchdog-timer reset is an internal reset from the watchdog timer.
Output of the independent watchdog timer reset from the independent watchdog timer can be selected by settings in the IWDT reset control register (IWDTRCR) or option function select register 0 (OFS0).
When output of the watchdog timer reset is selected, a watchdog timer reset is generated if the watchdog timer underflows, or if data is written when refresh operation is disabled. When the internal reset time (tRESW2) has elapsed after the watchdog timer reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.
For details on the watchdog timer reset, see section 33, Watchdog Timer (WDTA).

6.3.7 Software Reset

The software reset is an internal reset generated by the software reset circuit.
When A501h is written to SWRR, a software reset is generated. When the internal reset time (tRESW2) has elapsed after the software reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

6.3.8 Determination of Cold/Warm Start

By reading the CWSF flag in RSTSR1, the type of reset processing caused can be identified; that is, whether a power-on reset has caused the reset processing (cold start) or a reset signal input during operation has caused the reset processing (warm start).

The CWSF flag in RSTSR1 is set to 0 when a power-on reset occurs (cold start); otherwise the flag is not set to 0. The flag is set to 1 when 1 is written to it through programming; it is not set to 0 even when 0 is written.

Figure 6.3 shows an example of cold/warm start determination operation.

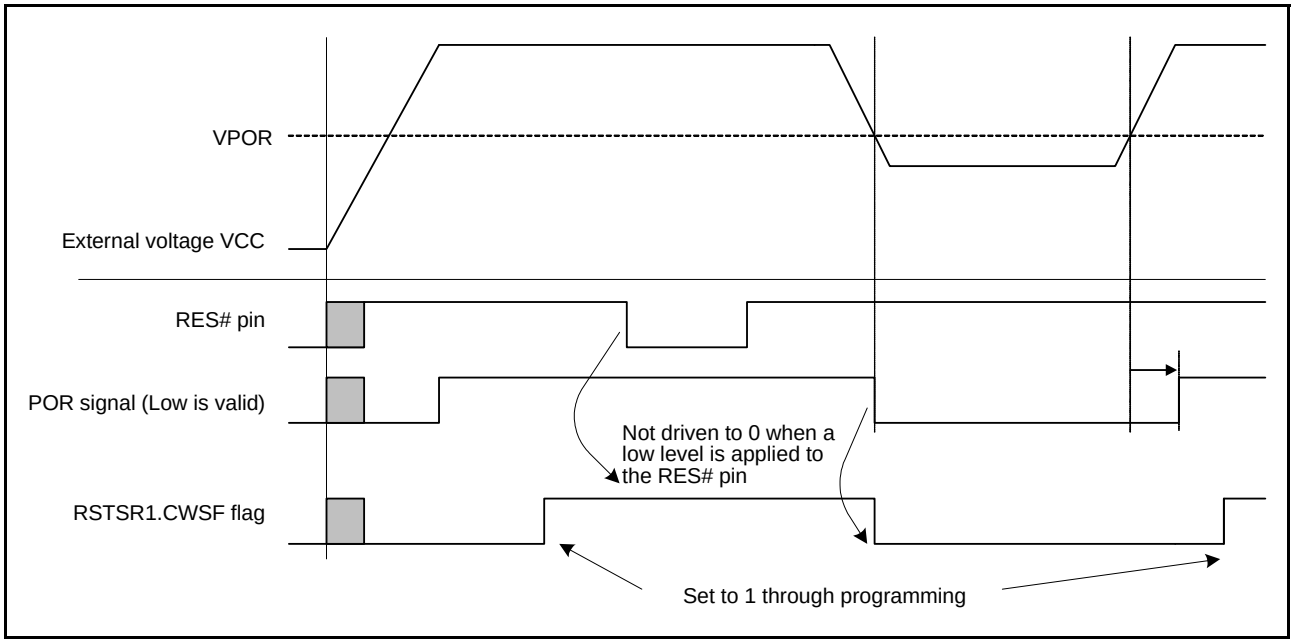


Figure 6.3 Example of Cold/Warm Start Determination Operation

6.3.9 Determination of Reset Generation Source

Reading RSTSR0 and RSTSR2 determines which reset was used to execute the reset exception handling. Figure 6.4 shows an example of the flow to identify a reset generation source.

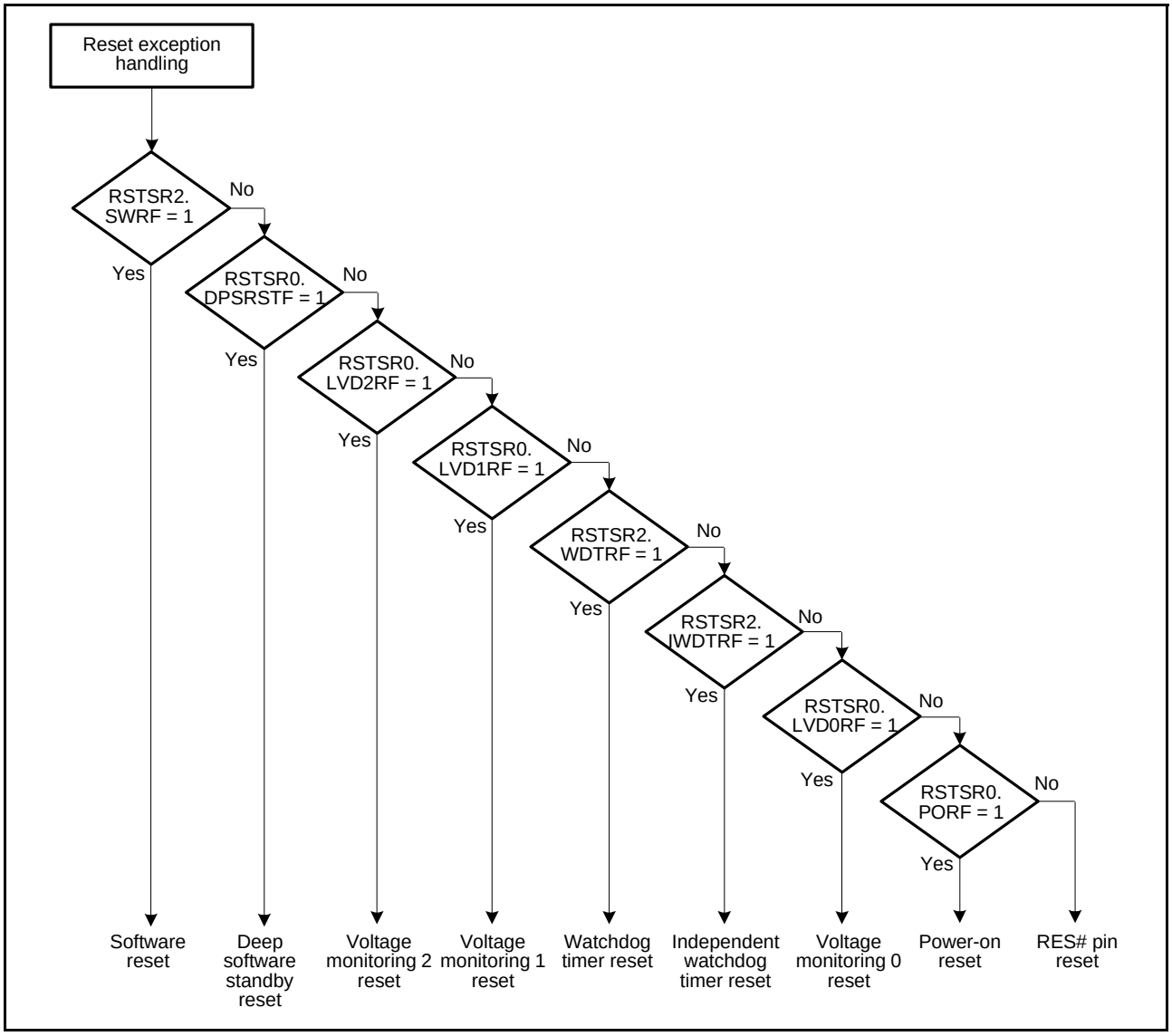


Figure 6.4 Example of Reset Generation Source Determination Flow

6.4 Usage Notes

6.4.1 Notes on Using Power-On Reset and PLL Circuit Together

When using a power-on reset and the PLL circuit together, set the LVD1CR1.LVD1IDTSEL[1:0] bits or LVD2CR1.LVD2IDTSEL[1:0] bits to 01b, and select the voltage monitoring interrupt to be generated when a drop (Vcc < Vdet) is detected.
In addition, at the beginning of the interrupt handling routine, set the SCKCR3.CKSEL[2:0] bits to a value other than 100b to select a clock source other than the PLL circuit, then set the PLLCR2.PLLEN bit to 1 to stop the PLL circuit.

7. Option-Setting Memory

7.1 Overview

The option-setting memory is a collective term for the registers listed below.

- Serial programmer command control register (SPCC)
- OCD/serial programmer ID setting register (OSIS)
- Option function select register 0 (OFS0)
- Option function select register 1 (OFS1)
- Endian select register (MDE)
- TM enable flag register (TMEF)
- TM identification data register (TMINF)
- UB code A
- UB code B

The option-setting memory determines the state of this MCU after a reset. Option-setting memory is allocated to the configuration setting area and user boot area of the flash memory, and the available methods of setting are different for the two areas. For the setting procedure, see section 7.6, Setting the Option-Setting Memory.
Figure 7.1 shows the option-setting memory area.

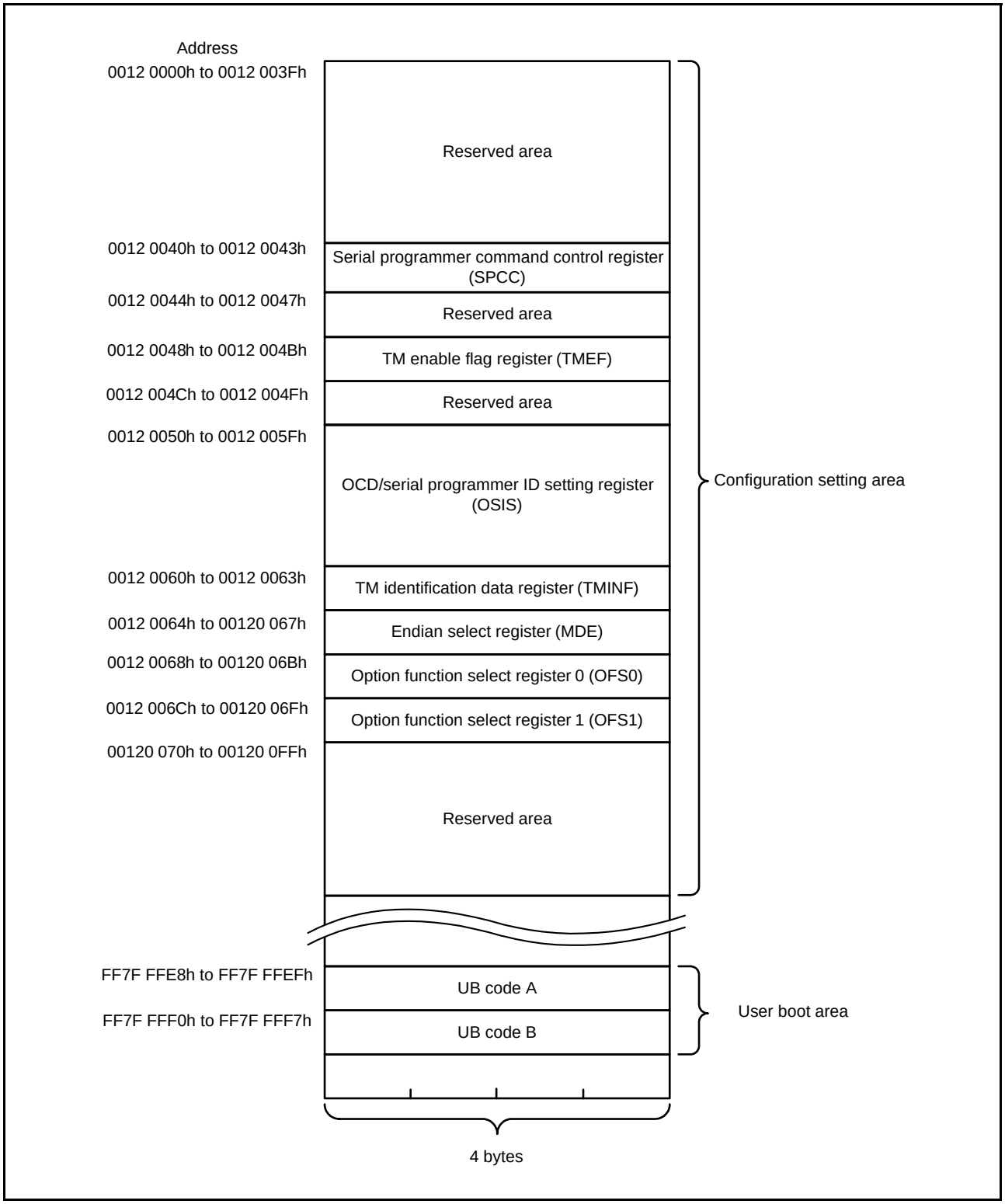


Figure 7.1 Option-Setting Memory Area

7.2 Register Descriptions

7.2.1 Serial Programmer Command Control Register (SPCC)

This register is used to enable or disable serial ID code protection, and to permit or prohibit the connection of a serial programmer, or the execution of block erasure commands, programming commands, or read commands.
The SPCC register should be set in 32-bit units.

Address(es): 0012 0040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RDPR	WRPR	SEPR	—	SPE	—	—	IDE	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b24	IDE	ID Code Protection Enable	0: ID code protection is enabled after a reset.*2 1: ID code protection is disabled after a reset.	R
b26, b25	—	Reserved	When reading, this bit returns to the value written by the user. The write value should be 1.	R
b27	SPE	Serial Programmer Connection Enable	0: Connection of a serial programmer is prohibited after a reset. 1: Connection of a serial programmer is permitted after a reset.	R
b28	—	Reserved	When reading, these bits return to the value written by the user. The write value should be 1.	R
b29	SEPR	Block Erasure Command Protect	0: Execution of block erasure commands is prohibited after a reset. 1: Execution of block erasure commands is permitted after a reset.	R
b30	WRPR	Programming Command Protect	0: Execution of programming commands is prohibited after a reset. 1: Execution of programming commands is permitted after a reset.	R
b31	RDPR	Read Command Protect	0: Execution of read commands is prohibited after a reset. 1: Execution of read commands is permitted after a reset.	R

Note 1. The value of the blank product is FFFF FFFFh. It is set to the written value after written by the user.
Note 2. When ID code protection is to be used, set the RDPR, SEPR, and WRPR bits to 0.

IDE Bit (ID Code Protection Enable)

This bit enables or disables ID code protection by a serial programmer. Using this function requires setting of the RDPR, WRPR, and SEPR bits to 0.

SPE Bit (Serial Programmer Connection Enable)

This bit enables or disables the connection of a serial programmer.

SEPR Bit (Block Erasure Command Protect)

This bit enables or disables execution of block erasure commands by the serial programmer.

WRPR Bit (Programming Command Protect)

This bit enables or disables execution of programming commands by the serial programmer.

RDPR Bit (Read Command Protect)

This bit enables or disables execution of read commands by the serial programmer.

7.2.2 OCD/Serial Programmer ID Setting Register (OSIS)

This register is used as the location for storage of the ID for ID code protection of the OCD/serial programmer.
When connecting the OCD/serial programmer, write values so that the chip can judge whether the connection is to be permitted.
This register checks whether a code transmitted from the OCD/serial programmer match the ID code in the option-setting memory.
When the ID codes match, connection of the OCD/serial programmer is permitted; if not, connection with the OCD/serial programmer is not possible.
Enabling ID code protection through the serial programmer requires setting of the IDE, SPE, RDPR, WRPR, and SEPR bits in the SPCC register, in addition to setting of this register.
The OSIS register should be set in 32-bit units.

Address	Bit 31			Bit 0
0012 0050h to 0012 0053h	OCD/serial ID4	OCD/serial ID3	OCD/serial ID2	OCD/serial ID1
0012 0054h to 0012 0057h	OCD/serial ID8	OCD/serial ID7	OCD/serial ID6	OCD/serial ID5
0012 0058h to 0012 005Bh	OCD/serial ID12	OCD/serial ID11	OCD/serial ID10	OCD/serial ID9
0012 005Ch to 0012 005Fh	OCD/serial ID16	OCD/serial ID15	OCD/serial ID14	OCD/serial ID13

OCD/Serial ID 1 to 16

These fields hold the ID for use in ID authentication for the OCD/serial programmer.

7.2.3 Option Function Select Register 0 (OFS0)

Address(es): 0012 0068h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	WDTRSTIRQS	WDTRPSS[1:0]		WDTRPES[1:0]		WDTCKS[3:0]				WDTTOPS[1:0]		WDTSTRT	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	IWDTSLCSTP	—	IWDTRSTIRQS	IWDTRPSS[1:0]		IWDTRPES[1:0]		IWDTCKS[3:0]				IWDTTOPS[1:0]		IWDTSTRT	—
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b1	IWDTSTRT	IWDT Start Mode Select	0: IWDT is automatically activated in auto-start mode after a reset 1: IWDT is halted after a reset	R
b3, b2	IWDTTOPS[1:0]	IWDT Timeout Period Select	b3 b2 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R
b7 to b4	IWDTCKS[3:0]	IWDT-Dedicated Clock Frequency Division Ratio Select	b7 b4 0 0 0 0: No division 0 0 1 0: Divide-by-16 0 0 1 1: Divide-by-32 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 0 1: Divide-by-256 Settings other than above are prohibited.	R
b9, b8	IWDTRPES[1:0]	IWDT Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (No window end position setting)	R
b11, b10	IWDTRPSS[1:0]	IWDT Window Start Position Select	b11 b10 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (No window start position setting)	R
b12	IWDTRSTIRQS	IWDT Reset Interrupt Request Select	0: Non-maskable interrupt request or plain interrupt request is enabled 1: Reset is enabled	R
b13	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b14	IWDTSLCSTP	IWDT Sleep Mode Count Stop Control	0: Counting stop is disabled 1: Counting stop is enabled when entering sleep, software standby, deep software standby, or all-module clock stop mode	R
b16, b15	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b17	WDTSTRT	WDT Start Mode Select	0: WDT is automatically activated in auto-start mode after a reset 1: WDT is stopped after a reset	R

Bit	Symbol	Bit Name	Description	R/W
b19, b18	WDTTOPS[1:0]	WDT Timeout Period Select	b19 b18 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R
b23 to b20	WDTCKS[3:0]	WDT Clock Frequency Division Ratio Select	b23 b20 0 0 0 1: Divide-by-4 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 1 0: Divide-by-512 0 1 1 1: Divide-by-2048 1 0 0 0: Divide-by-8192 Settings other than above are prohibited.	R
b25, b24	WDTRPES[1:0]	WDT Window End Position Select	b25 b24 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (No window end position setting)	R
b27, b26	WDTRPSS[1:0]	WDT Window Start Position Select	b27 b26 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (No window start position setting)	R
b28	WDTRSTIRQS	WDT Reset Interrupt Request Select	0: Non-maskable interrupt request or plain interrupt request is enabled 1: Reset is enabled	R
b31 to b29	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. It is set to the written value after written by the user.

When erasing the block including the OFS0 register, the OFS0 register value becomes FFFF FFFFh.
The setting in the OFS0 register is ignored in user boot mode, and this register functions similarly when it is set to FFFF FFFFh.

IWDTSTRT Bit (IWDT Start Mode Select)

This bit selects the mode in which the IWDT is activated after a reset (stopped state or activated in auto-start mode). When activated in auto-start mode, the OFS0 register setting for the IWDT is effective.

IWDTTOPS[1:0] Bits (IWDT Timeout Period Select)

These bits select the timeout period, i.e. the time it takes for the down-counter to underflow, as 1024, 4096, 8192, or 16384 cycles of the frequency-divided clock set by the IWDTCKS[3:0] bits. The time (number of clock cycles for the IWDT) it takes to underflow after a refresh operation is determined by the combination of the IWDTCKS[3:0] bits and IWDTTOPS[1:0] bits.
For details, see section 34, Independent Watchdog Timer (IWDTa).

IWDTCKS[3:0] Bits (IWDT-Dedicated Clock Frequency Division Ratio Select)

These bits select, from 1/1, 1/16, 1/32, 1/64, 1/128, and 1/256, the division ratio of the prescaler to divide the frequency of the clock for the IWDT. Using the setting of these bits together with the IWDTTOPS[1:0] bit setting, the IWDT counting period can be set from 1024 to 4194304 clock cycles for the IWDT.
For details, see section 34, Independent Watchdog Timer (IWDTa).

IWDTRPES[1:0] Bits (IWDT Window End Position Select)

These bits select the position of the end of the window for the down-counter as 0%, 25%, 50%, or 75% of the value being counted by the counter. The value of the window end position must be smaller than the value of the window start position (window start position > window end position). If the value for the window end position is greater than the value for the

window start position, only the value for the window start position is effective.
The counter values corresponding to the settings for the start and end positions of the window in the IWDTRPSS[1:0] and IWDTRPES[1:0] bits vary with the setting of the IWDTTOPS[1:0] bits.
For details, refer to section 34, Independent Watchdog Timer (IWDTa).

IWDTRPSS[1:0] Bits (IWDT Window Start Position Select)

These bits select the position where the window for the down-counter starts as 25%, 50%, 75%, or 100% of the value being counted (the point at which counting starts is 100% and the point at which an underflow occurs is 0%). The interval between the positions where the window starts and ends becomes the period in which refreshing is possible, and refreshing is not possible outside this period.
For details, refer to section 34, Independent Watchdog Timer (IWDTa).

IWDTRSTIRQS Bit (IWDT Reset Interrupt Request Select)

The setting of this bit selects the operation on an underflow of the down-counter or generation of a refresh error. An independent watchdog timer reset, a non-maskable interrupt request, or a interrupt request is selectable.

IWDTSLCSTP Bit (IWDT Sleep Mode Count Stop Control)

This bit selects to stop counting when entering sleep, software standby, deep software standby, or all-module clock stop mode.
For details, see section 34, Independent Watchdog Timer (IWDTa).

WDTSTRT Bit (WDT Start Mode Select)

This bit selects the mode in which the WDT is activated after a reset (stopped state or activated in auto-start mode). When activated in auto-start mode, the OFS0 register setting for the WDT is effective.

WDTTOPS[1:0] Bits (WDT Timeout Period Select)

These bits select the timeout period, i.e. the time it takes for the down-counter to underflow, as 1024, 4096, 8192, or 16384 cycles of the frequency-divided clock set by the WDTCKS[3:0] bits. The time (number of PCLKB cycles) it takes to underflow after a refresh operation is determined by a combination of the WDTCKS[3:0] bits and WDTTOPS[1:0] bits.
For details, see section 33, Watchdog Timer (WDTA).

WDTCKS[3:0] Bits (WDT Clock Frequency Division Ratio Select)

These bits select, from 1/4, 1/64, 1/128, 1/512, 1/2048, and 1/8192, the division ratio of the prescaler to divide the frequency of PCLKB. Using the setting of these bits together with the WDTTOPS[1:0] bit setting, the WDT counting period can be set from 4096 to 134217728 PCLKB cycles.
For details, see section 33, Watchdog Timer (WDTA).

WDTRPES[1:0] Bits (WDT Window End Position Select)

These bits select the position of the end of the window on the down-counter as 0%, 25%, 50%, or 75% of the value being counted by the counter. The value of the window end position must be smaller than the value of the window start position (window start position > window end position). If the value for the window end position is greater than the value for the window start position, only the value for the window start position is effective.
The counter values corresponding to the settings for the start and end positions of the window in the WDTRPSS[1:0] and WDTRPES[1:0] bits vary with the setting of the WDTTOPS[1:0] bits.
For details, refer to section 33, Watchdog Timer (WDTA).

WDTRPSS[1:0] Bits (WDT Window Start Position Select)

These bits select the position where the window for the down-counter starts as 25%, 50%, 75%, or 100% of the value being counted (the point at which counting starts is 100% and the point at which an underflow occurs is 0%). The interval between the positions where the window starts and ends becomes the period in which refreshing is possible, and refreshing is not possible outside this period.
For details, refer to section 33, Watchdog Timer (WDTA).

WDTRSTIRQS Bit (WDT Reset Interrupt Request Select)

The setting of this bit selects the operation on an underflow of the down-counter or generation of a refresh error. A watchdog timer reset, a non-maskable interrupt request, or a interrupt request is selectable.
For details, refer to section 33, Watchdog Timer (WDTA).

7.2.4 Option Function Select Register 1 (OFS1)

Address(es): 0012 006Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	HOCOEN	—	—	—	—	—	LVDAS	VDSEL[1:0]	
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b1, b0	VDSEL[1:0]	Voltage Detection 0 Level Select	b1 b0 0 0: Reserved 0 1: Selects 2.94 V 1 0: Selects 2.87 V 1 1: Selects 2.80 V	R
b2	LVDAS	Voltage Detection 0 Circuit Start	0: Voltage monitor 0 reset is enabled after a reset 1: Voltage monitor 0 reset is disabled after a reset	R
b7 to b3	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b8	HOCOEN	HOCO Oscillation Enable	0: HOCO oscillation is enabled after a reset 1: HOCO oscillation is disabled after a reset	R
b31 to b9	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. It is set to the written value after written by the user.

When erasing the block including the OFS1 register, the OFS1 register value becomes FFFF FFFFh.
The setting in the OFS1 register is ignored in user boot mode, and this register functions similarly when it is set to FFFF FFFFh.

VDSEL[1:0] Bits (Voltage Detection 0 Level Select)

These bits select the voltage detection level of the voltage detection 0 circuit.

LVDAS Bit (Voltage Detection 0 Circuit Start)

This bit selects whether the voltage monitor 0 reset is enabled or disabled after a reset.

HOCOEN Bit (HOCO Oscillation Enable)

This bit selects whether the HOCO oscillation enable bit is effective or not after a reset.
Setting the HOCOEN bit to 0 allows the HOCO oscillation to be started before the CPU starts operation, and therefore reduces the waiting time for oscillation stabilization.
Note that even if the HOCOEN bit is set to 0, the system clock source is not switched to HOCO. The system clock source is switched to HOCO only by modifying the clock source select bits (SCKCR3.CKSEL[2:0]) from the CPU.

7.2.5 Endian Select Register (MDE)

Address(es): 0012 0064h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	MDE[2:0]		
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	MDE[2:0]	Endian Select	b2 b0 0 0 0: Big endian 1 1 1: Little endian Settings other than above are prohibited.	R
b31 to b3	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. It is set to the written value after written by the user.

This register selects the endian for the CPU.
When erasing the block including the MDE register, the MDE register value becomes FFFF FFFFh.

MDE[2:0] Bits (Endian Select)

These bits select little endian or big endian for the CPU.

7.2.6 TM Enable Flag Register (TMEF)

Address(es): 0012 0048h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	TMEF[2:0]			—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b26 to b24	TMEF[2:0]	TM Enable	b26 b24 0 0 0: TM function is enabled 1 1 1: TM function is disabled Settings other than above are prohibited.	R
b31 to b27	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. It is set to the written value after written by the user.
The TMEF register should be set in 32-bit units.

The TMEF register is used to enable the TM function for blocks 8 and 9 in the code flash memory. To enable the TM function, see section 63.12.36, TM Setting Command. When the TMEF[2:0] bits are rewritten while the TM function is enabled, rewriting these bits is ignored. To disable the TM function, see section 63.12.35, Configuration Clearing Command.

7.2.7 TM Identification Data Register (TMINF)

Address(es): 0012 0060h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	The value set by the user*1															

Note 1. The value of the blank product is FFFF FFFFh. It is set to the written value after written by the user.
The TMEF register should be set in 32-bit units.

The user can store any desired 32-bit value in this register.
The value acts as a code for identifying the programs in the TM-enabled area.
When the TMINF register is rewritten while the TM function is enabled, rewriting this register is ignored. To erase the contents of the TMINF register, see section 63.12.35, Configuration Clearing Command.

7.3 UB Codes

UB codes A and B are required if user boot mode is to be employed. This MCU will start up in user boot mode on release from the reset state if the four conditions below are satisfied.

- UB code A is 5573 6572h and 426F 6F74h.
- UB code B is FFFF FF07h and 0008 C04Ch.
- The low level is being input on the MD pin.
- The high level is being input on the UB pin.

7.3.1 UB Code A

UB code A consists of two 32-bit words. Set UB code A to 55736572h and 426F6F74h. Do not set any values other than these values in user boot mode. Set UB code A to FFFF FFFFh and FFFF FFFFh in any boot modes other than user boot mode.

Figure 7.2 shows the structure of UB code A in memory. Set UB code A in 32-bit units.

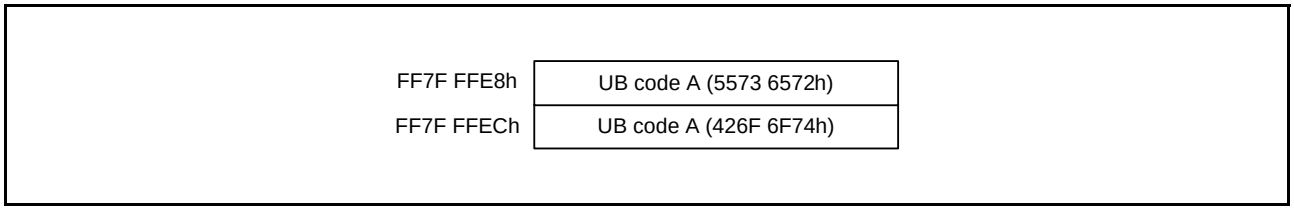


Figure 7.2 UB Code A Structure

7.3.2 UB Code B

UB code B consists of two words, i.e. 32 bits. Set UB code B to FFFF FF07h and 0008 C04Ch. Do not set any values other than these values in user boot mode. Set UB code B to FFFF FFFFh and FFFF FFFFh in any boot modes other than user boot mode.

Figure 7.3 shows the structure of UB code B in memory. Set UB code B in 32-bit units.

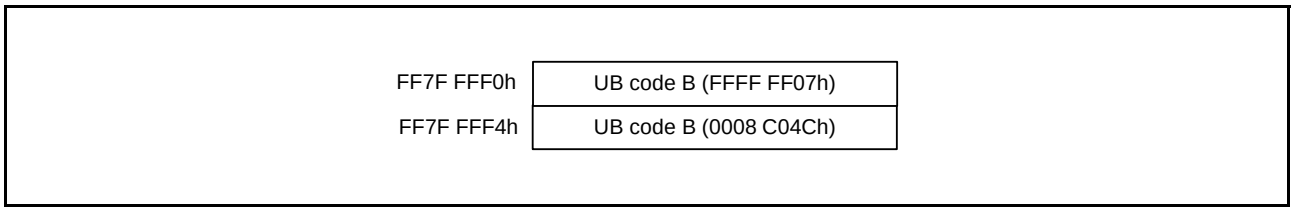


Figure 7.3 UB Code B Structure

7.4 Programming and Erasure of the Option-Setting Memory in Individual Operating Modes

Table 7.1 shows programming and erasure of the option-setting memory in the individual operating modes.

Table 7.1 Programming and Erasure of the Option-Setting Memory in Individual Operating Modes

	Boot Mode (SCI Interface, USB Interface)		Self-Programming		Parallel Writer	
Option-Setting Memory	**Programming**	**Erasure**	**Programming**	**Erasure**	**Programming**	**Erasure**
Serial programmer command control register (SPCC)	✓*1	✓*1	✓*2	×	✓*3	✓*3
OCD/serial programmer ID setting register (OSIS)	✓*1	✓*1	✓*2	×	✓*3	✓*3
Endian select register (MDE)	✓*1	✓*1	✓*2	×	✓*3	✓*3
Option function select register 0 (OFS0)	✓*1	✓*1	✓*2	×	✓*3	✓*3
Option function select register 1 (OFS1)	✓*1	✓*1	✓*2	×	✓*3	✓*3
TM enable flag register (TMEF)	✓*1	✓*1	✓*2	×	✓*3	✓*3
TM identification data register (TMINF)	✓*1	✓*1	✓*2	×	✓*3	✓*3
UB code A	✓*1	✓*1	×	×	✓*3	✓*3
UB code B	✓*1	✓*1	×	×	✓*3	✓*3

✓: Possible

×: Not possible

Note 1. The commands for boot mode (for the SCI and USB interfaces) are used for programming or erasure. For details, see section 63.10, Boot Mode.

Note 2. The configuration setting command is used for programming. For how to use the configuration setting command, see the Flash Memory User's Manual: Hardware Interface.

Note 3. The parallel writer is used for programming and erasure. For details, see the manual of the parallel writer you are using.

7.5 Settings of the Option-Setting Memory and ID Code Authentication, Reading, Programming, and Erasure

Table 7.2 shows the settings of the option-setting memory and ID code authentication, reading, programming, and erasure.

Table 7.2 Settings of the Option-Setting Memory and ID Code Authentication, Reading, Programming, and Erasure

No.	SPCC.SPE	SPCC.IDE	OSIS	SPCC.RDPR	SPCC.WRPR	SPCC.SEPR	Connection of a Serial Programmer	Reading, Programming and Erasure after the Connection of a Serial Programmer
1	0	0	Any value	Must be fixed to 0	Must be fixed to 0	Must be fixed to 0	Connection prohibited	—
2	0	1		x	x	x	Connection prohibited	—
3	1	0		Must be fixed to 0	Must be fixed to 0	Must be fixed to 0	ID code authentication*1	Reading permitted, programming permitted, erasure permitted
4	1	1		0	0	0	Connection permitted	Reading prohibited, programming prohibited, erasure prohibited
5				1	0	0	Connection permitted	Reading permitted, programming prohibited, erasure prohibited
6				0	1	0	Connection permitted	Reading prohibited, programming permitted, erasure prohibited
7				1	1	0	Connection permitted	Reading permitted, programming permitted, erasure prohibited
8				0	0	1	Connection permitted	Reading prohibited, programming prohibited, erasure permitted
9				1	0	1	Connection permitted	Reading permitted, programming prohibited, erasure permitted
10				0	1	1	Connection permitted	Reading prohibited, programming permitted, erasure permitted
11				1	1	1	Connection permitted	Reading permitted, programming permitted, erasure permitted

x: Don’t care

Note 1. This determines whether the ID code sent by the serial programmer matches the ID code set in the OSIS register. When the ID codes match, connection is permitted; if not, connection is not possible.

7.6 Setting the Option-Setting Memory

7.6.1 Allocation of Data in the Option-Setting Memory

Data for programming in the option-setting memory should be allocated to the addresses shown in Figure 7.1. The allocation of data is for use by tools such as a flash programming software or an on-chip debugger.

Note: Programming formats vary depending on the compiler. Refer to the compiler manual for details.

- Configuration setting area

An example of source code for setting the option-setting memory in the configuration setting area is shown below.

Setting 1EFFFFFFh in the serial programmer command control register (SPCC)
.ORG 000120040h
.LWORD 01EFFFFFFh

Setting the following ID codes in the OCD/serial programmer ID setting register (OSIS)
ID1 = 01h, ID2 = 02h, ID3 = 03h, ID4 = 04h, ID5 = 05h, ID6 = 06h, ID7 = 07h, ID8 = 08h
ID9 = 09h, ID10 = 0Ah, ID11 = 0Bh, ID12 = 0Ch, ID13 = 0Dh, ID14 = 0Eh, ID15 = 0Fh, ID16 = 10h
.ORG 000120050h
.LWORD 004030201h, 008070605h, 00C0B0A09h, 0100F0E0Dh

Setting FFFFFFF8h in the endian select register (MDE)
.ORG 000120064h
.LWORD 0FFFFFFF8h

Setting 01234567h in the option function select register 0 (OFS0)
.ORG 000120068h
.LWORD 001234567h

Setting 89ABCDEFh in the option function select register 1 (OFS1)
.ORG 00012006Ch
.LWORD 089ABCDEFh

- User boot area

An example of source code for setting the option-setting memory in the user boot area is shown below.

Setting UB codes A and B
.ORG 0FF7FFFE8h
.LWORD 055736572h, 0426F6F74h
.LWORD 0FFFFFF07h, 00008C04Ch

7.6.2 Setting Data for Programming the Option-Setting Memory

Simply allocating data according to the procedure described in section 7.6.1, Allocation of Data in the Option-Setting Memory, does not actually write the data to the option-setting memory; either of the actions described below is also required.

(1) Changing the Option-Setting Memory by Self-Programming

The configuration setting command can be used to write data to the option-setting memory in the configuration setting area.
For details of the configuration setting command, see the Flash Memory User's Manual: Hardware Interface. The UB code cannot be set by self-programming.

(2) Debugging through an OCD or Programming by a Flash Writer (Flash Programming Software or Parallel Writer)

Since the procedure will depend on the tool you are using, see the manual for the tool. There are two setting procedures:

- Read the data allocated as described in section 7.6.1, Allocation of Data in the Option-Setting Memory, from an object file or Motorola S-format file generated by the compiler, and write the data to the MCU.
- Use the tool's GUI to program the same data as in the allocation of values in section 7.6.1, Allocation of Data in the Option-Setting Memory.

Some tools require three Motorola S-format files, one for each of the user area, user boot area, and configuration setting area.
Figure 7.4 shows an example of setting the option-setting memory by using a Motorola S-format file for each area.

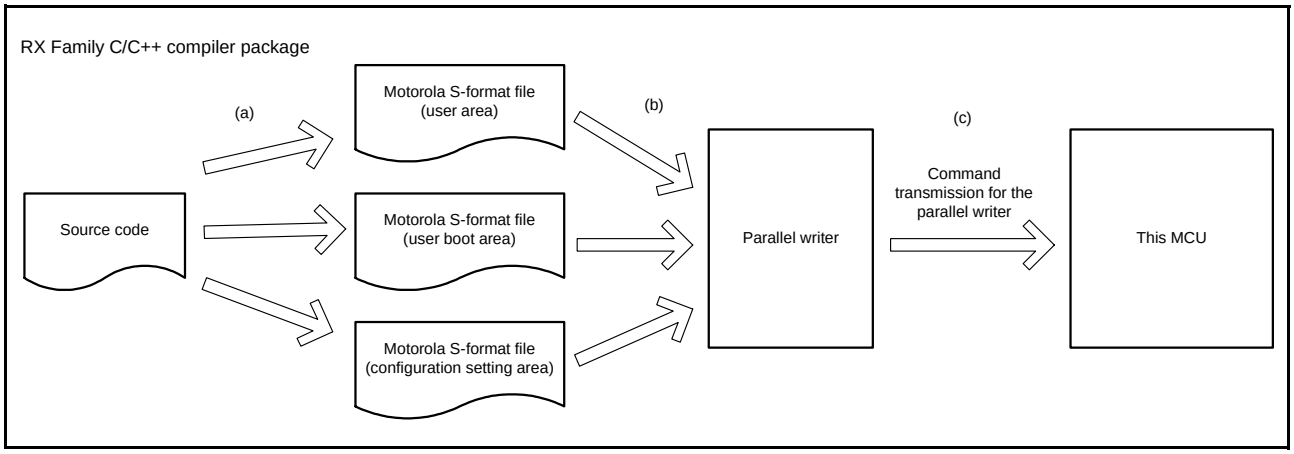


Figure 7.4 Example for Setting the Option-Setting Memory

(a) Use the RX Family C/C++ compiler package to produce separate Motorola S-format files for the user area, user boot area, and configuration setting area.
(b) Read the output Motorola S-format files with the parallel writer.
(c) The parallel writer transmits the data it has read to the MCU, and then writes the data to each of the user area, user boot area, and configuration setting area.

For output of the separate Motorola S-format files by the RX Family C/C++ compiler package, use the linker output option. For details, see the RX Family C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual.

7.7 Usage Note

7.7.1 Data for Programming Reserved Areas and Reserved Bits in the Option-Setting Memory

When reserved areas and reserved bits in the option-setting memory are within the scope of programming, write 1 as the value for all bits of reserved areas and all reserved bits. Normal operation cannot be guaranteed if 0 is written to such bits.

8. Voltage Detection Circuit (LVDA)

The voltage detection circuit (LVDA) monitors the voltage level input to the VCC pin using a program.

8.1 Overview

For voltage detection 0, the detection voltage is selectable from among three different levels and the reset from voltage monitoring 0 can be enabled or disabled after a reset by using the option function select register 1 (OFS1).
For voltage detection 1 and voltage detection 2, the detection voltage is selectable from among three different levels by using the voltage detection level select register (LVDLVLR).
The reset from voltage monitoring 0, reset/interrupt from voltage monitoring 1, and reset/interrupt from voltage monitoring 2 can be used.

Table 8.1 lists the specifications of the voltage detection circuit. Figure 8.1 is a block diagram of the voltage detection circuit. Figure 8.2 is a block diagram of the voltage monitoring 1 interrupt/reset circuit. Figure 8.3 is a block diagram of the voltage monitoring 2 interrupt/reset circuit.

Table 8.1 Voltage Detection Circuit Specifications

Item		Voltage Monitoring 0	Voltage Monitoring 1	Voltage Monitoring 2
VCC monitoring	Monitored voltage	Vdet0	Vdet1	Vdet2
	Detected event	Voltage drops past Vdet0	Voltage rises or drops past Vdet1	Voltage rises or drops past Vdet2
	Detection voltage	Selectable from among three different levels by using OFS1.VDSEL[1:0] bits	Selectable from among three different levels by using LVDLVLR.LVD1LVL[3:0] bits	Selectable from among three different levels by using LVDLVLR.LVD2LVL[3:0] bits
	Monitoring flag	None	LVD1SR.LVD1MON flag: Monitors whether voltage is higher or lower than Vdet1	LVD2SR.LVD2MON flag: Monitors whether voltage is higher or lower than Vdet2
			LVD1SR.LVD1DET flag: Vdet1 passage detection	LVD2SR.LVD2DET flag: Vdet2 passage detection
Process upon voltage detection	Reset	Voltage monitoring 0 reset	Voltage monitoring 1 reset	Voltage monitoring 2 reset
		Reset when Vdet0 > VCC CPU restart after specified time with VCC > Vdet0	Reset when Vdet1 > VCC CPU restart timing selectable: after specified time with VCC > Vdet1 or Vdet1 > VCC	Reset when Vdet2 > VCC CPU restart timing selectable: after specified time with VCC > Vdet2 or Vdet2 > VCC
	Interrupt	No interrupt	Voltage monitoring 1 interrupt	Voltage monitoring 2 interrupt
			Non-maskable interrupt or maskable interrupt selectable	Non-maskable interrupt or maskable interrupt selectable
			Interrupt request issued when Vdet1 > VCC and VCC > Vdet1 or either	Interrupt request issued when Vdet2 > VCC and VCC > Vdet2 or either
Digital filter	Enable/ Disable switching	Digital filter function not available	Available	Available
	Sampling time	—	1/n LOCO frequency × 2 (n: 2, 4, 8, 16)	1/n LOCO frequency × 2 (n: 2, 4, 8, 16)
Event linking		None	Available Output of event signals on detection of Vdet crossings	Available Output of event signals on detection of Vdet crossings

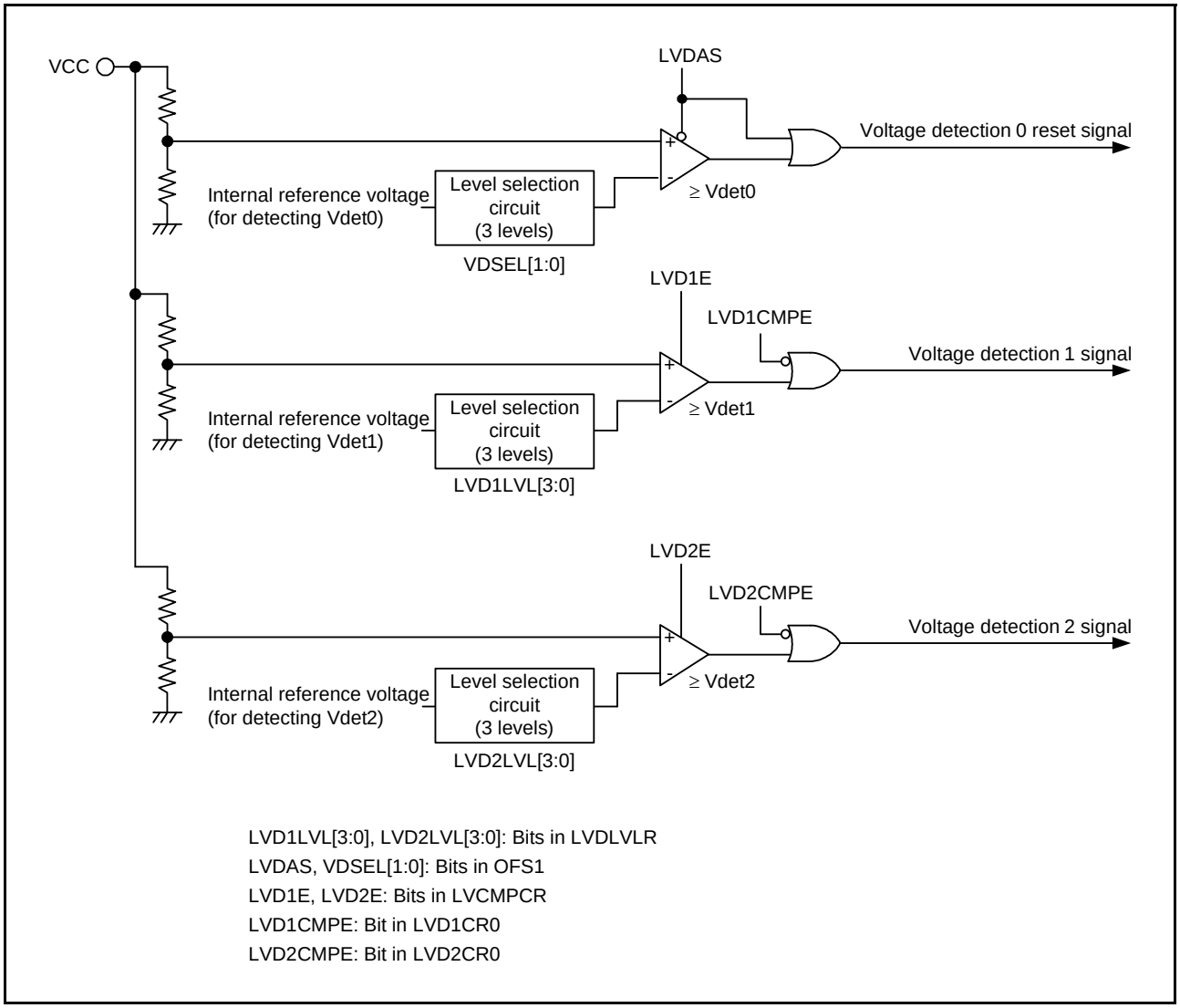


Figure 8.1 Block Diagram of Voltage Detection Circuit

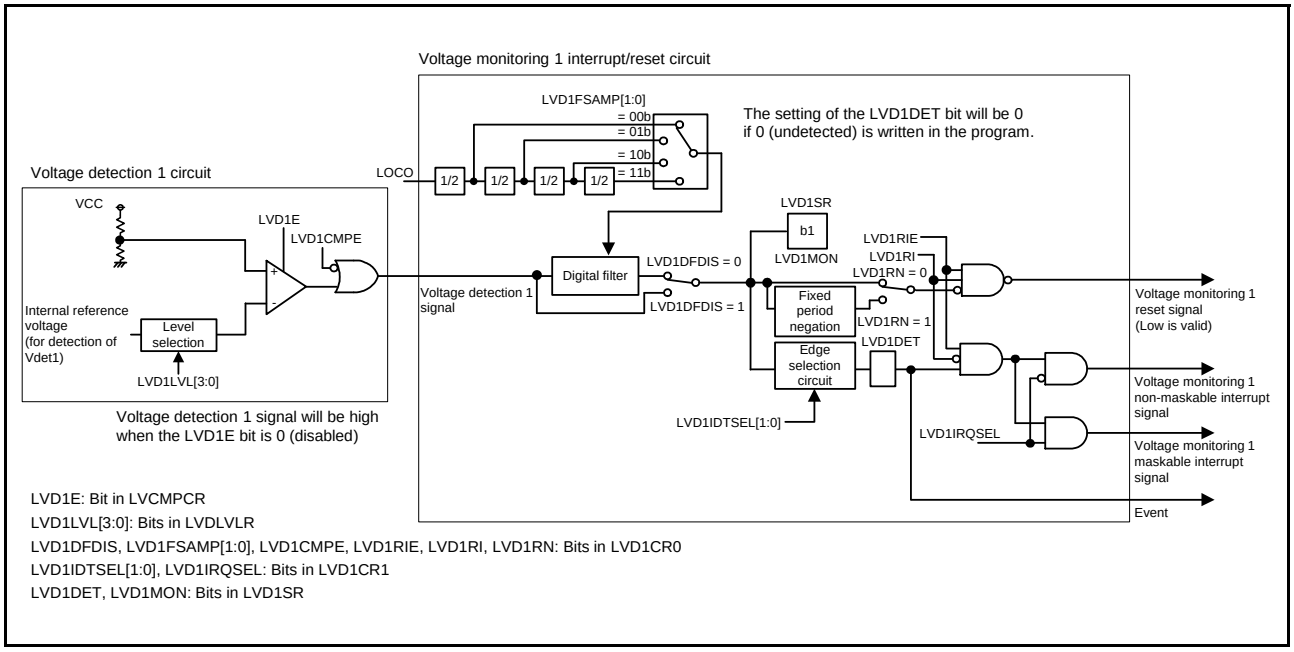


Figure 8.2 Block Diagram of Voltage Monitoring 1 Interrupt/Reset Circuit

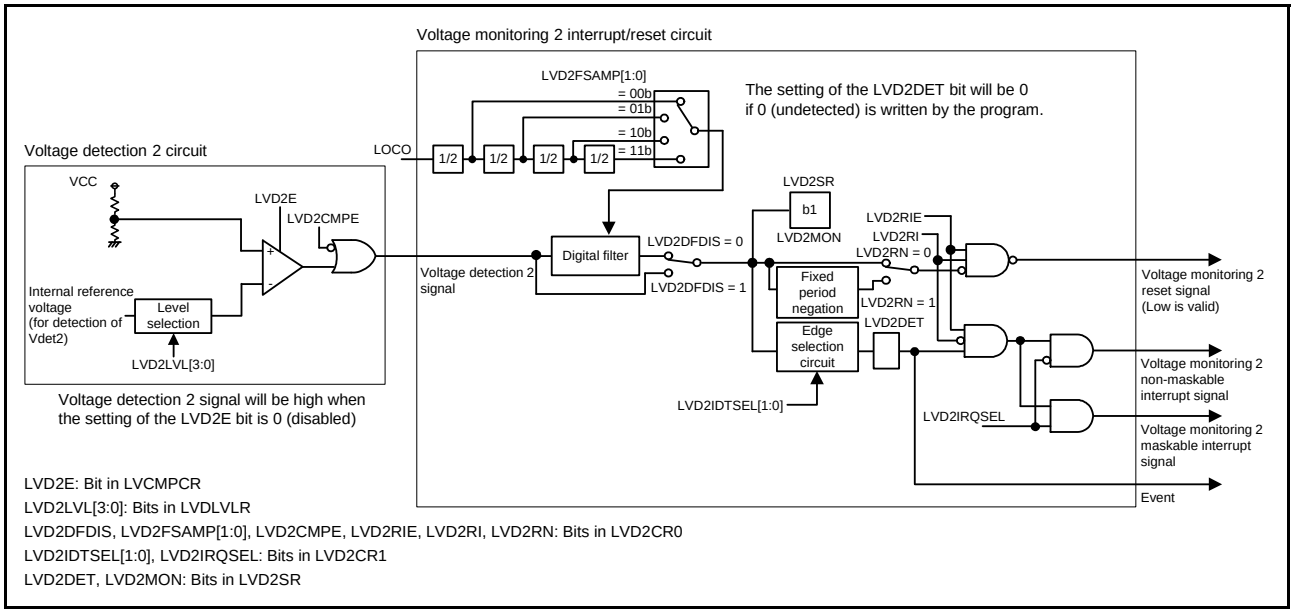


Figure 8.3 Block Diagram of Voltage Monitoring 2 Interrupt/Reset Circuit

8.2 Register Descriptions

8.2.1 Voltage Monitoring 1 Circuit Control Register 1 (LVD1CR1)

Address(es): 0008 00E0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	LVD1IRQSEL	LVD1IDTSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LVD1IDTSEL[1:0]	Voltage Monitoring 1 Interrupt Generation Condition Select	b1 b0 0 0: When VCC ≥ Vdet1 (rise) is detected 0 1: When VCC < Vdet1 (drop) is detected 1 0: When drop and rise are detected 1 1: Settings prohibited	R/W
b2	LVD1IRQSEL	Voltage Monitoring 1 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt*1	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When enabling maskable interrupts, do not change the value of the NMIER.LVD1EN bit on the ICU side from the reset state.

8.2.2 Voltage Monitoring 1 Circuit Status Register (LVD1SR)

Address(es): 0008 00E1h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	LVD1MON	LVD1DET
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	LVD1DET	Voltage Monitoring 1 Voltage Change Detection Flag	0: Not detected 1: Vdet1 passage detection	R(/W) *1
b1	LVD1MON	Voltage Monitoring 1 Signal Monitor Flag	0: VCC < Vdet1 1: VCC ≥ Vdet1 or LVD1MON is disabled	R
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit. After writing 0 to this bit, it takes 2 system clock cycles for the bit to be read as 0.

LVD1DET Flag (Voltage Monitoring 1 Voltage Change Detection Flag)

The LVD1DET flag is enabled when the LVCMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.LVD1CMPE bit is 1 (voltage monitoring 1 circuit comparison result output enabled).
The LVD1DET flag should be set to 0 after LVD1CR0.LVD1RIE is set to 0 (disabled). LVD1CR0.LVD1RIE can be set to 1 (enabled) after a period of two or more cycles of PCLKB has elapsed.
Depending on the number of cycles of PCLKB defined for access to read an I/O register, two or more cycles than PCLKB may have to be secured as waiting time.

LVD1MON Flag (Voltage Monitoring 1 Signal Monitor Flag)

The LVD1MON flag is enabled when the LVCMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.LVD1CMPE bit is 1 (voltage monitoring 1 circuit comparison result output enabled).

8.2.3 Voltage Monitoring 2 Circuit Control Register 1 (LVD2CR1)

Address(es): 0008 00E2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	LVD2IRQSEL	LVD2IDTSEL [1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LVD2IDTSEL [1:0]	Voltage Monitoring 2 Interrupt Generation Condition Select	b1 b0 0 0: When VCC ≥ Vdet2 (rise) is detected 0 1: When VCC < Vdet2 (drop) is detected 1 0: When drop and rise are detected 1 1: Settings prohibited	R/W
b2	LVD2IRQSEL	Voltage Monitoring 2 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt*1	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When enabling maskable interrupts, do not change the value of the NMIER.LVD1EN bit on the ICU side from the reset state.

8.2.4 Voltage Monitoring 2 Circuit Status Register (LVD2SR)

Address(es): 0008 00E3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	LVD2MON	LVD2DET
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	LVD2DET	Voltage Monitoring 2 Voltage Change Detection Flag	0: Not detected 1: Vdet2 passage detection	R(/W) *1
b1	LVD2MON	Voltage Monitoring 2 Signal Monitor Flag	0: VCC < Vdet2 1: VCC ≥ Vdet2 or LVD2MON is disabled	R
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit. After writing 0 to this bit, it takes 2 system clock cycles for the bit to be read as 0.

LVD2DET Flag (Voltage Monitoring 2 Voltage Change Detection Flag)

The LVD2DET flag is enabled when the LVCMPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.LVD2CMPE bit is 1 (voltage monitoring 2 circuit comparison result output enabled).

The LVD2DET flag should be set to 0 after LVD2CR0.LVD2RIE bit is set to 0 (disabled). LVD2CR0.LVD2RIE bit can be set to 1 (enabled) after a period of two or more cycles of PCLKB has elapsed.

Depending on the number of cycles of PCLKB defined for access to read an I/O register, two or more cycles of PCLKB may have to be secured as waiting time.

LVD2MON Flag (Voltage Monitoring 2 Signal Monitor Flag)

The LVD2MON flag is enabled when the LVCMPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.LVD2CMPE bit is 1 (voltage monitoring 2 circuit comparison result output enabled).

8.2.5 Voltage Monitoring Circuit Control Register (LVCMPCR)

Address(es): 0008 C297h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	LVD2E	LVD1E	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	LVD1E	Voltage Detection 1 Enable*1	0: Voltage detection 1 circuit disabled 1: Voltage detection 1 circuit enabled	R/W
b6	LVD2E	Voltage Detection 2 Enable*2	0: Voltage detection 2 circuit disabled 1: Voltage detection 2 circuit enabled	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. The voltage of VCC = AVCC0 = AVCC1 when LVD1 is enabled must be set to at least 80 mV above the maximum value of the voltage detection 1 level selected by the LVDLVLR.LVD1LVL[3:0] bits.
Note 2. The voltage of VCC = AVCC0 = AVCC1 when LVD2 is enabled must be set to at least 80 mV above the maximum value of the voltage detection 2 level selected by the LVDLVLR.LVD2LVL[3:0] bits.

LVD1E Bit (Voltage Detection 1 Enable)

When using voltage detection 1 interrupt/reset or the LVD1SR.LVD1MON bit, set the LVD1E bit to 1. The voltage detection 1 circuit starts once td(E-A) passes after the LVD1E bit value is changed from 0 to 1. When using the voltage detection 1 circuit in deep software standby mode, do not set the DPSBYCR.DEEPCUT[1:0] bits to 11b.

LVD2E Bit (Voltage Detection 2 Enable)

When using voltage detection 2 interrupt/reset or the LVD2SR.LVD2MON bit, set the LVD2E bit to 1. The voltage detection 2 circuit starts once td(E-A) passes after the LVD2E bit value is changed from 0 to 1. When using the voltage detection 2 circuit in deep software standby mode, do not set the DPSBYCR.DEEPCUT[1:0] bits to 11b.

8.2.6 Voltage Detection Level Select Register (LVDLVLR)

Address(es): 0008 C298h

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD2LVL[3:0]				LVD1LVL[3:0]			
Value after reset:	1	0	1	1	1	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	LVD1LVL[3:0]	Voltage Detection 1 Level Select (Standard voltage during drop in voltage)	b3 b0 1 0 0 1: 2.99 V (Vdet1_1) 1 0 1 0: 2.92 V (Vdet1_2) 1 0 1 1: 2.85 V (Vdet1_3) Settings other than above are prohibited.	R/W
b7 to b4	LVD2LVL[3:0]	Voltage Detection 2 Level Select (Standard voltage during drop in voltage)	b7 b4 1 0 0 1: 2.99 V (Vdet1_1) 1 0 1 0: 2.92 V (Vdet1_2) 1 0 1 1: 2.85 V (Vdet1_3) Settings other than above are prohibited.	R/W

The contents of the LVDLVLR register can only be changed if the LVCMPCR.LVD1E and LVCMPCR.LVD2E bits (voltage detection n circuit disable; n = 1, 2) are both 0. The voltage detection circuits 1 and 2 should not be set at the same voltage detection level.

8.2.7 Voltage Monitoring 1 Circuit Control Register 0 (LVD1CR0)

Address(es): 0008 C29Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD1RN	LVD1RI	LVD1FSAMP[1:0]		—	LVD1CMPE	LVD1DFDIS	LVD1RIE
Value after reset:	1	0	0	0	x	0	1	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	LVD1RIE	Voltage Monitoring 1 Interrupt/Reset Enable	0: Disabled 1: Enabled	R/W
b1	LVD1DFDIS	Voltage Monitoring 1 Digital Filter Disable Mode Select	0: Digital filter enabled 1: Digital filter disabled	R/W
b2	LVD1CMPE	Voltage Monitoring 1 Circuit Comparison Result Output Enable	0: Voltage monitoring 1 circuit comparison result output disabled. 1: Voltage monitoring 1 circuit comparison result output enabled.	R/W
b3	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b5, b4	LVD1FSAMP[1:0]	Sampling Clock Select	b5 b4 0 0: 1/2 LOCO frequency 0 1: 1/4 LOCO frequency 1 0: 1/8 LOCO frequency 1 1: 1/16 LOCO frequency	R/W
b6	LVD1RI	Voltage Monitoring 1 Circuit Mode Select	0: Voltage monitoring 1 interrupt during Vdet1 passage 1: Voltage monitoring 1 reset enabled when the voltage falls to and below Vdet1	R/W
b7	LVD1RN	Voltage Monitoring 1 Reset Negate Select	0: Negation follows a stabilization time (tLVD1) after VCC > Vdet1 is detected. 1: Negation follows a stabilization time (tLVD1) after assertion of the LVD1 reset.	R/W

LVD1RIE Bit (Voltage Monitoring 1 Interrupt/Reset Enable)

Ensure that neither a voltage monitoring 1 reset nor a voltage monitoring 1 interrupt is generated during programming or erasure of the flash memory.

LVD1DFDIS Bit (Voltage Monitoring 1 Digital Filter Disable Mode Select)

Set the LOCOCR.LCSTP bit to 0 (the LOCO operates) if the LVD1DFDIS bit is 0 (enabling the digital filter circuit). Set the LVD1DFDIS bit to 1 (digital filter circuit disabled) when using voltage monitoring 1 circuit in software standby mode or deep software standby mode.

LVD1FSAMP [1:0] Bits (Sampling Clock Select)

The LVD1FSAMP[1:0] bits can be modified only when the LVD1DFDIS bit is 1 (digital filter circuit disabled). The LVD1FSAMP[1:0] bits should not be modified when the LVD1DFDIS bit is 0 (digital filter circuit enabled).

LVD1RI Bit (Voltage Monitoring 1 Circuit Mode Select)

When the LVD1RI bit is 1 (voltage monitoring 1 reset selected) or when the LVD2CR0.LVD2RI bit is 1 (voltage monitoring 2 reset selected), a transition to deep software standby mode cannot be made, instead a transition to software standby mode is made. To enter deep software standby mode, set the LVD1RI bit to 0 (voltage monitoring 1 interrupt selected) and the LVD2CR0.LVD2RI bit to 0 (voltage monitoring 2 interrupt selected).

LVD1RN Bit (Voltage Monitoring 1 Reset Negate Select)

If the LVD1RN bit is to be set to 1 (negation follows a stabilization time after assertion of the LVD1 reset signal), set the LOCOCR.LCSTP bit to 0 (the LOCO operates). Furthermore, if a transition to software standby or deep software standby is to be made, the only possible value for the LVD1RN bit is 0 (negation follows a stabilization time after VCC > Vdet1 is detected). Do not set the LVD1RN bit to 1 (negation follows a stabilization time after assertion of the LVD1 reset signal) when this is the case.

8.2.8 Voltage Monitoring 2 Circuit Control Register 0 (LVD2CR0)

Address(es): 0008 C29Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD2RN	LVD2RI	LVD2FSAMP[1:0]		—	LVD2CMPE	LVD2DFDIS	LVD2RIE
Value after reset:	1	0	0	0	x	0	1	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	LVD2RIE	Voltage Monitoring 2 Interrupt/Reset Enable	0: Disabled 1: Enabled	R/W
b1	LVD2DFDIS	Voltage Monitoring 2 Digital Filter Disable Mode Select	0: Digital filter enabled 1: Digital filter disabled	R/W
b2	LVD2CMPE	Voltage Monitoring 2 Circuit Comparison Result Output Enable	0: Voltage monitoring 2 circuit comparison result output disabled. 1: Voltage monitoring 2 circuit comparison result output enabled.	R/W
b3	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b5, b4	LVD2FSAMP[1:0]	Sampling Clock Select	b5 b4 0 0: 1/2 LOCO frequency 0 1: 1/4 LOCO frequency 1 0: 1/8 LOCO frequency 1 1: 1/16 LOCO frequency	R/W
b6	LVD2RI	Voltage Monitoring 2 Circuit Mode Select	0: Voltage monitoring 2 interrupt during Vdet2 passage 1: Voltage monitoring 2 reset enabled when the voltage falls to and below Vdet2	R/W
b7	LVD2RN	Voltage Monitoring 2 Reset Negate Select	0: Negation follows a stabilization time (tLVD2) after VCC > Vdet2 is detected. 1: Negation follows a stabilization time (tLVD2) after assertion of the LVD2 reset.	R/W

LVD2RIE Bit (Voltage Monitoring 2 Interrupt/Reset Enable)

Ensure that neither a voltage monitoring 2 reset nor a voltage monitoring 2 interrupt is generated during programming or erasure of the flash memory.

LVD2DFDIS Bit (Voltage Monitoring 2 Digital Filter Disable Mode Select)

Set the LOCOCR.LCSTP bit to 0 (the LOCO operates) if the LVD1DFDIS bit is 0 (enabling the digital filter circuit). Set the LVD2DFDIS bit to 1 (digital filter circuit disabled) when using voltage monitoring 2 circuit in software standby mode or deep software standby mode.

LVD2FSAMP[1:0] Bits (Sampling Clock Select)

The LVD2FSAMP[1:0] bits can be modified only when the LVD2DFDIS bit is 1 (digital filter circuit disabled). The LVD2FSAMP[1:0] bits should not be modified when the LVD2DFDIS bit is 0 (digital filter circuit enabled).

LVD2RI Bit (Voltage Monitoring 2 Circuit Mode Select)

When the LVD2RI it is 1 (voltage monitoring 2 reset selected) or when the LVD1CR0.LVD1RI bit is 1 (voltage monitoring 1 reset selected), a transition to deep software standby mode cannot be made, instead a transition to software standby mode is made. To enter to deep software standby mode, set the LVD2RI bit to 0 (voltage monitoring 2 interrupt selected) and the LVD1CR0.LVD1RI bit to 0 (voltage monitoring 1 interrupt selected).

LVD2RN Bit (Voltage Monitoring 2 Reset Negate Select)

If the LVD2RN bit is to be set to 1 (negation follows a stabilization time after assertion of the LVD2 reset signal), set the LOCOCR.LCSTP bit to 0 (the LOCO operates). Furthermore, if a transition to software standby or deep software standby is to be made, the only possible value for the LVD2RN bit is 0 (negation follows a stabilization time after VCC > Vdet2 is detected). Do not set the LVD2RN bit to 1 (negation follows a stabilization time after assertion of the LVD2 reset signal) when this is the case.

8.3 VCC Input Voltage Monitor

8.3.1 Monitoring Vdet0

Monitoring Vdet0 is not possible.

8.3.2 Monitoring Vdet1

Table 8.2 lists the procedures for setting up monitoring against Vdet1. After the settings are completed, results of comparison by voltage monitoring 1 can be monitored by using the LVD1SR.LVD1MON flag.

Table 8.2 Procedures for Setting up Monitoring against Vdet1

Step		Monitoring the Results of Comparison by Voltage Monitoring 1
Setting the voltage detection 1 circuit	1	Select the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits.
	2	Set LVCMPCR.LVD1E = 1 (enabling the voltage detection 1 circuit).
	3	Wait for at least td(E-A) (LVD operation stabilization time after LVD is enabled).*1
Setting the digital filter *2	4	Select the sampling clock for the digital filter by setting the LVD1CR0.LVD1FSAMP[1:0] bits.
	5	Set LVD1CR0.LVD1DFDIS = 0 (enabling the digital filter).
	6	Wait for at least 2n + 3 cycles of the LOCO (where n = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n).
Enabling output	7	Set LVD1CR0.LVD1CMPE = 1 (enabling output of the results of comparison by voltage monitoring 1).

Note 1. Steps 4 to 6 can be performed during the waiting time of step 3. For details of td(E-A), see section 64, Electrical Characteristics.
Note 2. Steps 4 to 6 are not required if the digital filter is not in use.

8.3.3 Monitoring Vdet2

Table 8.3 lists the procedures for setting up monitoring against Vdet2. After the settings are completed, results of comparison by voltage monitoring 2 can be monitored by using the LVD2SR.LVD2MON flag.

Table 8.3 Procedures for Setting up Monitoring against Vdet2

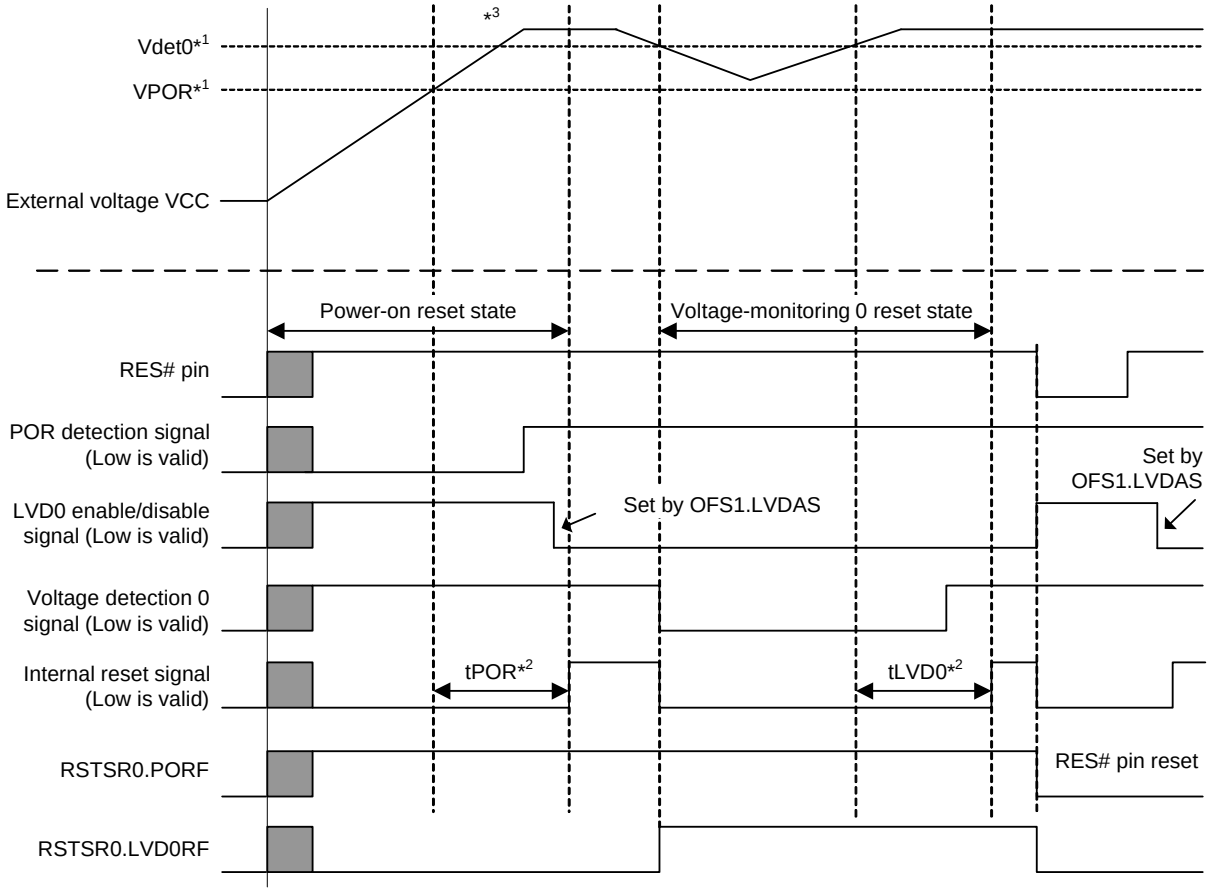
Step		Monitoring the Results of Comparison by Voltage Monitoring 2
Setting the voltage detection 2 circuit	1	Select the detection voltage by setting the LVDLVLR.LVD2LVL[3:0] bits.
	2	Set LVCMPCR.LVD2E = 1 (enabling the voltage detection 2 circuit).
	3	Wait for at least td(E-A) (LVD operation stabilization time after LVD is enabled).*1
Setting the digital filter *2	4	Select the sampling clock for the digital filter by setting the LVD2CR0.LVD2FSAMP[1:0] bits.
	5	Set LVD2CR0.LVD2DFDIS = 0 (enabling the digital filter).
	6	Wait for at least 2n + 3 cycles of the LOCO (where n = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n).
Enabling output	7	Set LVD2CR0.LVD2CMPE = 1 (enabling output of the results of comparison by voltage monitoring 2).

Note 1. Steps 4 to 6 can be performed during the waiting time of step 3. For details of td(E-A), see section 64, Electrical Characteristics.
Note 2. Steps 4 to 6 are not required if the digital filter is not in use.

8.4 Reset from Voltage Monitor 0

When using the reset from voltage monitor 0, clear the OFS1.LVDAS bit to 0 (enabling the voltage monitor 0 reset after a reset).

Figure 8.4 shows an example of operations for a voltage monitoring 0 reset.



Note: For details of the electrical characteristics, see section 64, Electrical Characteristics.
Note 1. VPOR indicates the detection level for a power-on reset and Vdet0 indicates the detection level for a voltage monitor 0 reset.
Note 2. tPOR indicates the period of a power-on reset and tLVD0 indicates the period of a voltage monitor 0 reset.
Note 3. At the time the power-supply voltage rises, VCC must rise to at least the minimum guaranteed voltage before release from the POR reset state.

Figure 8.4 Example of Voltage Monitoring 0 Reset Operation

8.5 Interrupt and Reset from Voltage Monitor 1

An interrupt or reset can be generated in response to the results of comparison by the voltage detection 1 circuit. Table 8.4 lists the procedures for setting bits related to the voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitoring operates. Table 8.5 shows the procedure for setting bits related to the voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitoring stops. Figure 8.5 shows an example of operations for a voltage monitor 1 interrupt. For the operation of the voltage monitor 1 reset, see Figure 6.2 in section 6, Resets. Furthermore, if you intend to use the voltage monitoring 1 circuit in software standby or deep software standby mode, make settings for the voltage monitoring 1 circuit according to the following procedures.

(1) Setting in software standby mode

- Disable the digital filter (LVD1DFDIS = 1).
- After VCC > Vdet1 is detected, negate the voltage monitoring 1 reset signal (LVD1RN = 0) following a stabilization time.

(2) Settings in deep software standby mode

- Disable the digital filter (LVD1DFDIS = 1).
- Enable voltage monitoring 1 interrupts (LVD1RI = 0). If the voltage monitoring 1 reset is enabled (LVD1RI = 1), a transition to deep software standby mode will not be possible, and the transition will be to software standby mode instead.
- When the DPSBYCR.DEEPCUT[1:0] bits are 11b, the voltage monitoring 1 circuit is stopped. If you intend to use the voltage monitoring 1 circuit in deep software standby mode, set the DPSBYCR.DEEPCUT[1:0] bits to a value other than 11b.

Table 8.4 Procedures for Setting Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset so that Voltage Monitoring Operates

Step		Voltage Monitoring 1 Interrupt (Voltage Monitoring 1 ELC Event Output)	Voltage Monitoring 1 Reset
Setting the voltage detection 1 circuit	1	Select the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits.	
	2	Set LVCMPCR.LVD1E = 1 (enabling the voltage detection 1 circuit).*4	
	3	Wait for at least td(E-A) (LVD operation stabilization time after LVD is enabled).*1	
Setting the digital filter *2	4	Select the sampling clock for the digital filter by setting the LVD1CR0.LVD1FSAMP[1:0] bits.	
	5	Set LVD1CR0.LVD1DFDIS = 0 (enabling the digital filter).	
	6	Wait for at least 2n + 3 cycles of the LOCO (where n = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n).	
Setting the voltage monitoring 1 interrupt or reset	7	Set LVD1CR0.LVD1RI = 0 (selecting the voltage monitoring 1 interrupt).	• Set LVD1CR0.LVD1RI = 1 (selecting the voltage monitoring 1 reset). • Select the type of the reset negation by setting the LVD1CR0.LVD1RN bit.
	8	• Select the timing of interrupt requests by setting the LVD1CR1.LVD1IDTSEL[1:0] bits. • Select the type of interrupt by setting the LVD1CR1.LVD1IRQSEL bit.	—
Enabling output	9	Set LVD1SR.LVD1DET = 0.	
	10	Set LVD1CR0.LVD1RIE = 1 (enabling the voltage monitoring 1 interrupt or reset).*3	
	11	Set LVD1CR0.LVD1CMPE = 1 (enabling output of the results of comparison by voltage monitoring 1).	

Note 1. Steps 4 to 10 can be performed during the waiting time of step 3. For details of td(E-A), see section 64, Electrical Characteristics.
Note 2. Steps 4 to 6 are not required if the digital filter is not in use.
Note 3. Step 10 is not required if only the ELC event signal is to be output.
Note 4. The voltage of VCC = AVCC0 = AVCC1 when LVD1 is enabled must be set to at least 80 mV above the maximum value of the voltage detection 1 level selected by the LVDLVLR.LVD1LVL[3:0] bits.

Table 8.5 Procedures for Setting Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset so that Voltage Monitoring Stops

Step		Voltage Monitoring 1 Interrupt (Voltage Monitoring 1 ELC Event Output), Voltage Monitoring 1 Reset
Settings to stop enabling of output	1	Set LVD1CR0.LVD1CMPE = 0 (disabling output of the results of comparison by voltage monitoring 1).
	2	Wait for at least 2n + 3 cycles of the LOCO (where n = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n).*1
	3	Set LVD1CR0.LVD1RIE = 0 (disabling the voltage monitoring 1 interrupt or reset).*2
Stopping the digital filter	4	Set LVD1CR0.LVD1DFDIS = 1 (disabling the digital filter).*1, *3
Stopping the voltage detection 1 circuit	5	Set LVCMPCR.LVD1E = 0 (disabling the voltage detection 1 circuit).

Note 1. Steps 2 and 4 are not required if the digital filter is not in use.
Note 2. Step 3 is not required if only the ELC event signal is to be output.
Note 3. To disable the digital filter from its enabled state and then re-enable it, disable it and wait for at least two cycles of the LOCO before re-enabling it.

If the voltage monitoring 1 interrupt or voltage monitoring 1 reset setting is to be made again after it has been used and stopped once, the following steps in the procedures for stopping and making the setting can be omitted according to the condition.

- Setting or stopping the voltage detection 1 circuit is not required if the setting for the voltage detection 1 circuit is not to be changed.
- Setting or stopping the digital filter is not required if the setting for the digital filter is not to be changed.
- Setting the voltage monitoring 1 interrupt or reset is not required if the setting for the voltage monitoring 1 interrupt or voltage monitoring 1 reset is not to be changed.

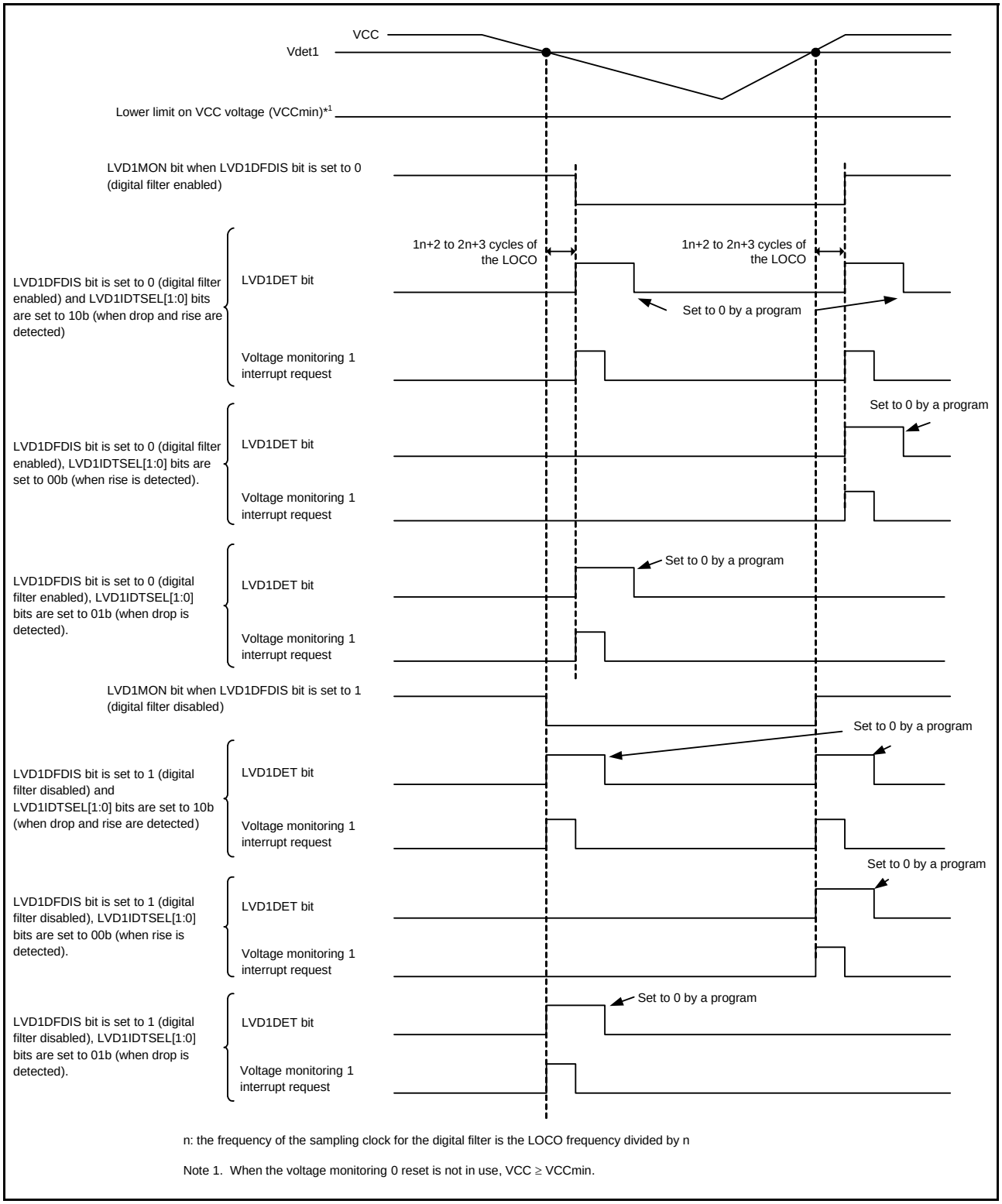


Figure 8.5 Example of Voltage Monitoring 1 Interrupt Operation

8.6 Interrupt and Reset from Voltage Monitor 2

An interrupt or reset can be generated in response to the results of comparison by the voltage detection 2 circuit. Table 8.6 shows the procedures for setting bits related to the voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitoring operates.Table 8.7 shows the procedure for setting bits related to the voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitoring stops. Figure 8.6 shows an example of operations for a voltage monitor 2 interrupt. For the operation of the voltage monitor 2 reset, see Figure 6.2 in section 6, Resets. Furthermore, if you intend to use the voltage monitoring 2 circuit in software standby or deep software standby mode, make settings for the voltage monitoring 2 circuit according to the following procedures.

(1) Setting in software standby mode

- Disable the digital filter (LVD2DFDIS = 1).
- After VCC > Vdet2 is detected, negate the voltage monitoring 2 reset signal (LVD2RN = 0) following a stabilization time.

(2) Settings in deep software standby mode

- Disable the digital filter (LVD2DFDIS = 1).
- Enable voltage monitoring 2 interrupts (LVD2RI = 0). If the voltage monitoring 2 reset is enabled (LVD2RI = 1), a transition to deep software standby mode will not be possible, and the transition will be to software standby mode instead.
- When the DPSBYCR.DEEPCUT[1:0] bits are 11b, the voltage monitoring 2 circuit is stopped. If you intend to use the voltage monitoring 2 circuit in deep software standby mode, set the DPSBYCR.DEEPCUT[1:0] bits to a value other than 11b.

Table 8.6 Procedures for Setting Bits Related to the Voltage Monitoring 2 Interrupt and Voltage Monitoring 2 Reset so that Voltage Monitoring Operates

Step		Voltage Monitoring 2 Interrupt (Voltage Monitoring 2 ELC Event Output)	Voltage Monitoring 2 Reset
Setting the voltage detection 2 circuit	1	Select the detection voltage by setting the LVDLVLR.LVD2LVL[3:0] bits.	
	2	Set LVCMPCR.LVD2E = 1 (enabling the voltage detection 2 circuit).*4	
	3	Wait for at least td(E-A) (LVD operation stabilization time after LVD is enabled).*1	
Setting the digital filter *2	4	Select the sampling clock for the digital filter by setting the LVD2CR0.LVD2FSAMP[1:0] bits.	
	5	Set LVD2CR0.LVD2DFDIS = 0 (enabling the digital filter).	
	6	Wait for at least 2n + 3 cycles of the LOCO (where n = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n).	
Setting the voltage monitoring 2 interrupt or reset	7	Set LVD2CR0.LVD2RI = 0 (selecting the voltage monitoring 2 interrupt).	• Set LVD2CR0.LVD2RI = 1 (selecting the voltage monitoring 2 reset). • Select the type of the reset negation by setting the LVD2CR0.LVD2RN bit.
	8	• Select the timing of interrupt requests by setting the LVD2CR1.LVD2IDTSEL[1:0] bits. • Select the type of interrupt by setting the LVD2CR1.LVD2IRQSEL bit.	—
Enabling output	9	Set LVD2SR.LVD2DET = 0.	
	10	Set LVD2CR0.LVD2RIE = 1 (enabling the voltage monitoring 2 interrupt or reset).*3	
	11	Set LVD2CR0.LVD2CMPE = 1 (enabling output of the results of comparison by voltage monitoring 2).	

Note 1. Steps 4 to 10 can be performed during the waiting time of step 3. For details of td(E-A), see section 64, Electrical Characteristics.
Note 2. Steps 4 to 6 are not required if the digital filter is not in use.
Note 3. Step 10 is not required if only the ELC event signal is to be output.
Note 4. The voltage of VCC = AVCC0 = AVCC1 when LVD2 is enabled must be set to at least 80 mV above the maximum value of the voltage detection 2 level selected by the LVDLVLR.LVD2LVL[3:0] bits.

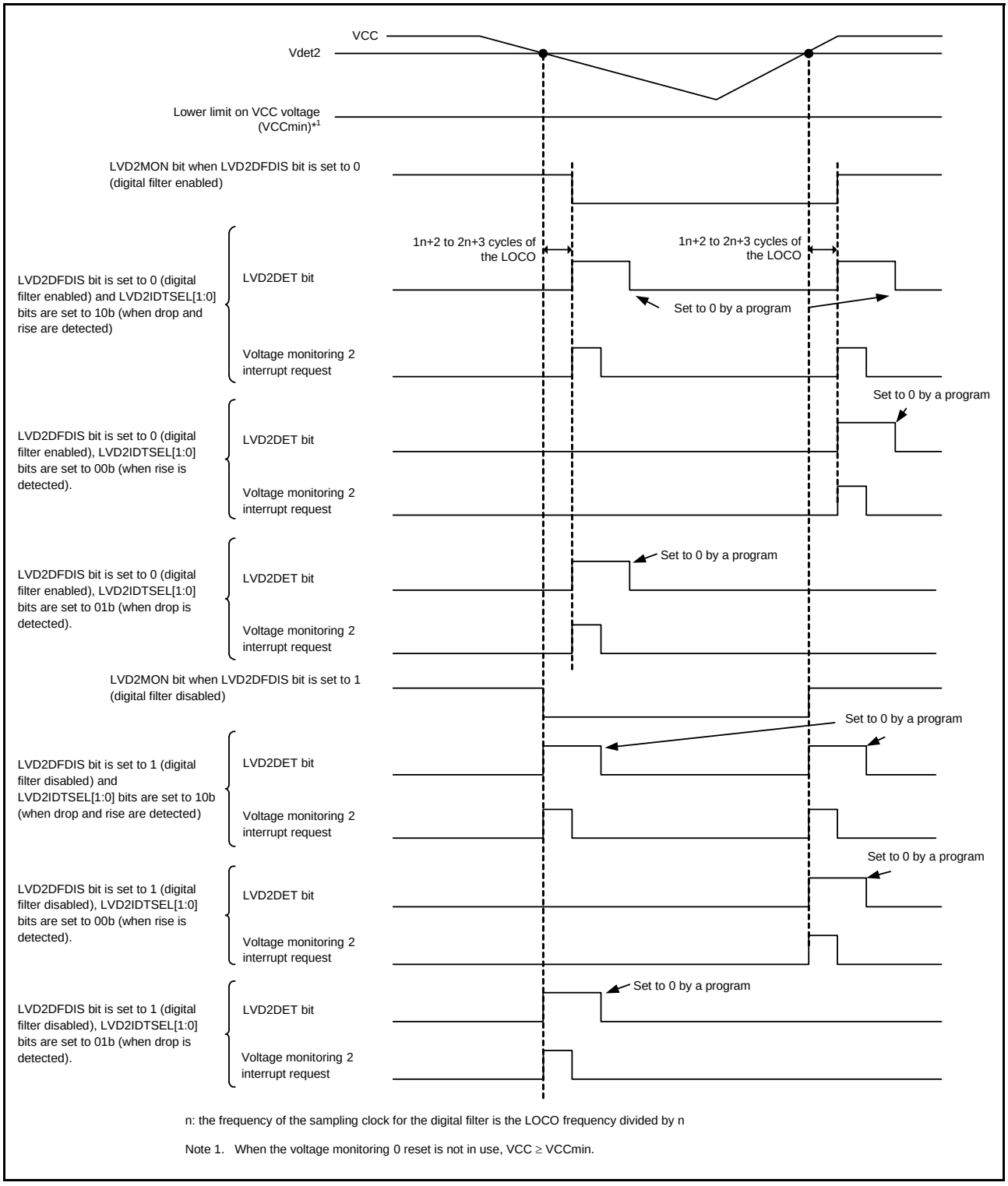
Table 8.7 Procedures for Setting Bits Related to the Voltage Monitoring 2 Interrupt and Voltage Monitoring 2 Reset so that Voltage Monitoring Stops

Step		Voltage Monitoring 2 Interrupt (Voltage Monitoring 2 ELC Event Output), Voltage Monitoring 2 Reset
Settings to stop enabling of output	1	Set LVD2CR0.LVD2CMPE = 0 (disabling output of the results of comparison by voltage monitoring 2).
	2	Wait for at least 2n + 3 cycles of the LOCO (where n = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n).*1
	3	Set LVD2CR0.LVD2RIE = 0 (disabling the voltage monitoring 2 interrupt or reset).*2
Stopping the digital filter	4	Set LVD2CR0.LVD2DFDIS = 1 (disabling the digital filter).*1, *3
Stopping the voltage detection 1 circuit	5	Set LVCMPCR.LVD2E = 0 (disabling the voltage detection 2 circuit).

Note 1. Steps 2 and 4 are not required if the digital filter is not in use.
Note 2. Step 3 is not required if only the ELC event signal is to be output.
Note 3. To disable the digital filter from its enabled state and then re-enable it, disable it and wait for at least two cycles of the LOCO before re-enabling it.

If the voltage monitoring 2 interrupt or voltage monitoring 2 reset setting is to be made again after it has been used and stopped once, the following steps in the procedures for stopping and making the setting can be omitted according to the condition.

- Setting or stopping the voltage detection 2 circuit is not required if the setting for the voltage detection 2 circuit is not to be changed.
- Setting or stopping the digital filter is not required if the setting for the digital filter is not to be changed.
- Setting the voltage monitoring 2 interrupt or reset is not required if the setting for the voltage monitoring 2 interrupt or voltage monitoring 2 reset is not to be changed.



n: the frequency of the sampling clock for the digital filter is the LOCO frequency divided by n

Note 1. When the voltage monitoring 0 reset is not in use, VCC ≥ VCCmin.

Figure 8.6 Example of Voltage Monitoring 2 Interrupt Operation

8.7 Event Link Output

The LVD can output the event signals to the event link controller (ELC).

(1) Vdet1 passage detection event

Thc LVD outputs the event signal when it is detected that the voltage has passed the Vdet1 voltage while both the voltage detection 1 circuit and the voltage monitoring l circuit comparison result output are enabled.

(2) Vdet2 passage detection event

The LVD outputs the event signal when it is detected that the voltage has passed the Vdet2 voltage while both the voltage detection 2 circuit and the voltage monitoring 2 circuit comparison result output are enabled.

When enabling the LVD's event link output function, be sure to make settings for enabling the LVD before enabling the LVD event link function of the ELC. To stop the LVD's event link output function, be sure to make settings for stopping the LVD before disabling the LVD event link function of the ELC.

8.7.1 Interrupt Handling and Event Linking

The LVD has the bits to separately enable or disable the voltage monitoring l and 2 interrupts. When an interrupt source is generated and the interrupt is enabled by the interrupt enable bit, the interrupt signal (LVD1RIE and LVD2RIE) is output to the CPU.
On the contrary, as soon as an interrupt source is generated, the event link signal is output as the event signal to the other module via the ELC regardless of the state of the interrupt enable bit.
It is possible to output voltage monitoring l and 2 interrupts in software standby and deep software standby modes. The event signals for the ELC in software standby and deep software standby modes, are output as follows:

- When the event Vdet1/Vdet2 are detected in software standby mode, no event signals are generated for the ELC because no clock is presented in software standby mode. Since Vdet1/Vdet2 passage detection flags are preserved, however, when the supply of the clock is resumed after restoring from software standby mode, the event signals for the ELC are output according to the state of the Vdet1/Vdet2 passage detection flags.
- If events of passing Vdet1/Vdet2 are detected in deep software standby mode, no event signals are generated for the ELC.

9. Clock Generation Circuit

9.1 Overview

This MCU incorporates a clock generation circuit.
Table 9.1 lists the specifications of the clock generation circuit. Figure 9.1 shows a block diagram of the clock generation circuit.

Table 9.1 Specifications of Clock Generation Circuit (1/2)

Item	Specification
Use	• Generates the system clock (ICLK) to be supplied to the CPU, DMAC, DTC, code flash memory, and RAM. • Generates the peripheral module clock (PCLKA) to be supplied to the ETHERC, EDMAC, EPTPC, USBA, RSPI, SCIF, MTU3, GPT, and AES.*1 • Generates the peripheral module clock (PCLKB) to be supplied to peripheral modules. • Generates the peripheral module clocks (for analog conversion) (PCLKC: unit 0; PCLKD: unit 1) to be supplied to S12AD. • Generates the flash-IF clock (FCLK) to be supplied to the flash interface. • Generates the external bus clock (BCLK) to be supplied to the external bus. • Generates the SDRAM clock (SDCLK) to be supplied to the SDRAM. • Generates the USB clock (UCLK) to be supplied to the USBb and the PHY in the USBA. • Generates the USBA clock (USBMCLK) to be supplied to the PHY in the USBA. • Generates the CAC clock (CACCLK) to be supplied to the CAC. • Generates the CAN clock (CANMCLK) to be supplied to the CAN. • Generates the RTC sub-clock (RTCSCLK) to be supplied to the RTC. • Generates the RTC main clock (RTCMCLK) to be supplied to the RTC. • Generates the IWDT-dedicated clock (IWDTCLK) to be supplied to the IWDT. • Generates the JTAG clock (JTAGTCK) to be supplied to the JTAG.
Operating frequency*2	• ICLK: 120 MHz (max) • PCLKA: 120 MHz (max) • PCLKB: 60 MHz (max) • PCLKC: 60 MHz (max) • PCLKD: 60 MHz (max) • FCLK: 4 MHz to 60 MHz (for programming and erasing the code flash memory and data flash memory) 60 MHz (max) (for reading from the data flash memory) • BCLK: 120 MHz (max) • BCLK pin output: 60 MHz (max) • SDCLK pin output: 60 MHz (max) • UCLK: 48 MHz (max) • USBMCLK: 20 MHz, 24 MHz • CACCLK: Same as the clock from respective oscillators. • CANMCLK: 24 MHz (max) • RTCSCLK: 32.768 kHz • RTCMCLK: 8 MHz to 16 MHz • IWDTCLK: 120 kHz • JTAGTCK: 10 MHz (max)
Main clock oscillator	• Resonator frequency: 8 MHz to 24 MHz • External clock input frequency: 24 MHz (max) • Connectable resonator or additional circuit: ceramic resonator, crystal resonator • Connection pin: EXTAL, XTAL • Oscillation stop detection function: When an oscillation stop is detected with the main clock, the system clock source is switched to LOCO, and MTU3 and GPT output can be forcedly driven to the high-impedance.
Sub-clock oscillator	• Resonator frequency: 32.768 kHz • Connectable resonator or additional circuit: crystal resonator • Connection pin: XCIN, XCOUT
PLL frequency synthesizer	• Input clock source: Main clock, HOCO • Input pulse frequency division ratio: Selectable from 1, 2, and 3 • Input frequency: 8 MHz to 24 MHz • Frequency multiplication ratio: Selectable from 10 to 30 • Output clock frequency of the PLL frequency synthesizer: 120 MHz to 240 MHz

Table 9.1 Specifications of Clock Generation Circuit (2/2)

Item	Specification
High-speed on-chip oscillator (HOCO)	• Selectable from 16 MHz, 18 MHz, and 20 MHz • HOCO power supply control
Low-speed on-chip oscillator (LOCO)	Oscillation frequency: 240 kHz
IWDT-dedicated on-chip oscillator	Oscillation frequency: 120 kHz
JTAG external clock input (TCK)	Input clock frequency: 10 MHz (max)
Control of output on the BCLK pin	• BCLK clock output or high output is selectable • BCLK or BCLK/2 is selectable
Control of output on the SDCLK pin	SDCLK clock output or high output is selectable
Event linking (output)	Detection of stopping of the main clock oscillator
Event linking (input)	Switching of the clock source to the low-speed on-chip oscillator

Note 1. Restrictions in relation to the clock when ETHERC is in use are as follows.
12.5 MHz ≤ PCLKA ≤ 120 MHz

Note 2. Restrictions on setting clock frequency: ICLK ≥ BCLK, PCLKA ≥ PCLKB, PCLKB ≥ PCLKC, PCLKB ≥ PCLKD
Restrictions on clock frequency ratio: (N: integer)
ICLK:FCLK = N:1 or 1:N; ICLK:PCLKA = N:1 or 1:N; ICLK:PCLKB = N:1 or 1:N;
ICLK:PCLKC = N:1 or 1:N; ICLK:PCLKD = N:1 or 1:N

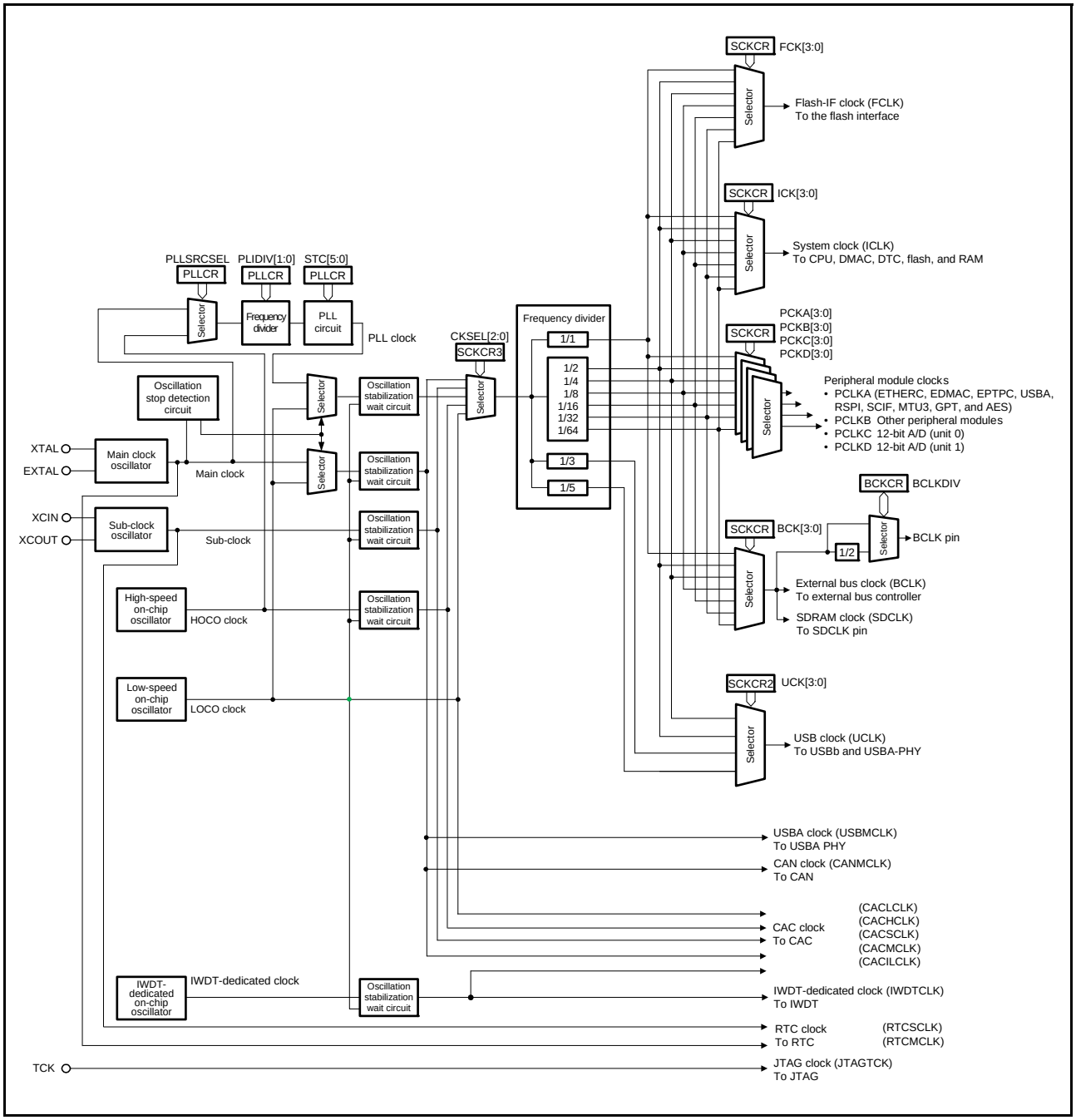


Figure 9.1 Block Diagram of Clock Generation Circuit

Table 9.2 lists the input/output pins of the clock generation circuit.

Table 9.2 Input/Output Pins of Clock Generation Circuit

Pin Name	I/O	Description
XTAL	Output	These pins are used to connect a crystal resonator. The EXTAL pin can also be used to input an external clock. For details, section 9.3.2, External Clock Input.
EXTAL	Input	
XCIN	Input	These pins are used to connect a 32.768-kHz crystal resonator.
XCOUT	Output	
TCK	Input	This pin is used to input the clock for the JTAG.
BCLK	Output	This pin is used to supply external devices with the external bus clock (BCLK).
SDCLK	Output	This pin is used to supply external devices with the SDRAM clock (SDCLK).

9.2 Register Descriptions

9.2.1 System Clock Control Register (SCKCR)

Address(es): 0008 0020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FCK[3:0]				ICK[3:0]				PSTOP1	PSTOP0	—	—	BCK[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PCKA[3:0]				PCKB[3:0]				PCKC[3:0]				PCKD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PCKD[3:0]	Peripheral Module Clock D (PCLKD) Select*1	b3 b0 0 0 0 0: ×1/1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W
b7 to b4	PCKC[3:0]	Peripheral Module Clock C (PCLKC) Select*1	b7 b4 0 0 0 0: ×1/1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W
b11 to b8	PCKB[3:0]	Peripheral Module Clock B (PCLKB) Select*1	b11 b8 0 0 0 0: ×1/1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W
b15 to b12	PCKA[3:0]	Peripheral Module Clock A (PCLKA) Select*1	b15 b12 0 0 0 0: ×1/1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W
b19 to b16	BCK[3:0]	External Bus Clock (BCLK) Select*1, *2	b19 b16 0 0 0 0: ×1/1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W
b21, b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b22	PSTOP0	SDCLK Pin Output Control	0: SDCLK pin output is enabled. 1: SDCLK pin output is disabled. (Fixed high)	
b23	PSTOP1	BCLK Pin Output Control*3	0: BCLK pin output is enabled. 1: BCLK pin output is disabled. (Fixed high)	R/W
b27 to b24	ICK[3:0]	System Clock (ICLK) Select*1, *2, *4	b27 b24 0 0 0 0: ×1/1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W
b31 to b28	FCK[3:0]	Flash-IF Clock (FCLK) Select*1, *4	b31 b28 0 0 0 0: ×1/1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W

Note 1. The setting for division by one is prohibited if the frequency of the clock signal from the PLL circuit is higher than 120 MHz while the SCKCR3.CKSEL[2:0] bits are selecting the PLL.
Note 2. Do not make a setting such that the ICLK runs at a lower frequency than the external bus clock.
Note 3. When operation of the external bus clock is selected, the P53 I/O port pin function is not available because it is multiplexed on the same pin as the BCLK pin function.
Note 4. When the SCKCR3.CKSEL[2:0] bits are selecting the sub-clock oscillator in low-speed operating mode 2, division by one is the only frequency division setting allowed for the ICLK and FCLK.

SCKCR should not be modified in the following cases:

- The operating power control mode transition status flag in the operating power control register (OPCCR.OPCMTSF) is 1 (a transition to operating power control mode in progress)
- The code flash memory P/E mode entry bit or the data flash memory P/E mode entry bit in the flash P/E mode entry register (FENTRYR.FENTRYC or FENTRYR.FENTRYD) is 1 (P/E mode)
- Time period from WAIT instruction execution for a transition to sleep mode, to return from sleep mode to normal operating mode

9.2.2 System Clock Control Register 2 (SCKCR2)

Address(es): 0008 0024h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	UCK[3:0]				—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b4	UCK[3:0]	USB Clock (UCLK) Select *1	b7 b4 0 0 0 1: ×1/2 0 0 1 0: ×1/3 0 0 1 1: ×1/4 0 1 0 0: ×1/5 Settings other than above are prohibited when USB is in use. When USB is not in use, these bits are read as 0001b. The write value should be 0001b.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not select division by two when the SCKCR3.CKSEL[2:0] bits are 010b (selecting the main clock oscillator).

SCKCR2 should not be modified in the following cases:

- The operating power control mode transition status flag in the operating power control register (OPCCR.OPCMTSF) is 1 (a transition to operating power control mode in progress)
- The code flash memory P/E mode entry bit or the data flash memory P/E mode entry bit in the flash P/E mode entry register (FENTRYR.FENTRYC or FENTRYR.FENTRYD) is 1 (P/E mode)
- Time period from WAIT instruction execution for a transition to sleep mode, to return from sleep mode to normal operating mode

UCK[3:0] Bits (USB Clock (UCLK) Select)

These bits select the frequency of the USB clock (UCLK).

The duty ratio is 2:1 when × 1/3 is selected while it is 3:2 when × 1/5 is selected.

9.2.3 System Clock Control Register 3 (SCKCR3)

Address(es): 0008 0026h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CKSEL[2:0]			—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	CKSEL[2:0]	Clock Source Select	b10 b8 0 0 0: LOCO 0 0 1: HOCO 0 1 0: Main clock oscillator 0 1 1: Sub-clock oscillator 1 0 0: PLL circuit Settings other than above are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SCKCR3 should not be modified in the following cases:

- The operating power control mode transition status flag in the operating power control register (OPCCR.OPCMTSF) is 1 (a transition to operating power control mode in progress)
- The code flash memory P/E mode entry bit or the data flash memory P/E mode entry bit in the flash P/E mode entry register (FENTRYR.FENTRYC or FENTRYR.FENTRYD) is 1 (P/E mode)
- Time period from WAIT instruction execution for a transition to sleep mode, to return from sleep mode to normal operating mode

CKSEL[2:0] Bits (Clock Source Select)

These bits select the source of the system clock (ICLK), peripheral module clocks (PCLKA, PCLKB, PCLKC, and PCLKD), flash-IF clock (FCLK), external bus clock (BCLK), SDRAM clock (SDCLK), and USB clock (UCLK) from low-speed on-chip oscillator (LOCO), high-speed on-chip oscillator (HOCO), the main clock oscillator, the sub-clock oscillator, and the PLL circuit.

Transitions to clock sources which are not in operation are prohibited.

9.2.4 PLL Control Register (PLLCR)

Address(es): 0008 0028h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	STC[5:0]						—	—	—	PLLSRCSEL	—	—	PLIDIV[1:0]	
Value after reset:	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0

Set this bit to 0 when using the USB.

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PLIDIV[1:0]	PLL Input Frequency Division Ratio Select	b1 b0 0 0: ×1 0 1: ×1/2 1 0: ×1/3 1 1: Setting prohibited	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	PLLSRCSEL	PLL Clock Source Select *1	0: Main clock oscillator 1: HOCO	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	STC[5:0]	Frequency Multiplication Factor Select	b13 b8 0 1 0 0 1 1: ×10.0 0 1 0 1 0 0: ×10.5 0 1 0 1 0 1: ×11.0 0 1 0 1 1 0: ×11.5 0 1 0 1 1 1: ×12.0 0 1 1 0 0 0: ×12.5 0 1 1 0 0 1: ×13.0 0 1 1 0 1 0: ×13.5 0 1 1 0 1 1: ×14.0 0 1 1 1 0 0: ×14.5 0 1 1 1 0 1: ×15.0 0 1 1 1 1 0: ×15.5 0 1 1 1 1 1: ×16.0 1 0 0 0 0 0: ×16.5 b13 b8 1 0 0 0 0 1: ×17.0 1 0 0 0 1 0: ×17.5 1 0 0 0 1 1: ×18.0 1 0 0 1 0 0: ×18.5 1 0 0 1 0 1: ×19.0 1 0 0 1 1 0: ×19.5 1 0 0 1 1 1: ×20.0 1 0 1 0 0 0: ×20.5 1 0 1 0 0 1: ×21.0 1 0 1 0 1 0: ×21.5 1 0 1 0 1 1: ×22.0 1 0 1 1 0 0: ×22.5 1 0 1 1 0 1: ×23.0 1 0 1 1 1 0: ×23.5 b13 b8 1 0 1 1 1 1: ×24.0 1 1 0 0 0 0: ×24.5 1 1 0 0 0 1: ×25.0 1 1 0 0 1 0: ×25.5 1 1 0 0 1 1: ×26.0 1 1 0 1 0 0: ×26.5 1 1 0 1 0 1: ×27.0 1 1 0 1 1 0: ×27.5 1 1 0 1 1 1: ×28.0 1 1 1 0 0 0: ×28.5 1 1 1 0 0 1: ×29.0 1 1 1 0 1 0: ×29.5 1 1 1 0 1 1: ×30.0 Settings other than above are prohibited.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Set this bit to 0 when using the USB.

Writing to the PLLCR is prohibited when the PLLCR2.PLLEN bit is 0 (the PLL operates).

PLIDIV[1:0] Bits (PLL Input Frequency Division Ratio Select)

These bits select the frequency division ratio of the PLL clock source.

Set these bits so that the frequency of PLL input signal is within the range of 8 MHz to 24 MHz.

PLLSRCSEL Bit (PLL Clock Source Select)

This bit selects the clock source for the PLL.

STC[5:0] Bits (Frequency Multiplication Factor Select)

These bits select the frequency multiplication factor of the PLL circuit.

Set these bits so that the output frequency is within the range of the output clock frequency of the PLL circuit (120 MHz to 240 MHz).

9.2.5 PLL Control Register 2 (PLLCR2)

Address(es): 0008 002Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PLLEN
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	PLLEN	PLL Stop Control	0: PLL is operating. 1: PLL is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PLLEN Bit (PLL Stop Control)

This bit runs or stops the PLL circuit.
The PLL clock source is selectable as the main clock oscillator and HOCO.
Selecting the main clock oscillator as the PLL clock source with the PLLCR.PLLSRCSEL bit requires setting the main clock oscillator wait control register (MOSCWTCR).
After the setting of the PLLEN bit has been changed to make the PLL run, only start using the PLL clock after confirming that the OSCOVFSR.PLOVF flag has been set to 1.
That is, a fixed time for stabilization is required after the setting for PLL operation. A fixed time is also required for oscillation to stop after the setting to stop PLL operation. Accordingly, take note of the following limitations when starting and stopping PLL operation by the PLLEN bit. The following notes apply when selecting the main clock oscillator as the PLL clock source.

- Setting PLL operation with the PLLEN bit is possible regardless of the setting of the OSCOVFSR.PLOVF flag. However, the time until deactivation of the PLL is completed (the time until the PLOVF flag is set to 0 after the setting to stop PLL operation) means that writing to the PLLCR2 register takes longer than the setting to operate the PLL.
- The PLL can be stopped by the PLLEN bit regardless of the setting of the OSCOVFSR.PLOVF flag. However, the wait for oscillation to become stable, i.e. the oscillation settling time (the time until the OSCOVFSR.PLOVF flag is set to 1 after the setting to operate the PLL), means that writing to the PLLCR2 register takes longer than the setting to stop PLL operation.
- Regardless of whether or not the PLL clock is selected as the system clock, confirm that the OSCOVFSR.PLOVF flag has been set to 1 before executing a WAIT instruction to place the chip to software standby or deep software standby after the setting to operate the PLL.
- When a transition to software standby or deep software standby is to follow the setting to stop the PLL, confirm that the OSCOVFSR.PLOVF flag has been set to 0 before executing the WAIT instruction.

Writing of 1 to the PLLEN bit (stopping the PLL) is prohibited while the PLL clock is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).
Writing of 0 to the PLLEN bit (making the PLL operate) is prohibited when the setting of the operating power control mode select bits in the operating power control register (OPCCR.OPCM[2:0]) is for low-speed operating mode 1 or 2.

9.2.6 External Bus Clock Control Register (BCKCR)

Address(es): 0008 0030h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	BCLKDIV
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BCLKDIV	BCLK Pin Output Select	0: BCLK 1: 1/2 BCLK	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

BCKCR should not be modified in the following cases:

- The operating power control mode transition status flag in the operating power control register (OPCCR.OPCMTSF) is 1 (a transition to operating power control mode in progress)
- The code flash memory P/E mode entry bit or the data flash memory P/E mode entry bit in the flash P/E mode entry register (FENTRYR.FENTRYC or FENTRYR.FENTRYD) is 1 (P/E mode).
- Time period from WAIT instruction execution for a transition to sleep mode, to return from sleep mode to normal operating mode

BCLKDIV Bit (BCLK Pin Output Select)

This bit selects the clock signal for output from the BCLK pin.
Either the BCLK clock with the frequency selected by the BCK[3:0] bits in SCKCR or the BCLK clock divided by 2 can be selected. To control external bus control signals at the falling edge of the BCLK pin, set this bit to 1.

9.2.7 Main Clock Oscillator Control Register (MOSCCR)

Address(es): 0008 0032h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MOSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	MOSTP	Main Clock Oscillator Stop	0: Main clock oscillator is operating. 1: Main clock oscillator is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

MOSTP Bit (Main Clock Oscillator Stop)

This bit runs or stops the main clock oscillator.

The main clock oscillator is operated or stopped by the MOSTP bit and main clock oscillator forced oscillation bit in the main clock oscillator forced oscillation control register (MOFCR.MOFXIN). The main clock oscillator can be started by setting the MOSTP bit to operating or by setting the MOFXIN bit to forced oscillation. When the MOFXIN bit is set to forced oscillation, the oscillator operates even in software standby mode and deep software standby mode.

When the main clock is to be used as the source to drive counting by the realtime clock, use the MOFCR.MOFXIN bit to make the main clock run. If the main clock is only to be used as the system clock or PLL clock source, use the MOSCCR.MOSTP bit.

When the main clock is to be used as the clock source for counting and also for another purpose, use both the MOFCR.MOFXIN and MOSCCR.MOSTP bits to make the main clock run.

When changing the value of the MOSTP bit or MOFCR.MOFXIN bit, execute subsequent instructions after reading the bit and checking that its value has actually been updated (refer to (2) Notes on writing to I/O registers, in section 5, I/O Registers).

When the main clock is not to be used as the clock source for counting by the realtime clock, the main clock oscillator wait control register (MOSCWTCR) must be set. In this case, after the setting of the MOSCCR.MOSTP bit has been changed to make the main clock run, only start using the main clock after confirming that the OSCOVFSR.MOOVF flag has been set to 1.

When the main clock is to be used as the clock source for counting by the realtime clock, after the setting of MOFCR.MOFXIN bit has been changed to make the main clock run, only start using the main clock as the clock source for counting by the realtime clock after software has checked that the main clock oscillation stabilization time (recommended) (tMAINOSC) has elapsed. When an external clock signal is to be input for supply to the main clock oscillator, waiting for the main clock oscillation stabilization time (tMAINOSC) is not necessary.

For the main clock oscillator, a fixed time is required for oscillation to become stable after the settings for operation have been made. Furthermore, a fixed time is required for oscillation to actually stop after the settings to stop oscillation have been made. Accordingly, take note of the following limitations when starting and stopping operation.

- Main-clock operation can be selected with the MOSTP bit regardless of the setting of the OSCOVFSR.MOOVF flag. However, the time until deactivation of the main clock is completed (the time until the OSCOVFSR.MOOVF flag is set to 0 after the setting to stop operation) means that writing to the MOSCCR register takes longer than the setting to operate the main clock.
- The main clock can be stopped by the MOSTP bit regardless of the setting of the OSCOVFSR.MOOVF flag. However, the wait for oscillation to become stable, i.e. the oscillation settling time (the time until the OSCOVFSR.MOOVF flag is set to 1 after the setting to operate the main clock), means that writing to the MOSCCR register takes longer than the setting to stop operation.

- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.MOOVF flag has been set to 1 before executing a WAIT instruction to place the chip to software standby or deep software standby after the setting to operate the main clock oscillator by the MOSTP bit.
- When a transition to software standby or deep software standby is to follow the setting to stop the main clock oscillator, confirm that the OSCOVFSR.MOOVF flag has been set to 0 before executing the WAIT instruction.

Writing of 1 to the MOSTP bit (stopping the main clock oscillator) is prohibited in either of the following cases:

- The main clock oscillator is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).
- 0 (PLL operation) is selected by the PLL stop control bit in PLL control register 2 (PLLCR2.PLLEN), and the main clock oscillator is selected by the PLL clock source select bit in the PLL control register (PLLCR.PLLSRCSEL).

9.2.8 Sub-Clock Oscillator Control Register (SOSCCR)

Address(es): 0008 0033h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SOSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SOSTP	Sub-Clock Oscillator Stop	0: Sub-clock oscillator is operating. 1: Sub-clock oscillator is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SOSTP Bit (Sub-Clock Oscillator Stop)

This bit runs or stops the sub-clock oscillator.

The SOSTP bit and the sub-clock control bit in the RTC control register 3 (RCR3.RTCEN) controls whether to operate or stop the sub-clock oscillator. If one of these bits is set so as to enable the operation, the sub-clock oscillator runs.

When changing the value of the SOSTP bit or RCR3.RTCEN bit, execute subsequent instructions after reading the bit and checking that its value has actually been updated (refer to (2) Notes on writing to I/O registers, in section 5, I/O Registers).

When the sub-clock is to be used as the source to drive counting by the realtime clock, use the RCR3.RTCEN and SOSTP bits. When the sub-clock is not to be used as the clock source for counting by the realtime clock but is to be used as the system clock, use the SOSCCR.SOSTP bit.

When the SOSTP bit is set to operate the sub-clock, be sure to set the sub-clock oscillator wait control register (SOSCWTCR) beforehand. Furthermore, after selecting sub-clock operation, only start using the sub-clock after confirming that the OSCOVFSR.SOOVF flag has been set to 1.

A fixed time for stabilization is required for oscillation to become stable after selecting sub-clock operation with the SOSTP bit. Furthermore, a fixed time is required for oscillation to actually stop after the setting to stop operation. Accordingly, take note of the following when starting and stopping operation with the SOSTP bit.

- Sub-clock operation can be selected with the SOSTP bit regardless of the setting of the OSCOVFSR.SOOVF flag. However, the time until deactivation of the sub-clock is completed (the time until the OSCOVFSR.SOOVF flag is set to 0 after the setting to stop operation) means that writing to the SOSCCR register takes longer than the setting to operate the sub-clock.
- The sub-clock can be stopped by the SOSTP bit regardless of the setting of the OSCOVFSR.SOOVF flag. However, the wait for oscillation to become stable, i.e. the oscillation settling time (the time until the OSCOVFSR.SOOVF flag is set to 1 after the setting to operate the sub-clock), means that writing to the SOSCCR register takes longer than the setting to stop operation.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.SOOVF flag has been set to 1 before executing a WAIT instruction to place the chip on software standby or deep software standby after the setting to operate the sub-clock oscillator by the SOSTP bit.
- When a transition to software standby or deep software standby is to follow the setting to stop the sub-clock oscillator, confirm that the OSCOVFSR.SOOVF flag has been set to 0 after the setting to stop the sub-clock oscillator and before executing the WAIT instruction.

Writing of 1 to the SOSTP bit (stopping the sub-clock oscillator) is prohibited while the sub-clock oscillator is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).

9.2.9 Low-Speed On-Chip Oscillator Control Register (LOCOCR)

Address(es): 0008 0034h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LCSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	LCSTP	LOCO Stop	0: LOCO is operating. 1: LOCO is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

LCSTP Bit (LOCO Stop)

This bit runs or stops the LOCO.

After the setting of the LCSTP bit has been changed so that the LOCO operates, only start using the LOCO after the LOCO clock oscillation stabilization waiting time (tLOCOWT) has elapsed.

That is, a fixed time for stabilization of oscillation is required for oscillation to become stable after setting LOCO operation with the LCSTP bit. A fixed time is also required for oscillation to stop after the setting to stop the oscillator. Accordingly, take note of the following limitations when starting and stopping the oscillator.

- When restarting the LOCO after it has been stopped, allow at least five cycles of the LOCO as an interval over which it is still stopped.
- Ensure that oscillation by the LOCO is stable when making the setting to stop the LOCO.
- Regardless of whether or not it is selected as the system clock, ensure that oscillation by the LOCO is stable before executing a WAIT instruction to place the chip on software standby or deep software standby.
- When a transition to software standby or deep software standby is to follow the setting to stop the LOCO, wait for at least three cycles of the LOCO after the setting to stop the LOCO and before executing the WAIT instruction.

Writing of 1 to the LCSTP bit (stopping the LOCO) is prohibited while the LOCO is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).

Writing of 1 to the LCSTP bit (stopping the LOCO) is prohibited if detection of oscillation stopping is enabled by the oscillation stop detection enable bit in the oscillation stop detection control register (OSTDCR.OSTDE).

Since the LOCO clock is used to measure the waiting time for other oscillators, the LOCO clock oscillates while the waiting time for other oscillators is being measured, regardless of the setting of LCSTP bit. Therefore, the LOCO clock may be unintentionally supplied even if the LCSTP bit is set to be stopped.

9.2.10 IWDT-Dedicated On-Chip Oscillator Control Register (ILOCOCR)

Address(es): 0008 0035h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ILCSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	ILCSTP	IWDT-Dedicated On-Chip Oscillator Stop	0: IWDT-dedicated on-chip oscillator is operating. 1: IWDT-dedicated on-chip oscillator is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

When the IWDT start mode select bit in the option function select register 0 (OFS0.IWDTSTRT) is 0 (IWDT operating), the setting of this register is invalid; it is valid only when the OFS0.IWDTSTRT bit is set to 1 (IWDT stopped). The ILCSTP bit cannot be changed from 0 (IWDT-dedicated on-chip oscillator operating) to 1 (IWDT-dedicated on-chip oscillator stopped) while ILOCOCR is valid.

ILCSTP Bit (IWDT-Dedicated On-Chip Oscillator Stop)

This bit runs or stops the IWDT-dedicated on-chip oscillator.

After the setting of the ILCSTP bit has been changed to make the IWDT-dedicated on-chip oscillator run, confirm that the OSCOVFSR.ILCOVF flag has been set to 1 before starting to use the oscillator.

When a transition to software standby or deep software standby mode is to follow the setting to start the IWDT-dedicated on-chip oscillator, confirm that the OSCOVFSR.ILCOVF flag has been set to 1 before executing the WAIT instruction.

9.2.11 High-Speed On-Chip Oscillator Control Register (HOCOCR)

Address(es): 0008 0036h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	HCSTP
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Bit Name	Description	R/W
b0	HCSTP	HOCO Stop	0: HOCO is operating. 1: HOCO is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The HCSTP bit value after a reset is 0 when the OFS1.HOCOEN bit is 0. It is 1 when the OFS1.HOCOEN bit is 1.

HCSTP Bit (HOCO Stop)

This bit runs or stops the HOCO.
After the setting of the HCSTP bit has been changed to make the HOCO run, confirm that the OSCOVFSR.HCOVF flag has been set to 1 before starting to use the oscillator.
A fixed time for stabilization is required for oscillation to become stable after setting HOCO operation with the HCSTP bit. Furthermore, a fixed time is required for oscillation to actually stop after the setting to stop operation. Accordingly, take note of the following when starting and stopping operation with the HCSTP bit.

- Setting HOCO operation with the HCSTP bit is possible regardless of the setting of the OSCOVFSR.HCOVF flag. However, the time until deactivation of the HOCO is completed (the time until the OSCOVFSR.HCOVF flag is set to 0 after the setting to stop HOCO operation) means that writing to the HOCOCR register takes longer than the setting to operate the HOCO.
- The HOCO can be stopped by the HCSTP bit regardless of the setting of the OSCOVFSR.HCOVF flag. However, the wait for oscillation to become stable, i.e. the oscillation settling time (the time until the OSCOVFSR.HCOVF flag is set to 1 after the setting to operate the HOCO), means that writing to the HOCOCR register takes longer than the setting to stop HOCO operation.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.HCOVF flag has been set to 1 before executing a WAIT instruction to place the chip on software standby or deep software standby after selecting HOCO operation with the HCSTP bit.
- When a transition to software standby or deep software standby is to follow the setting to stop the HOCO, confirm that the OSCOVFSR.HCOVF flag has been set to 0 after the setting to stop the HOCO and before executing the WAIT instruction.

Writing of 1 to the HCSTP bit (stopping the HOCO) is prohibited while the HOCO is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]), or the HOCO is selected as the clock source for the PLL by the PLLCR.PLLSRCSEL bit, and the PLL is selected by the SCKCR3.CKSEL[2:0] bits.
Writing of 0 to the HCSTP bit (making the HOCO operate) is prohibited when the setting of the operating power control mode select bits in the operating power control register (OPCCR.OPCM[2:0]) is for low-speed operating mode 2.

9.2.12 High-Speed On-Chip Oscillator Control Register 2 (HOCOCR2)

Address(es): 0008 0037h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	HCFRQ[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	HCFRQ[1:0]	HOCO Frequency Setting	b1 b0 0 0: 16 MHz 0 1: 18 MHz 1 0: 20 MHz Settings other than above are prohibited.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Writing to the HOCOCR2 register is prohibited when the HOCOCR.HCSTP bit is 0 (making the HOCO run).

9.2.13 Oscillation Stabilization Flag Register (OSCOVFSR)

Address(es): 0008 003Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ILCOVF	HCOVF	PLOVF	SOOVF	MOOVF
Value after reset:	0	0	0	0/1*1	0/1*2	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MOOVF	Main Clock Oscillation Stabilization Flag	0: MOSTP = 1 (stopping the main clock oscillator) or oscillation of the main clock has not yet become stable.*3, *5 1: Oscillation of the main clock is stable so the clock is available for use as the system clock.*4	R
b1	SOOVF	Sub-Clock Oscillation Stabilization Flag	0: SOSTP = 1 (stopping the sub-clock oscillator) or oscillation of the sub-clock has not yet become stable.*5 1: Oscillation of the sub-clock is stable so the clock is available for use as the system clock.*4	R
b2	PLOVF	PLL Clock Oscillation Stabilization Flag	0: The PLL clock is stopped or oscillation of the PLL clock has not yet become stable. 1: Oscillation of the PLL clock is stable so the clock is available for use as the system clock.	R
b3	HCOVF*2	HOCO Clock Oscillation Stabilization Flag	0: The HOCO clock is stopped or oscillation of the HOCO clock has not yet become stable. 1: Oscillation of the HOCO clock is stable so the clock is available for use as the system clock.	R
b4	ILCOVF*1	IWDT-Dedicated Clock Oscillation Stabilization Flag	0: The IWDT-dedicated on-chip oscillator is stopped or oscillation of the IWDT-dedicated on-chip oscillator has not yet become stable. 1: Oscillation of the IWDT-dedicated on-chip oscillator is stable.	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The ILCOVF flag value after a reset is 1 when the OFS0.IWDTSTRT bit is 0. It is 0 when the OFS0.IWDTSTRT bit is 1.
Note 2. The HCOVF flag value after a reset is 1 when the OFS1.HOCOEN bit is 0. It is 0 when the OFS1.HOCOEN bit is 1.
Note 3. The MOOVF flag does not reflect the control of the main clock oscillator by the MOFCR.MOFXIN bit. Accordingly, if the MOSTP bit is set to 1 while the MOFXIN bit is 1, the main clock oscillator continues to oscillate but the setting of the MOOVF flag becomes 0.
Note 4. The SOOVF flag does not reflect the control of the sub-clock oscillator by the RCR3.RTCEN bit. Accordingly, if the SOSTP bit is set to 1 while the RTCEN bit is 1, the sub-clock oscillator continues to oscillate but the setting of the SOOVF flag becomes 0.
Note 5. If the value set in the wait control register of the main clock oscillator and sub-clock oscillator is not sufficient for the given oscillation stabilization time, the oscillation stabilization flag is set to 1 and supply of the clock signal to the internal circuits starts before oscillation is stable. This may cause malfunction of this MCU, so ensure that the setting of the wait control register is at least the oscillation settling time for the oscillator considering the maximum frequency of the LOCO clock.

OSCOVFSR contains flags to indicate the states of operation of the counters within the oscillation stabilization wait circuits for the individual oscillators.
The counters measure the waiting times until each oscillator output clock is supplied to the internal circuits after oscillation starts, and an overflow of a counter indicates the start of clock supply from the corresponding oscillator to the internal circuits.

MOOVF Flag (Main Clock Oscillation Stabilization Flag)

This flag indicates the state of operation of the counter that measures the waiting time for the main clock oscillator.
[Setting condition]

- After the main clock oscillator has stopped and the MOSCCR.MOSTP bit is set to 0, the number of LOCO clock cycles corresponding to the setting of the MOSCWTCR register being counted and supply of the main clock within the LSI starting.

[Clearing condition]

- After the main clock oscillator has started to operate and the MOSCCR.MOSTP bit has been set to 1, deactivation of the main clock oscillator being completed.

SOOVF Flag (Sub-Clock Oscillation Stabilization Flag)

This flag indicates the state of operation of the counter that measures the waiting time for the sub-clock oscillator.
[Setting condition]

- After the sub-clock oscillator has stopped and the SOSCCR.SOSTP bit is set to 0, the number of LOCO clock cycles corresponding to the setting of the SOSCWTCR register being counted and supply of the sub-clock within the LSI starting.

[Clearing condition]

- After the sub-clock oscillator has started to operate and the SOSCCR.SOSTP bit has been set to 1, deactivation of the sub-clock oscillator being completed.

PLOVF Flag (PLL Clock Oscillation Stabilization Flag)

This flag indicates the state of operation of the counter that measures the waiting time for the PLL.
[Setting condition]

- After the PLL has stopped and the PLLCR2.PLLEN bit is set to 0, 62 cycles of the LOCO clock being counted and supply of the PLL clock within the LSI starting.

If oscillation by the PLL clock source selected by the PLLCR.PLLSRCSEL bit is not stable when the PLLEN bit is set to 0, counting of LOCO clock cycles proceeds after the oscillation of the PLL clock source has been stabilized.

[Clearing condition]

- After the PLL has started to operate and the PLLCR2.PLLEN bit has been set to 1, deactivation of the PLL being completed.

HCOVF Flag (HOCO Clock Oscillation Stabilization Flag)

This flag indicates the state of operation of the counter that measures the waiting time for the high-speed on-chip oscillator.
[Setting condition]

- After the high-speed on-chip oscillator has stopped and the HOCOCR.HCSTP bit is set to 0, 25 cycles of the LOCO clock being counted and supply of the HOCO clock within the LSI starting.

[Clearing condition]

- After the high-speed on-chip oscillator has started to operate and the HOCOCR.HCSTP bit has been set to 1, deactivation of the high-speed on-chip oscillator being completed.

ILCOVF Flag (IWDT-Dedicated Clock Oscillation Stabilization Flag)

This flag indicates the state of operation of the counter that measures the waiting time for the IWDT-dedicated on-chip oscillator.
[Setting condition]

- After the IWDT-dedicated on-chip oscillator has stopped and the ILOCOCR.ILCSTP bit is set to 0, 34 cycles of the LOCO clock being counted and supply of the IWDT-dedicated clock within the LSI starting.

9.2.14 Oscillation Stop Detection Control Register (OSTDCR)

Address(es): 0008 0040h

	b7	b6	b5	b4	b3	b2	b1	b0
	OSTDE	—	—	—	—	—	—	OSTDIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OSTDIE	Oscillation Stop Detection Interrupt Enable	0: The oscillation stop detection interrupt is disabled. Oscillation stop detection is not notified to the POE. 1: The oscillation stop detection interrupt is enabled. Oscillation stop detection is notified to the POE.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	OSTDE	Oscillation Stop Detection Function Enable	0: Oscillation stop detection function is disabled. 1: Oscillation stop detection function is enabled.	R/W

OSTDCR register is used to enable the oscillation stop detection function for the main clock oscillator and conveying of interrupts in response.

OSTDIE Bit (Oscillation Stop Detection Interrupt Enable)

If the oscillation-stop detection flag in the oscillation-stop detection status register (OSTDSR.OSTDF) requires clearing, do this after clearing the OSTDIE bit to 0. Wait for at least two cycles of PCLKB before again setting the OSTDIE bit to 1. According to the number of cycles for access to read out a given I/O register, waiting time longer than two cycles of PCLKB may have to be secured.

OSTDE Bit (Oscillation Stop Detection Function Enable)

This bit enables or disables the oscillation stop detection function.

When the OSTDE bit is 1 (oscillation stop detection function enabled), the LOCO stop bit (LOCOCR.LCSTP) is set to 0 and the LOCO operation is started. The LOCO cannot be stopped while the oscillation stop detection function is enabled; writing 1 to the LOCOCR.LCSTP bit (LOCO stopped) is invalid.

When the oscillation stop detection flag in the oscillation stop detection status register (OSTDSR.OSTDF) is 1 (main clock oscillation stop detected), writing 0 to the OSTDE bit is invalid.

When the OSTDE bit is 1, a transition cannot be made to software standby mode or deep software standby mode. To make a transition to software standby mode or deep software standby mode, execute the WAIT instruction with the OSTDE bit being 0.

To check the oscillation stop detection flag in the oscillation stop detection status register (OSTDSR.OSTDF), wait for at least three cycles of ICLK after the OSTDE bit has been set to 1 (oscillation stop detection enabled).

9.2.15 Oscillation Stop Detection Status Register (OSTDSR)

Address(es): 0008 0041h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	OSTDF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OSTDF	Oscillation Stop Detection Flag	0: Stopping of the main clock oscillator has not been detected. 1: Stopping of the main clock oscillator has been detected.	R(/W) *1
b7 to b1	—	Reserved	These bits are read as 0 and cannot be modified.	R

Note 1. This bit can only be set to 0.

The OSTDSR register indicates the detection of stopping of the main clock oscillator.

OSTDF Flag (Oscillation Stop Detection Flag)

This bit is a flag to indicate the main clock status. When the OSTDF flag is 1, it indicates that the main clock oscillation stop has been detected.

Once the main clock oscillation stop is detected, the OSTDF flag is not set to 0 even though the main clock oscillation is restarted. The OSTDF flag is set to 0 by reading 1 from the bit and then writing 0. At least 3 ICLK cycles of wait time is necessary between writing 0 to OSTDF and reading OSTDF as 0. If the OSTDF flag is set to 0 from 1 while the main clock oscillation is stopped, the OSTDF flag becomes 0 and then returns to 1.

The OSTDF flag cannot be modified to 0 while the main clock oscillator or PLL is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]) (010b or 100b). The OSTDF flag should be set to 0 after switching the clock source to other sources than the main clock oscillator and PLL.

[Setting condition]

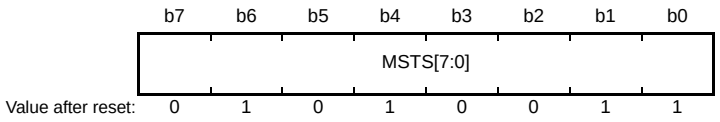
- The main clock oscillation is stopped with the OSTDCR.OSTDE being 1 (oscillation stop detection function enabled).

[Clearing condition]

- 1 is read and then 0 is written when the SCKCR3.CKSEL[2:0] bits are neither 010b nor 100b.

9.2.16 Main Clock Oscillator Wait Control Register (MOSCWTCR)

Address(es): 0008 00A2h



MOSCWTCR is used to control the waiting time until output of the signal from the main clock oscillator to the internal circuits starts. The oscillation stabilization wait circuit for the main clock oscillator measures the waiting time by counting the number of LOCO clock cycles corresponding to the setting of the MOSCWTCR register.
The oscillation stabilization wait circuit measures the waiting time and controls the clock supply within the LSI. When the main clock oscillator starts by setting the MOSCCR.MOSTP bit, the oscillation stabilization wait circuit starts counting the waiting time with the LOCO clock. The clock supply within the LSI is disabled over the period until counting of the set number of cycles is completed. After counting is completed, supply of the clock signal within the LSI starts and the OSCOVFSR.MOOVF flag is set to 1.
Counting of LOCO clock cycles by the oscillation stabilization wait circuit proceeds regardless of the setting of the LOCOCR.LCSTP bit. Hardware automatically controls running and stopping of the LOCO clock for measurement of the waiting time.
Values can only be written to MOSCWTCR while the MOSCCR.MOSTP bit is 1 or the OSCOVFSR.MOOVF flag is 1; do not attempt writing to MOSCWTCR if neither is the case.
The waiting time is not required when an external clock signal is input for the main clock oscillator. Set the MSTS[7:0] bits to 00h.

The value of the MSTS[7:0] bits required for correspondence with the waiting time required to secure stable oscillation by the main clock oscillator is obtained by using the maximum frequency for fLOCO in the formula below.

MSTS[7:0] > (tMAINOSC × (fLOCO_max) + 16)/32
(tMAINOSC: main clock oscillation stabilization time; fLOCO_max: maximum frequency for fLOCO)

If tMAINOSC is 1 ms and fLOCO_max is 264 kHz (the period is 1/3.78 µs), the formula gives MSTS[7:0] > (1 ms × (264 kHz) + 16)/32 = 8.75, so set the MSTS[7:0] bits to 9.

Waiting time:
When LOCO is at its highest frequency: (9 × 32 - 16) × (1/264 kHz = 3.78 µs) = 1.028 ms
When LOCO is at its normal frequency: (9 × 32 + 3) × (1/240 kHz = 4.18 µs) = 1.216 ms
When LOCO is at its lowest frequency: (9 × 32 + 10) × (1/216 kHz = 4.63 µs) = 1.380 ms

9.2.17 Sub-Clock Oscillator Wait Control Register (SOSCWTCR)

Address(es): 0008 00A3h

	b7	b6	b5	b4	b3	b2	b1	b0
	SSTS[7:0]							
Value after reset:	0	0	1	0	0	0	0	1

SOSCWTCR is used to control the waiting time until output of the signal from the sub-clock oscillator to the internal circuits starts. The oscillation stabilization wait circuit for the sub-clock oscillator measures the waiting time by counting the number of LOCO clock cycles corresponding to the setting of the SOSCWTCR register.
The oscillation stabilization wait circuit measures the waiting time and controls the clock supply within the LSI. When the sub-clock oscillator starts by setting the SOSCCR.SOSTP bit, the oscillation stabilization wait circuit starts counting the waiting time with the LOCO clock. The clock supply within the LSI is disabled over the period until counting of the set number of cycles is completed. After counting is completed, supply of the clock signal within the LSI starts and the OSCOVFSR.SOOVF flag is set to 1.
Counting of LOCO clock cycles by the oscillation stabilization wait circuit proceeds regardless of the setting of the LOCOCR.LCSTP bit. Hardware automatically controls running and stopping of the LOCO clock for measurement of the waiting time.
Values can only be written to SOSCWTCR while the SOSCCR.LCSTP bit is 1 or the OSCOVFSR.SOOVF flag is 1; do not attempt writing to SOSCWTCR if neither is the case.

The value of the SSTS[7:0] bits required for correspondence with the expected time to secure settling of oscillation by the sub-clock oscillator is obtained by using the maximum frequency for fLOCO in the formula below.

SSTS[7:0] > (tSUBOSC × (fLOCO_max) + 16)/16384
(tSUBOSC: sub-clock oscillation stabilization time; fLOCO_max: maximum frequency for fLOCO)

If tSUBOSC is 2 s and fLOCO is 264 kHz (the period is 1/3.78 μs), the formula gives SSTS[7:0] > (2 s × (264 kHz) + 16)/16384 = 32.22, so set the SSTS[7:0] bits to 33.

Waiting time:
When LOCO is at its highest frequency: (33 × 16384 - 16) × (1/264 kHz = 3.78 μs) = 2.044 s
When LOCO is at its normal frequency: (33 × 16384 + 3) × (1/240 kHz = 4.18 μs) = 2.260 s
When LOCO is at its lowest frequency: (33 × 16384 + 10) × (1/216 kHz = 4.63 μs) = 2.503 s

9.2.18 Main Clock Oscillator Forced Oscillation Control Register (MOFCR)

Address(es): 0008 C293h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MOSEL	MODRV2[1:0]		—	—	—	MOFXIN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MOFXIN	Main Clock Oscillator Forced Oscillation	0: Oscillator is not controlled by this bit. 1: The main clock oscillator is forcedly oscillated.	R/W
b3 to b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	MODRV2[1:0]	Main Clock Oscillator Driving Ability 2 Switching	b5 b4 0 0: 20.1 to 24 MHz 0 1: 16.1 to 20 MHz 1 0: 8.1 to 16 MHz 1 1: 8 MHz	R/W
b6	MOSEL	Main Clock Oscillator Switching	0: Resonator 1: External clock input	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

As well as selecting and de-selecting forcible starting of the main clock oscillator, the setting of the MOFCR also selects the driving ability, and the oscillator or an external clock signal.

MOFXIN Bit (Main Clock Oscillator Forced Oscillation)

This bit controls forced oscillation of the main clock oscillator. Although transitions to software standby or deep software standby normally stop oscillation of the main clock oscillator, setting this bit to 1 forces oscillation by the main clock oscillator so that it can still be used as the clock source for the RTC in either standby mode.
When changing the value of the MOSCCR.MOSTP bit or MOFXIN bit, execute subsequent instructions after reading the bit and checking that its value has actually been updated (refer to (2) Notes on writing to I/O registers, in section 5, I/O Registers).

MODRV2[1:0] Bits (Main Clock Oscillator Driving Ability 2 Switching)

These bits switch the driving ability of the main clock oscillator.
Specify the driving ability according to the frequency of a crystal connected to the main clock oscillator.
The frequency ranges specified in the bit description of the MODRV2[1:0] bits are the reference values of the crystal with capacitive load of 8 pF. A setting value may not fit within the frequency range depending on a crystal. Use values recommended by the resonator manufacturer.
In case of a ceramic resonator, it may be better to select lower frequency range than the frequency of the resonator (for example, specify 10b instead of 01b when a ceramic resonator with the frequency range from 16.1 to 20 MHz is used). Use values recommended by the resonator manufacturer.

MOSEL Bit (Main Clock Oscillator Switching)

This bit switches the source for the main clock oscillator.

9.2.19 High-Speed On-Chip Oscillator Power Supply Control Register (HOCOPCR)

Address(es): 0008 C294h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	HOCOPCNT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	HOCOPCNT	High-Speed On-Chip Oscillator Power Supply Control	0: Turns the power supply of the HOCO on. 1: Turns the power supply of the HOCO off.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

HOCOPCNT Bit (High-Speed On-Chip Oscillator Power Supply Control)

This bit controls the power supply for the HOCO.

When this bit is set to 0, the power supply of the HOCO is turned on, enabling oscillation.

When this bit is set to 1, the power supply of the HOCO is turned off, reducing power consumption.

When setting the HOCOPCNT bit to 1, set the HOCO stop bit in the high-speed on-chip oscillator control register (HOCOCR.HCSTP) to 1 (HOCO stopped) beforehand.

After the HOCOPCNT bit is changed from 1 to 0, oscillation settling time is required before the HOCOCR.HCSTP bit is set to 0. For details, see section 64, Electrical Characteristics.

Do not change the value of the HOCOPCNT bit in the following cases:

- When the HOCO is selected as the clock source by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).
- When the setting of the operating power control mode select bits in the operating power control register (OPCCR.OPCM[2:0]) is for low-speed operating mode 1 or 2.

9.3 Main Clock Oscillator

There are two ways of supplying the clock signal to the main clock oscillator: connecting an oscillator or the input of an external clock signal.

9.3.1 Connecting a Crystal Resonator

Figure 9.2 shows an example of connecting a crystal.
Connect capacitors referring to the capacitive load of the crystal to be used. In addition, a damping resistor Rd should be added, if necessary. The values of capacitors and resistor vary depending on the resonator and the oscillator driving ability. Use values recommended by the resonator manufacturer. If use of an external feedback resistor (Rf) is directed by the resonator manufacturer, insert an Rf between EXTAL and XTAL by following the instruction.
When connecting a resonator to supply the clock, the frequency of the crystal must be in the frequency range of the resonator for the main clock oscillator described in Table 9.1.
When a resonator is connected, setting the MOFCR.MODRV2[1:0] bits (Main Clock Oscillator Driving Ability 2 Switching) is required.
The frequency ranges that are specified in the bit description of the MODRV2[1:0] bits are the reference values of the crystal with capacitive load of C_L = 8 pF. The setting value may not fit within the frequency range depending on a crystal. Use values recommended by the resonator manufacturer.
In case of a ceramic resonator, it may be better to select lower frequency range than the frequency of the resonator. (For example, specify 10b instead of 01b when a ceramic resonator with the frequency range of 16.1 to 20 MHz is used). Use values recommended by the resonator manufacturer.

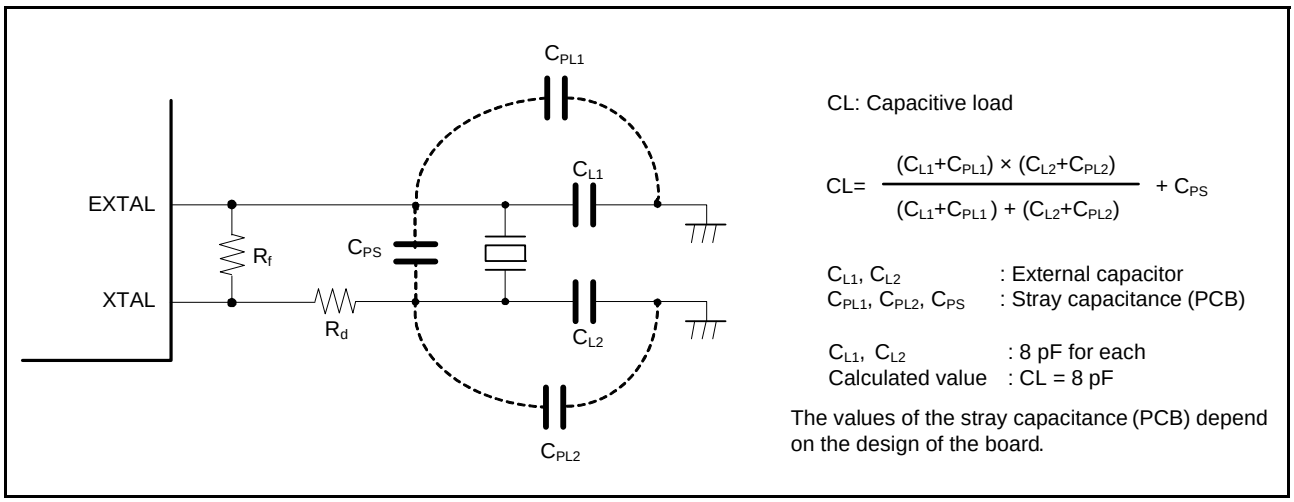


Figure 9.2 Example of Crystal Connection

Table 9.3 Damping Resistance (Reference Values)

Frequency (MHz)	8	12	16	20	24
Rd (Ω)	0	0	0	0	0

Figure 9.3 shows an equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in Table 9.4.

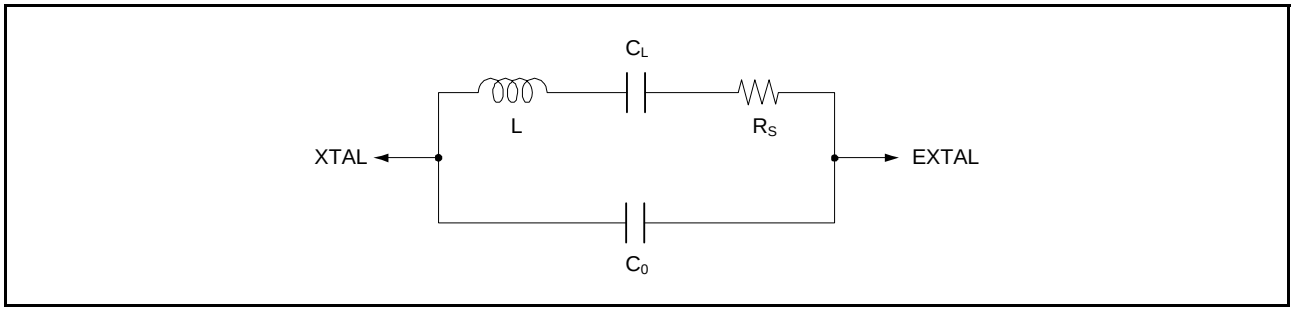


Figure 9.3 Equivalent Circuit of Crystal Resonator

Table 9.4 Crystal Characteristics (Reference Values)

Frequency (MHz)	8	12	16	20	24
R_S max (Ω)	300	100	80	50	50

9.3.2 External Clock Input

Figure 9.4 shows examples of connection of external clock input. To operate the oscillator by inputting an external clock signal, set the MOFCR.MOSEL bit to 1. At this time, the XTAL pin becomes high impedance.

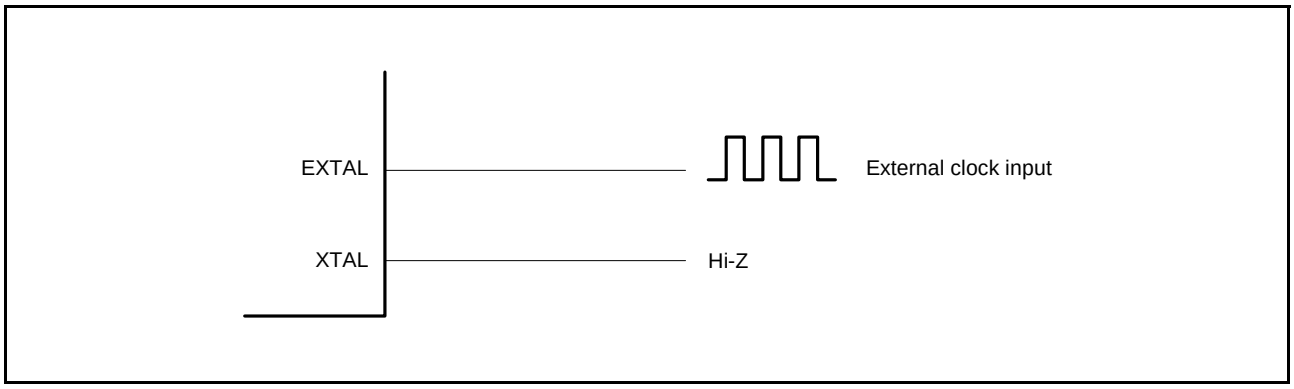


Figure 9.4 Equivalent Circuit for Crystal Resonator

9.3.3 Notes on the External Clock Input

The frequency of the external clock input can only be changed while the main clock oscillator is stopped. Do not change the frequency of the external clock input while the setting of the main clock oscillator stop bit (MOSCCR.MOSTP) is 0 (making the main clock oscillator run) or that of the main clock oscillator forced oscillation bit (MOFCR.MOFXIN) is 1 (forcing the main clock oscillator to run).

9.4 Sub-Clock Oscillator

To supply a clock to the sub-clock oscillator, connect a crystal resonator.

9.4.1 Connecting 32.768-kHz Crystal Resonator

To supply a clock to the sub-clock oscillator, connect a 32.768-kHz crystal resonator, as shown in Figure 9.5.
A damping resistor Rd should be added, if necessary. Since the resistor values vary depending on the resonator and the oscillation driving ability, use values recommended by the resonator manufacturer. If use of an external feedback resistor (Rf) is directed by the resonator manufacturer, insert an Rf between XCIN and XCOUT by following the instruction.
When connecting a resonator to supply the clock, the frequency of the resonator should be in the frequency range of the resonator for the sub-clock oscillator described in Table 9.1.

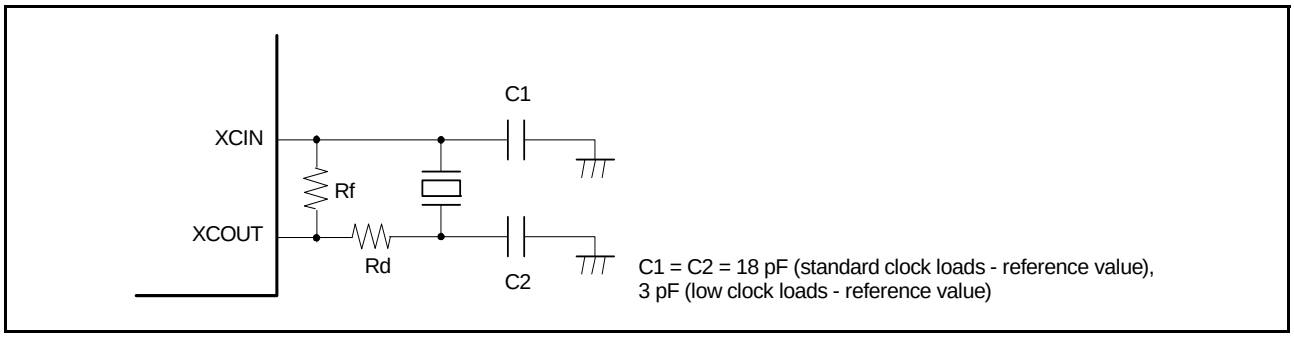


Figure 9.5 Connection Example of 32.768-kHz Crystal Resonator

Figure 9.6 shows an equivalent circuit for the 32.768-kHz crystal resonator. Use a crystal resonator that has the characteristics listed in Table 9.5.

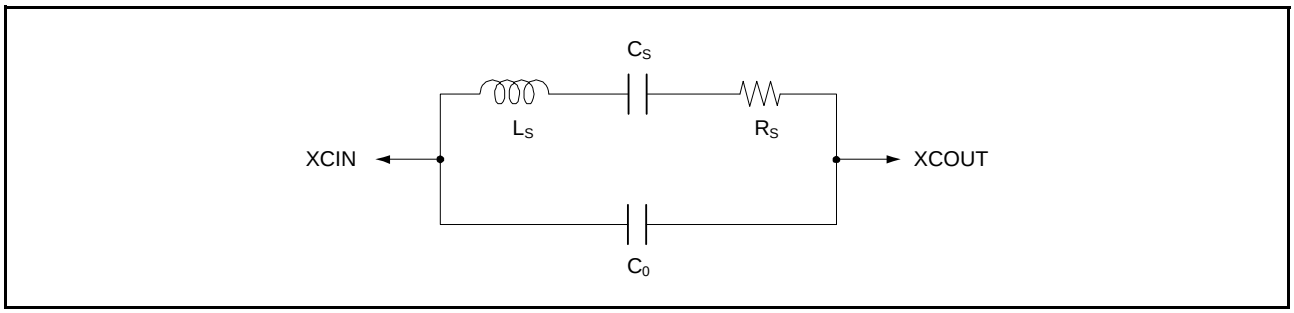


Figure 9.6 Equivalent Circuit for Crystal Resonator

Table 9.5 Crystal Resonator Characteristics (Reference Values)

Frequency (kHz)	32.768 (Low Clock Loads)	32.768 (Standard Clock Loads)
R_S max (kΩ)	60	60

9.4.2 Handling of Pins when Sub-Clock is Not Used

If the sub-clock is not in use, connect the XCIN pin to VSS via a resistor (to pull VSS down) and leave the XCOUT pin open-circuit as shown in Figure 9.7.
In addition, if an oscillator is not connected, set the sub-clock oscillator stop bit (SOSCCR.SOSTP) to 1 (stopping the oscillator) and the sub-clock oscillator control bit in RTC control register 3 (RCR3.RTCEN) to 0 (stopping the sub-clock oscillator). The value of some RTC registers related to the sub-clock will be undefined after a cold start. Accordingly, be sure to set these bits after a cold start.

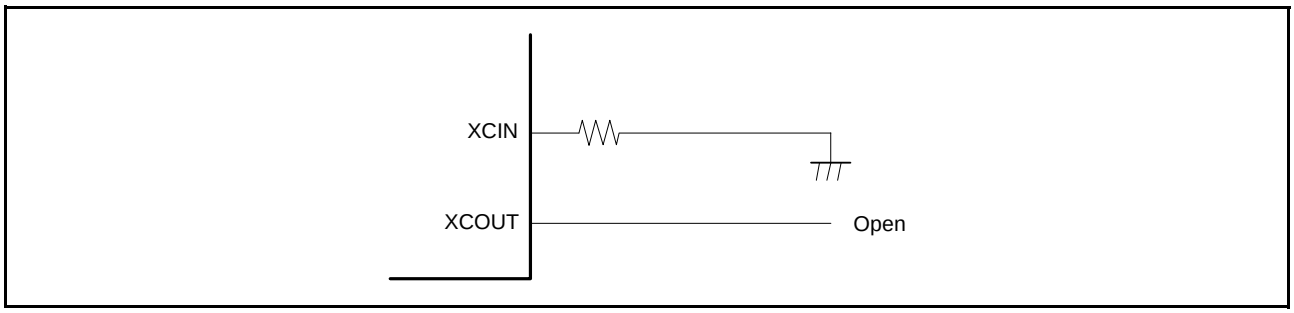


Figure 9.7 Pin Handling when Sub-Clock is Not Used

9.5 Oscillation Stop Detection Function

9.5.1 Oscillation Stop Detection and Operation after Detection

The oscillation stop detection function is used to detect the main clock oscillator stop and to supply LOCO clock pulses from the low-speed on-chip oscillator as the system clock source instead of the main clock or PLL clock. When the HOCO is selected as the clock source for the PLL and the PLL clock is selected as the system clock, the system clock is not switched to the LOCO even if stopping of the main clock is detected.

An oscillation stop detection interrupt request can be generated when an oscillation stop is detected. In addition, the MTU3 and GPT output can be forcedly driven to the high-impedance on the detection. For details, see section 24, Multi-Function Timer Pulse Unit (MTU3a) and section 25, Port Output Enable 3 (POE3a).

In the MCU, the input clock remaining at a given level over a certain period due, for example, to a malfunction of the main clock oscillator, is the criterion to detect stopping of the main clock. For details of the detection period, see Table 64.51, Oscillation Stop Detection Circuit Characteristics.

When an oscillation stop is detected, the main clock or PLL clock selected by the clock source select bits (SCKCR3.CKSEL[2:0]) is switched to the LOCO clock by the corresponding selectors in the former stage.

Therefore, on detection of oscillation stopping while the PLL clock or system clock is selected as the source of the main clock, the system clock source is switched to the LOCO clock without a change of CKSEL[2:0].

Switching between the main clock and LOCO clock or between the PLL clock and LOCO clock is controlled by the oscillation stop detection flag (OSTDSR.OSTDF). The clock source is switched to the LOCO clock when the OSTDF flag is 1, and is switched to the main clock or PLL clock again when the OSTDF flag is set to 0. At this time, if the main clock or PLL clock is selected with the CKSEL[2:0] bits, the OSTDF flag cannot be set to 0. To switch the clock source to the main clock or PLL clock again after the oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the main clock or PLL clock and clear the OSTDF flag to 0. After that, check that the OSTDF flag is not 1, and then set the CKSEL[2:0] bits to the main clock or PLL clock after the specified oscillation settling time has elapsed.

After a reset is released, the main clock oscillator is stopped and the oscillation stop detection function is disabled. To enable the oscillation stop detection function, activate the main clock oscillator and write 1 to the oscillation stop detection function enable bit (OSTDCR.OSTDE) after confirming that the OSCOVFSR.MOOVF flag or OSCOVFSR.PLOVF flag have been set to 1.

The oscillation stop detection function is provided against the main clock stop by an external cause. Therefore, the oscillation stop detection function should be disabled before the main clock oscillator is stopped by the software or a transition is made to software standby mode or deep software standby mode.

The clocks that are switched to the LOCO clock by the oscillation stop detection are: the main clock, PLL clock, CAC main clock (CACMCLK), CAN clock (CANMCLK), and USBA clock (USBMCLK), which are provided as the system clock sources. The main clock as the RTC main clock source (RTCMCLK) is not switched to the LOCO clock.

Note that the frequencies of the derived clock signals after switching to the LOCO depends on the settings of the system clock control registers (SCKCR, SCKCR2, or SCKCR3).

Figure 9.8 shows an example of a flowchart for initialization of the oscillation stop detection function.

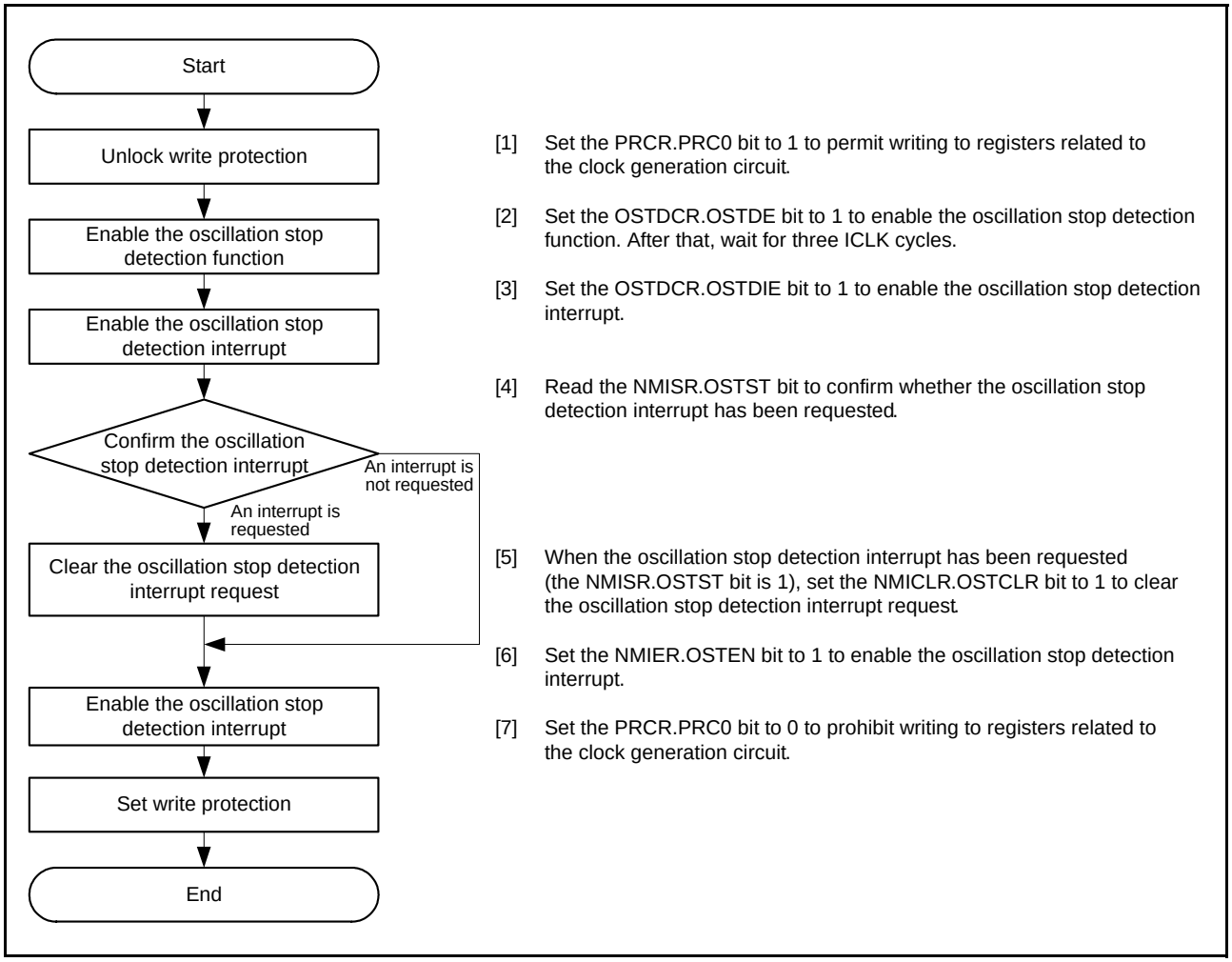
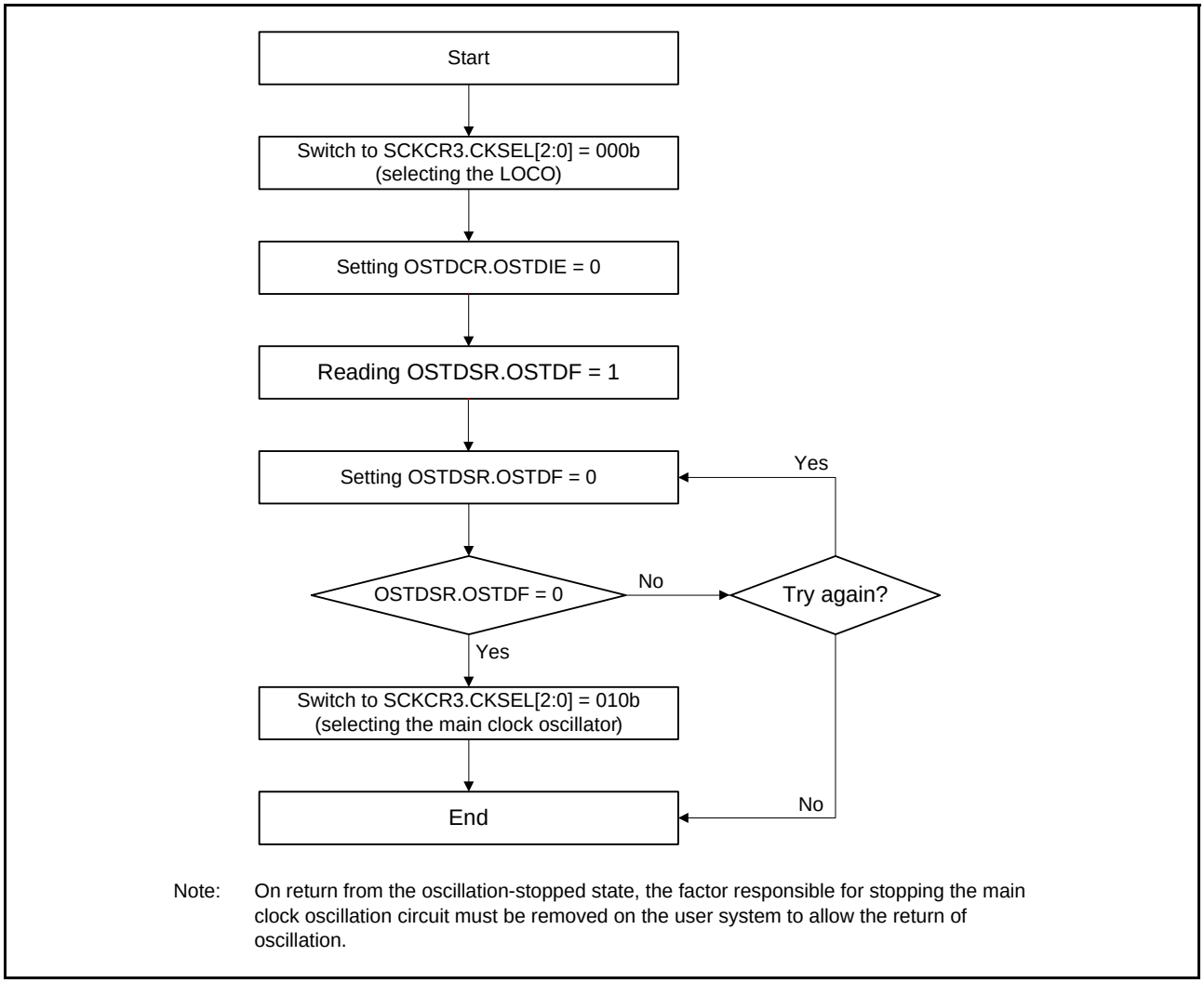


Figure 9.8 Flowchart Example for Initialization of Oscillation Stop Detection Function



Note: On return from the oscillation-stopped state, the factor responsible for stopping the main clock oscillation circuit must be removed on the user system to allow the return of oscillation.

Figure 9.9 Flowchart Example for Recovery from Detection of Oscillator Stop

9.5.2 Oscillation Stop Detection Interrupts

An oscillation-stop detection interrupt (OSTDI) will be generated if the oscillation-stop detection flag (OSTDSR.OSTDF) becomes 1 while the oscillation-stop detection interrupt enable bit in the oscillation stop detection control register (OSTDCR.OSTDIE) is 1 (enabling interrupt generation on oscillation stop detection). At this time, the main clock oscillator stop is notified to the port output enable 3 (POE). On accepting the notification of the oscillation stop, the POE sets the OSTST high-impedance flag in input level control/status register 6 (ICSR6.OSTSTF) to 1. After the oscillation stop is detected, wait for at least 10 cycles of PCLKB before writing to this ICSR6.OSTSTF flag. When the OSTDSR.OSTDF flag requires clearing, do so after clearing the oscillation-stop detection interrupt enable bit in the oscillation stop detection control register (OSTDCR.OSTDIE). Wait for at least two cycles of PCLKB clock before again setting the OSTDCR.OSTDIE bit to 1. According to the number of cycles for access to read out a given I/O register, waiting time longer than two cycles of PCLKB may have to be secured.

If the oscillation stop detection interrupt is to be used as a non-maskable interrupt, since non-maskable interrupts are disabled in the initial state after a reset release, set the corresponding bit in the NMIER register to 1 by software to enable non-maskable interrupts. If it is to be used as a maskable interrupt, do not change the value of the NMIER register from the value after a reset. For details, see section 15, Interrupt Controller (ICUA).

9.6 PLL Circuit

The PLL circuit has a function to multiply the frequency from the oscillator.

9.7 Internal Clock

Clock sources of internal clock signals are the main clock, sub-clock, HOCO clock, LOCO clock, PLL clock, dedicated clock for the IWDT, and the JTAG external clock. The internal clocks listed in the table below are produced from these sources.

Frequencies of the internal clocks are set by the combination of the divisors selected by the FCK[3:0], ICK[3:0], BCK[3:0], PCKA[3:0], PCKB[3:0], PCKC[3:0], and PCKD[3:0] bits in SCKCR, the BCLKDIV bit in BCKCR, the UCK[3:0] bits in SCKCR2, the clock source selected by the CKSEL[2:0] bits in SCKCR3, the bits that select the frequency of the PLL circuit (STC[5:0] and PLIDIV[1:0] in PLLCR), and the HCFRQ[1:0] bits in HOCOCR2. If the value of any of these bits is changed, subsequent operation will be at the frequency determined by the new value.

Table 9.6 Internal Clocks and Supply Destination Modules

	Type of Internal Clock	Clock Name	Supply Destination Module
1	System clock	ICLK	CPU, code flash memory, RAM (ECCRAM), ICU, BSC, DMAC, DTC, EXDMAC, MPU
2	Peripheral module clocks	PCLKA	MTU3, GPT, SCIF, RSPI, USBA, ETHERC, EPTPC, EDMAC, AES
		PCLKB	TPU, PPG, TMR, CMT, CMTW, RTC, WDT, IWDT, POE3, SCI, RIIC, CAN, USBb, USBA, SDHI, MMCIF, QSPI, SSI, SRC, PDC, S12AD, R12DA, temperature sensor, CRC, DOC, CAC, RNG, DES, SHA, standby RAM, I/O, MPC, ICU
		PCLKC	S12AD (unit 0)
		PCLKD	S12AD (unit 1)
3	Flash-IF clock	FCLK	Data flash memory, code flash memory
4	External bus clock	BCLK	BSC, I/O
5	SDRAM clock	SDCLK	I/O
6	USB clock	UCLK	USBb, USBA
7	USBA clock	USBMCLK	USBA
8	CAN clock	CANMCLK	CAN
9	CAC clocks	CACMCLK (Main clock)	CAC
		CACSCLK (Sub clock)	
		CACHCLK (HOCO clock)	
		CACLCLK (LOCO clock)	
		CACILCLK (IWDT-dedicated clock)	
10	RTC clocks	RTCMCLK (Main clock)	RTC
		RTCSCLK (Sub clock)	
11	IWDT-dedicated clock	IWDTCLK	IWDT, CAC
12	JTAG clock	JTAGTCK	Boundary scan

9.7.1 System Clock

The system clock (ICLK) is used as the operating clock of the CPU, DMAC, DTC, code flash memory, and RAM.
The ICLK frequency is specified by the ICK[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, the STC[5:0], and PLIDIV[1:0] bits in PLLCR, and the HCFRQ[1:0] bits in HOCOCR2.

9.7.2 Peripheral Module Clock

The peripheral module clocks (PCLKA, PCLKB, PCLKC, and PCLKD) are the operating clocks for use by peripheral modules.
The frequency of the given clock is specified by the PCKA[3:0], PCKB[3:0], PCKC[3:0], and PCKD[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, the STC[5:0], and PLIDIV[1:0] bits in PLLCR, and the HCFRQ[1:0] bits in HOCOCR2. The peripheral module clocks can be set to frequencies above that of the system clock.

9.7.3 Flash-IF Clock

The flash-interface clock (FCLK) is used as the operating clock for the flash-memory interfaces. That is, FCLK is used for programming and erasure of the code flash memory and data flash memory, and reading from the data flash memory.
The FCLK frequency is specified by the FCK[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR, and the HCFRQ[1:0] bits in HOCOCR2.

9.7.4 External Bus Clock

The external bus clock (BCLK) is an operating clock for the external bus controller and EXDMAC. It is also output externally from the BCLK pin to the external bus. When the external bus is enabled, P53 that is function-multiplexed with the BCLK pin cannot be used as an I/O port.
BCLK can be output from the BCLK pin by setting the SCKCR.PSTOP1 bit to 0 and setting the external bus enable bit in the system control register 0 (SYSCR0.EXBE) to 1. Make sure that modification of the SYSCR0.EXBE bit to 1 must always be performed while the PSTOP1 bit in SCKCR is 1.
When the BCKCR.BCLKDIV bit is set to 1, the BCLK clock divided by 2 is output from the BCLK pin.
The BCLK frequency is specified by the BCK[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR, and the HCFRQ[1:0] bits in HOCOCR2.
A frequency higher than the system clock (ICLK) should not be set for the BCLK. If such a frequency is set, the clock frequency will be the same as the ICLK.

9.7.5 SDRAM Clock

The SDRAM clock (SDCLK) is an operating clock for the external bus controller. It is output externally from the SDCLK pin for the SDRAM that is connected to the external bus.
When the SCKCR.PSTOP0 bit is set to 0 and the SDCLK enable bit (SDCLKE) in the external bus control register 1 (PFBCR1) is set to 1 (enabling SDCLK output), it selects output of SDCLK on the SDCLK pin. When changing the value of the PFBCR1.SDCLKE bit, make sure that the value of the SCKCR.PSTOP0 bit is 1.
The SDCLK clock frequency is specified by the SCKCR.BCK[3:0], SCKCR3.CKSEL[2:0], PLLCR.STC[5:0] and PLIDIV[1:0], and HOCOCR2.HCFRQ[1:0] bits.
A frequency higher than the system clock (ICLK) should not be set for the SDCLK. If such a frequency is set, the clock frequency will be the same as the ICLK.

9.7.6 USB Clock

The USB clock (UCLK) is used as the operating clock for USBb and the PHY clock for USBA. If UCLK is to be used as the PHY clock for USBA, set PCLKB to 60 MHz, and then set the PHYSET.HSEB bit to 1 to place USBA in CL only mode (FS mode or LS mode).
Use the SCKCR2.UCK[3:0], SCKCR3.CKSEL[2:0], PLLCR.STC[5:0], or PLLCR.PLIDIV[1:0] bits to set the UCLK frequency. The UCLK frequency must be set to 48 MHz. To derive the PHY clock for USBA, USBMCLK must also be supplied to the PLL in the USB-PHY layer of the USBA interface to generate the PHY clock. For details, see section 9.7.7, USBA Clock.

9.7.7 USBA Clock

The USBA clock (USBMCLK) is the clock signal for the PLL in the USB-PHY layer of the USBA interface.
The PLL of USBA generates the PHY clock. The frequency of the USBMCLK signal from the main clock oscillator will be 20 MHz or 24 MHz. Set the PHYSET.CLKSEL[1:0] bits in the USBA for either 20 MHz or 24 MHz in accord with the frequency of the main clock oscillator while the setting of the PHYSET.HSEB bit is 0. For the clock input to the PLL in the USB-PHY layer, the clock specification for USB2.0 must be strictly followed. For the clock settings, see section 39.3.3, Supplying Clocks.

9.7.8 CAN Clock

The CAN clock (CANMCLK) is an operating clock for the CAN.
CANMCLK is generated by the main clock oscillator.

9.7.9 CAC Clock (CACCLK)

The CAC clock (CACCLK) is an operating clock for the CAC.
CACCLK includes CACMCLK generated by the main clock oscillator, CACSCLK generated by the sub-clock oscillator, CACHCLK generated by the high-speed on-chip oscillator, CACLCLK generated by the low-speed on-chip oscillator, CACILCLK generated by the IWDT-dedicated on-chip oscillator, and PCLKB supplied to peripheral modules.

9.7.10 RTC Clock

The RTC clock (RTCSCLK, RTCMCLK) is the operating clock for the RTC.
RTCSCLK is generated by the sub-clock oscillator, and RTCMCLK is generated by the main clock oscillator.

9.7.11 IWDT-Dedicated Clock

The IWDT-dedicated clock (IWDTCLK) is the operating clock for the IWDT.
IWDTCLK is internally generated by the IWDT-dedicated on-chip oscillator.

9.7.12 JTAG Clock

The JTAG clock (JTAGTCK) is the operating clock for the JTAG.
JTAGTCK is generated by the JTAG external clock (TCK).

9.8 Clock Source Switching

In this MCU, the clock signal from the LOCO, which oscillates during release from the reset state, is used to start the fetching of CPU instructions after the internal reset time (tRESWT) has elapsed. After that, set up the clock to which the CPU will be switched while it is still driven by the LOCO, and read the oscillation stabilization flag register to confirm that oscillation of the given clock signal is stable before switching to the selected clock source.

(1) Example of procedure for settings to switch the system clock source from the LOCO to the PLL (clock source for the PLL: main clock) after release from an internal reset

1. After release from the internal reset state, set the driving ability by writing to the MODRV2[1:0] bits in the MOFCR register.
2. Set the oscillation settling time for the main clock oscillator by writing to the MSTS[7:0] bits in the MOSCWTCR register.
3. Make the main clock oscillator run by setting the MOSTP bit in the MOSCCR register.
4. Set the frequency multiplication factor in the PLLCR register (the initial setting for the PLL clock source selects the main clock oscillator).
5. Select PLL operation by writing to the PLL stop control bit in the PLLCR2 register.
6. Confirm that the PLL clock has become stable by reading the PLOVF flag in the OSCOVFSR register.
7. Set the division ratios after switching the clock sources by the SCKCR and SCKCR2 registers.
8. Change the clock signal from the LOCO clock to the PLL clock by writing to the CKSEL[2:0] bits in the SCKCR3 register.

(2) Example of procedure for settings to switch the system clock source from the LOCO to the PLL (clock source for the PLL: HOCO) after release from an internal reset

1. After release from the internal reset state, set the frequency by writing to the HCFRQ[1:0] bits in the HOCOCR2 register.
2. Make the HOCO clock run by setting the HCSTP bit in the HOCOCR register (the initial value depends on the value of the OFS1.HOCOEN bit; if the OFS1.HOCOEN bit is 0, an explicit setting to make the HOCO clock run is not required. The oscillation settling time for the HOCO is 25 cycles of the LOCO).
3. Set the frequency multiplication factor and set the HOCO clock as the PLL clock source by writing to the PLLCR register.
4. Select PLL operation by writing to the PLL stop control bit in the PLLCR2 register.
5. Confirm that the PLL clock has become stable by reading the PLOVF flag in the OSCOVFSR register.
6. Set the division ratios after switching the clock sources by the SCKCR and SCKCR2 registers.
7. Change the clock signal from the LOCO clock to the PLL clock by writing to the CKSEL[2:0] bits in the SCKCR3 register.

(3) Example of procedure for settings to switch the system clock source from the LOCO to the main clock after release from an internal reset

1. After release from the internal reset state, set the driving ability by writing to the MODRV2[1:0] bits in the MOFCR register.
2. Set the oscillation settling time for the main clock oscillator by writing to the MSTS[7:0] bits in the MOSCWTCR register.
3. Make the main clock oscillator run by setting the MOSTP bit in the MOSCCR register.
4. Confirm that the main clock has become stable by reading the MOOVF flag in the OSCOVFSR register.
5. Set the division ratios after switching the clock sources by the SCKCR and SCKCR2 registers.
6. Change the clock signal from the LOCO clock to the main clock by writing to the CKSEL[2:0] bits in the SCKCR3 register.

(4) Example of procedure for settings to switch the system clock source from the LOCO to the HOCO after release from an internal reset

1. After release from the internal reset state, set the frequency by writing to the HCFRQ[1:0] bits in the HOCOCR2 register.
2. Make the HOCO clock run by setting the HCSTP bit in the HOCOCR register (the initial value depends on the value of the OFS1.HOCOEN bit; if the OFS1.HOCOEN bit is 0, an explicit setting to make the HOCO clock run is not required. The oscillation settling time for the HOCO is 25 cycles of the LOCO).
3. Confirm that the HOCO clock has become stable by reading the HCOVF flag in the OSCOVFSR register.
4. Set the division ratios after switching the clock sources by the SCKCR and SCKCR2 registers.
5. Change the clock signal from the LOCO clock to the HOCO clock by writing to the CKSEL[2:0] bits in the SCKCR3 register.

9.9 Operations Linked by the ELC

9.9.1 Event Signal Output to the ELC

The clock generation circuit is capable of operation linked with another module set in advance when its interrupt request signal is used as an even signal by the event link controller (ELC) on detection of stopping of the main clock oscillation. The clock generation circuit outputs the event signal regardless of the setting of the corresponding oscillation stop detection interrupt enable bit (OSTDCR.OSTDIE). For details, see section 21, Event Link Controller (ELC).

9.9.2 Clock Source Switching on Reception of the Event Signal from the ELC

The clock generation circuit is capable of switching the clock source to the low-speed on-chip oscillator in response to the event set in advance when the event specified in the ELSRn register of the ELC occurs.
While this function is in use, clock source switching on return from sleep mode cannot be used. For details, see section 11.2.7, Sleep Mode Return Clock Source Switching Register (RSTCKCR).

9.10 Usage Notes

9.10.1 Notes on Clock Generation Circuit

(1) The frequencies of the system clock (ICLK), peripheral module clock (PCLKA to PCLKD), flash-IF clock (FCLK), external bus clock (BCLK), and SDRAM clock (SDCLK) supplied to each module change according to the settings of SCKCR. Each frequency should meet the following:
Select each frequency that is within the operation guaranteed range of clock cycle time (tcyc) specified in AC characteristics of electrical characteristics.
The frequencies must not exceed the ranges listed in Table 9.1.
The peripheral modules operate on the PCLKB and PCLKA. Note therefore that the operating speed of modules such as the timer and SCI varies before and after the frequency is changed.
Furthermore, PCLKC (unit 0) and PCLKD (unit 1) are available as clocks for the A/D converter.
Do not set a higher frequency than PCLKB, which serves as the operating clock of the A/D converter.

(2) The following relation is required between the frequencies of the system clock (ICLK) and external bus clock (BCLK).
ICLK ≥ BCLK
Also, the following relation is required between the frequencies of the peripheral module clocks.
PCLKA ≥ PCLKB; PCLKB ≥ PCLKC; PCLKB ≥ PCLKD

(3) Do not change the clock frequency during external bus access. Furthermore, when access via the external bus is to start after a change to the clock frequency, only start access via the bus after confirming that the change to the frequencies has been completed.

(4) If the clock frequency is to be changed by modifying the value of SCKCR, SCKCR2, SCKCR3, or BCKCR register, wait for writing of the value to the register to be complete and the new frequency to be stable before starting subsequent processing. For the procedure to confirm the completion of writing to I/O registers, see (2) Notes on writing to I/O registers, in section 5, I/O Registers.

9.10.2 Notes on Resonator

Since various resonator characteristics relate closely to the user's board design, adequate evaluation is required on the user side before use, referencing the resonator connection example shown in this section. The circuit constants for the resonator depend on the resonator to be used and the stray capacitance of the mounting circuit. Therefore, the circuit constants should be determined in full consultation with the resonator manufacturer. The voltage to be applied between the resonator pins must be within the absolute maximum rating.

9.10.3 Notes on Board Design

When using a crystal resonator, place the resonator and its load capacitors as close to the XTAL and EXTAL pins as possible. Other signal lines should be routed away from the oscillation circuit as shown in Figure 9.10 to prevent electromagnetic induction from interfering with correct oscillation.

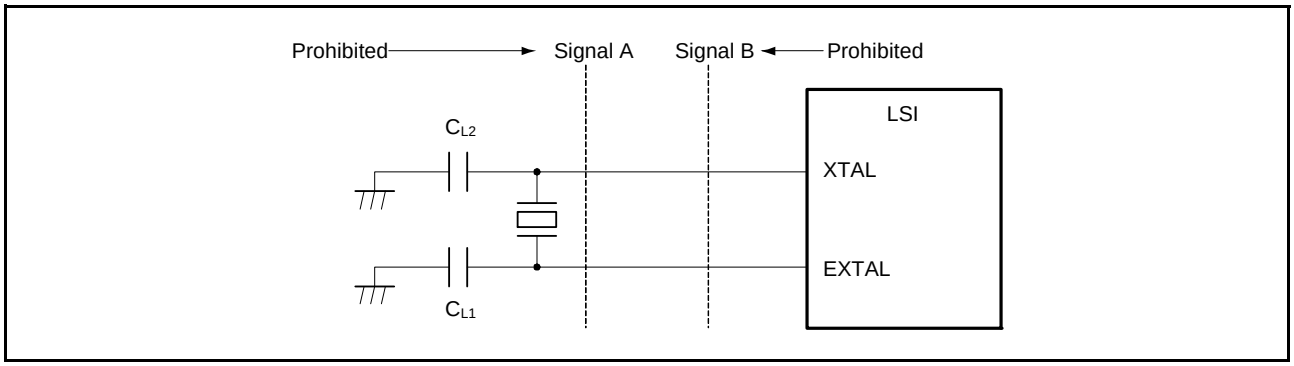


Figure 9.10 Notes on Board Design for Oscillation Circuit (Applies to the Sub-Clock Oscillator, in Case of the Main Clock Oscillator)

9.10.4 Notes on Resonator Connect Pin

When the main clock is not used, the EXTAL and XTAL pins can be used as general ports P36 and P37. When they are used as the general ports, the main clock should be stopped (MOSCCR.MOSTP should be set to 1 and MOFCR.MOFXIN should be set to 0). However, with the system using the main clock, the EXTAL (P36) and XTAL (P37) pins should not be used as output ports.
For the values of registers related to port settings, refer to Table 23.27, Register Settings.
Furthermore, since the main clock becomes essential if the function indicated below is in use, design the board so that both pins can be used for the main clock signal.

- Programming of flash memory in boot mode (USB interface)*1

Note 1. For the conditions on the oscillator in the various modes, see (8) in section 63.17, Usage Notes.

9.10.5 Notes on Sub-Clock Oscillator

The sub-clock can be used as the system clock, as the source to drive counting by the realtime clock, or as both. Accordingly, take note of the following limitations and points for caution regarding the settings, including when the sub-clock is not in use.

- To select the sub-clock as the system clock, set the SOSCCR.SOSTP bit; to set the sub-clock as the source to drive counting by the realtime clock, use the SOSCCR.SOSTP and RCR3.RTCEN bits. Furthermore, when the sub-clock is to be used as the clock source for counting by the realtime clock, the SOSCWTCR.SSTS[7:0] bits must be set to 00000000b after the oscillation stabilization waiting time has elapsed once the sub-clock has started oscillating. Make initial settings according to the procedure below and then follow the clock setting procedure described in section 32.3.2, Clock and Count Mode Setting Procedure.

  Procedure for initial settings:
  1. Set the RCR4.RCKSEL bit to 0 (selecting the sub-clock).
  2. Set the RCR3.RTCEN bit to 0.
  3. Read the RCR3.RTCEN bit and confirm that it is 0.
  4. Set the SOSCCR.SOSTP bit to 1.
  5. Read the SOSCCR.SOSTP bit and confirm that it is 1.
  6. Wait for the OSCOVFSR.SOOVF flag to be set to 0.
  7. Set the RCR3.RTCDV[2:0] bits.*1
  8. Confirm that the value of the RCR3.RTCDV[2:0] bits has changed.
  9. Set the SOSCWTCR.SSTS[7:0] bits to specify the wait time necessary for sub-clock oscillation.
  10. Set the SOSCCR.SOSTP bit to 0 (making the sub-clock oscillator run).
  11. Read the SOSCCR.SOSTP bit and confirm that it is 0.
  12. Wait for the OSCOVFSR.SOOVF flag to be set to 1.
  13. Set the SOSCWTCR.SSTS[7:0] bits to 00000000b.
  14. Set the RCR3.RTCEN bit to 1 (making the sub-clock oscillator run).
  15. Confirm that the value of the RCR3.RTCEN bit has changed to 1.

Note 1. If the RCR3.RTCDV[2:0] bits are set in this step, re-setting is not required in the clock setting procedure in section 32.3.2, Clock and Count Mode Setting Procedure.
Note 2. Steps 13 to 15 are not required if the sub-clock is used as the system clock only.

- When the sub-clock is used as the system clock, even if the RCR3.RTCEN bit is set to 1 and the sub-clock is already oscillating, wait for the OSCOVFSR.SOOVF flag to be set to 1 after the SOSCCR.SOSTP bit changes from 1 (stopped) to 0 (operating) before starting to use the sub-clock as the system clock.
- The state of the sub-clock control circuit is undefined after a cold start so the sub-clock must be initialized whether or not it is to be used. Setting both the SOSCCR.SOSTP bit and the RCR3.RTCEN bit to the “stopped” setting will initialize the control circuit. For initialization of the RCR3.RTCEN bit, see section 32, Realtime Clock (RTCd).
- For the sub-clock oscillator to operate, the RCR3.RTCDV[2:0] bits must also be set. This setting must be made while the sub-clock oscillator is stopped. Writing to these bits during operation is prohibited.
- If the RCR3.RTCEN bit is modified after the SOSCCR.SOSTP bit is modified, or if the SOSCCR.SOSTP bit is modified after the RCR3.RTCEN bit is modified, make sure that first modified bit has been rewritten before the subsequent bit is modified.

9.10.6 Notes on Using a Low CL Crystal Unit

When the RCR3.RTCDV[2:0] bits are 001b (drive capacity for low CL), the oscillator is susceptible to noise. Especially when the signal level of any pin near the XCIN or XCOUT pin is changed, the oscillation accuracy of the sub-clock oscillator may be affected. The accuracy is affected differently depending on the board traces and how the signal level of any pin near the XCIN or XCOUT pin is changed. When designing a board using a low CL crystal unit, refer to the application note “Design Guide for Low CL Sub-clock Circuits” (R01AN1187EJ) to reduce the influence from noise. The following are examples that may significantly affect oscillation accuracy:

(1) When connecting an on-chip debugging emulator to the FINED pin

Since the FINED pin (FINE interface pin) is near the XCIN and XCOUT pins, the oscillation accuracy of the sub-clock oscillator is affected when using the FINED pin in debugging. When using the FINED pin in debugging, keep using the low CL crystal unit and set the RCR3.RTCDV[2:0] bits to 110b (drive capacity for standard CL). However, this measure may affect the reliability of the crystal unit. Therefore, use this measure only when using an on-chip debugging emulator. Set the RCR3.RTCDV[2:0] bits to 001b (drive capacity for low CL) in mass production programs. The oscillation accuracy is not affected when connecting an on-chip debugging emulator to a JTAG pin (TCK, TRST, TMS, TDI, or TDO pin).

(2) When supplying an external clock to the main clock oscillator

When inputting an external clock to the EXTAL pin, the oscillation accuracy of the sub-clock oscillator may be affected. When inputting an inverted external clock to the XTAL pin, the oscillation accuracy will be affected more significantly.

9.10.7 Notes on Using Power-On Reset and PLL Circuit Together

When using a power-on reset and the PLL circuit together, set the LVD1CR1.LVD1IDTSEL[1:0] bits or LVD2CR1.LVD2IDTSEL[1:0] bits to 01b, and select the voltage monitoring interrupt to be generated when a drop (Vcc < Vdet) is detected.
In addition, at the beginning of the interrupt handling routine, set the SCKCR3.CKSEL[2:0] bits to a value other than 100b to select a clock source other than the PLL circuit, then set the PLLCR2.PLLEN bit to 1 to stop the PLL circuit.

10. Clock Frequency Accuracy Measurement Circuit (CAC)

10.1 Overview

The clock frequency accuracy measurement circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock to be used as a measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range.
When measurement is completed or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated.
Table 10.1 lists the specifications of the CAC and Figure 10.1 shows a block diagram of the CAC.

Table 10.1 CAC Specifications

Item	Description
Measurement target clocks	The frequency of the following clocks can be measured. • Main clock • Sub-clock • HOCO clock • LOCO clock • IWDTCLK clock • Peripheral module clock B (PCLKB)
Measurement reference clocks	• External clock input to the CACREF pin • Main clock • Sub-clock • HOCO clock • LOCO clock • IWDTCLK clock • Peripheral module clock B (PCLKB)
Selectable function	Digital filter function
Interrupt sources	• Measurement end interrupt • Frequency error interrupt • Overflow interrupt
Low power consumption function	Module stop state can be set.

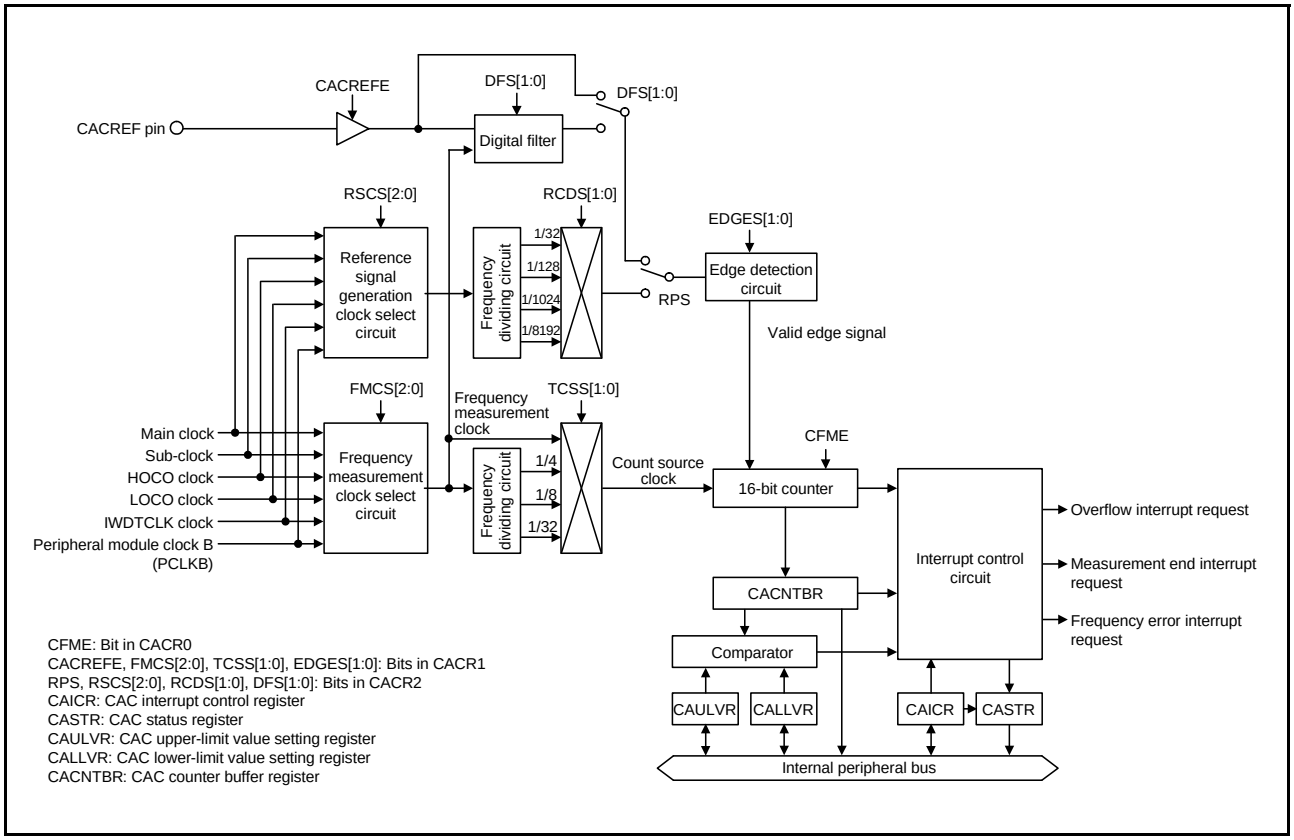


Figure 10.1 CAC Block Diagram

Table 10.2 shows the pin configuration of the CAC.

Table 10.2 Pin Configuration of CAC

Pin Name	I/O	Function
CACREF	Input	Measurement reference clock input pin

10.2 Register Descriptions

10.2.1 CAC Control Register 0 (CACR0)

Address(es): 0008 B000h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFME
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CFME	Clock Frequency Measurement Enable	0: Clock frequency measurement is disabled. 1: Clock frequency measurement is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CFME Bit (Clock Frequency Measurement Enable)

This bit specifies whether clock frequency measurement is enabled or disabled.

When rewriting this bit, more time is required than other bits for the new value to be reflected in the register. Further write access to this bit are ignored until the current write access is reflected in the register. Read the bit to confirm that the rewrite has been reflected in the register.

10.2.2 CAC Control Register 1 (CACR1)

Address(es): 0008 B001h

	b7	b6	b5	b4	b3	b2	b1	b0
	EDGES[1:0]		TCSS[1:0]		FMCS[2:0]			CACREFE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CACREFE	CACREF Pin Input Enable	0: CACREF pin input is disabled. 1: CACREF pin input is enabled.	R/W
b3 to b1	FMCS[2:0]	Measurement Target Clock Select	b3 b1 0 0 0: Main clock 0 0 1: Sub-clock 0 1 0: HOCO clock 0 1 1: LOCO clock 1 0 0: IWDTCLK clock 1 0 1: Peripheral module clock B (PCLKB) Settings other than above are prohibited.	R/W
b5, b4	TCSS[1:0]	Timer Count Clock Source Select	b5 b4 0 0: No division 0 1: ×1/4 clock 1 0: ×1/8 clock 1 1: ×1/32 clock	R/W
b7, b6	EDGES[1:0]	Valid Edge Select	b7 b6 0 0: Rising edge 0 1: Falling edge 1 0: Both rising and falling edges 1 1: Setting prohibited	R/W

Note 1. Set the CACR1 register when the CACR0.CFME bit is 0.

CACREFE Bit (CACREF Pin Input Enable)

This bit specifies whether the CACREF pin input is enabled or disabled.

FMCS[2:0]Bits (Measurement Target Clock Select)

These bits select the measurement target clock whose frequency is to be measured.

TCSS[1:0] Bits (Timer Count Clock Source Select)

These bits select the count clock source for the clock frequency accuracy measurement circuit.

EDGES[1:0]Bits (Valid Edge Select)

These bits select the valid edge for the reference signal.

10.2.3 CAC Control Register 2 (CACR2)

Address(es): 0008 B002h

	b7	b6	b5	b4	b3	b2	b1	b0
	DFS[1:0]		RCDS[1:0]		RSCS[2:0]			RPS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RPS	Reference Signal Select	0: CACREF pin input 1: Internal clock (internally generated signal)	R/W
b3 to b1	RSCS[2:0]	Measurement Reference Clock Select	b3 b1 0 0 0: Main clock 0 0 1: Sub-clock 0 1 0: HOCO clock 0 1 1: LOCO clock 1 0 0: IWDTCLK clock 1 0 1: Peripheral module clock B (PCLKB) Settings other than above are prohibited.	R/W
b5, b4	RCDS[1:0]	Measurement Reference Clock Frequency Division Ration Select	b5 b4 0 0: ×1/32 clock 0 1: ×1/128 clock 1 0: ×1/1024 clock 1 1: ×1/8192 clock	R/W
b7, b6	DFS[1:0]	Digital Filter Select	b7 b6 0 0: Digital filtering is disabled. 0 1: The sampling clock for the digital filter is the frequency measuring clock. 1 0: The sampling clock for the digital filter is the frequency measuring clock divided by 4. 1 1: The sampling clock for the digital filter is the frequency measuring clock divided by 16.	R/W

Note 1. Set the CACR2 register when the CACR0.CFME bit is 0.

RPS Bit (Reference Signal Select)

This bit selects whether to use the CACREF pin input or an internal clock (internally generated signal) as the reference signal.

RSCS[2:0]Bits (Measurement Reference Clock Select)

These bits select the clock source for generating the measurement reference clock.

RCDS[1:0]Bits (Measurement Reference Clock Frequency Division Ration Select)

These bits select the frequency division ratio of the measurement reference clock.

DFS[1:0]Bits (Digital Filter Select)

The setting of these bits enables or disables the digital filter and selects its sampling clock.

10.2.4 CAC Interrupt Request Enable Register (CAICR)

Address(es): 0008 B003h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	OVFFCL	MENDFCL	FERRFCL	—	OVFIE	MENDIE	FERRIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FERRIE	Frequency Error Interrupt Request Enable	0: Frequency error interrupt request is disabled. 1: Frequency error interrupt request is enabled.	R/W
b1	MENDIE	Measurement End Interrupt Request Enable	0: Measurement end interrupt request is disabled. 1: Measurement end interrupt request is enabled.	R/W
b2	OVFIE	Overflow Interrupt Request Enable	0: Overflow interrupt request is disabled. 1: Overflow interrupt request is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	FERRFCL	FERRF Clear	When 1 is written to this bit, the CASTR.FERRF flag is cleared. This bit is read as 0.	R/W
b5	MENDFCL	MENDF Clear	When 1 is written to this bit, the CASTR.MENDF flag is cleared. This bit is read as 0.	R/W
b6	OVFFCL	OVFF Clear	When 1 is written to this bit, the CASTR.OVFF flag is cleared. This bit is read as 0.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

FERRIE Bit (Frequency Error Interrupt Request Enable)

This bit specifies whether the frequency error interrupt request is enabled or disabled.

MENDIE Bit (Measurement End Interrupt Request Enable)

This bit specifies whether the measurement end interrupt request is enabled or disabled.

OVFIE Bit (Overflow Interrupt Request Enable)

This bit specifies whether the overflow interrupt request is enabled or disabled.

FERRFCL Bit (FERRF Clear)

Setting this bit to 1 clears the CASTR.FERRF flag.

MENDFCL Bit (MENDF Clear)

Setting this bit to 1 clears the CASTR.MENDF flag.

OVFFCL Bit (OVFF Clear)

Setting this bit to 1 clears the CASTR.OVFF flag.

10.2.5 CAC Status Register (CASTR)

Address(es): 0008 B004h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	OVFF	MENDF	FERRF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FERRF	Frequency Error Flag	0: The clock frequency is within the range corresponding to the settings. 1: The clock frequency has deviated beyond the range corresponding to the settings (frequency error).	R
b1	MENDF	Measurement End Flag	0: Measurement is in progress. 1: Measurement has ended.	R
b2	OVFF	Overflow Flag	0: The counter has not overflowed. 1: The counter has overflowed.	R
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FERRF Flag (Frequency Error Flag)

This flag indicates deviation of the clock frequency from the set value (frequency error).

[Setting condition]

- The clock frequency is outside of the setting range.

[Clearing condition]

- 1 is written to the CAICR.FERRFCL bit.

MENDF Flag (Measurement End Flag)

This flag indicates the end of measurement.

[Setting condition]

- Measurement has finished.

[Clearing condition]

- 1 is written to the CAICR.MENDFCL bit.

OVFF Flag (Overflow Flag)

This flag indicates that the counter has overflowed.

[Setting condition]

- The counter has overflowed.

[Clearing condition]

- 1 is written to the CAICR.OVFFCL bit.

10.2.6 CAC Upper-Limit Value Setting Register (CAULVR)

Address(es): 0008 B006h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CAULVR is a 16-bit readable/writable register that specifies the upper-limit value of the counter used for measuring the frequency. When the frequency rises above the value specified in this register, a frequency error is detected.
Write to this register when the CACR0.CFME bit is 0.
The counter value held in CACNTBR can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

10.2.7 CAC Lower-Limit Value Setting Register (CALLVR)

Address(es): 0008 B008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CALLVR is a 16-bit readable/writable register that specifies the lower-limit value of the counter used for measuring the frequency. When the frequency falls below the value specified in this register, a frequency error is detected.
Write to this register when the CACR0.CFME bit is 0.
The counter value held in CACNTBR can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

10.2.8 CAC Counter Buffer Register (CACNTBR)

Address(es): 0008 B00Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CACNTBR is a 16-bit read-only register that retains the counter value at the time a valid reference signal edge is input.

10.3 Operation

10.3.1 Measuring Clock Frequency

The clock frequency accuracy measurement circuit measures the clock frequency using the CACREF pin input or the internal clock as a reference. Figure 10.2 shows an operating example of the clock frequency accuracy measurement circuit.

The clock frequency accuracy measurement circuit operates as shown below when measuring the clock frequency.

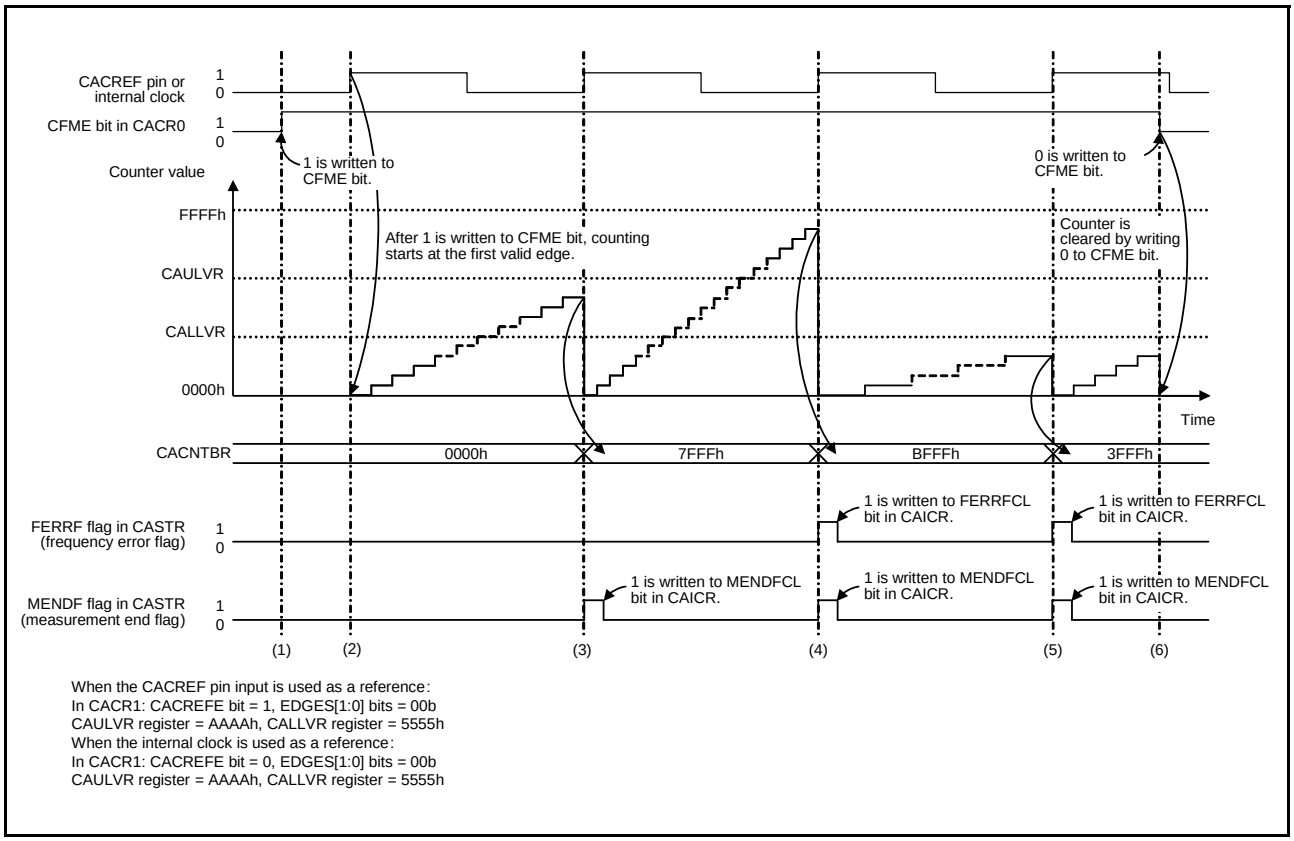


Figure 10.2 Operating Example of Clock Frequency Accuracy Measurement Circuit

(1) When the CACREF pin input is used as a reference (CACR1.CACREFE bit = 1), clock frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is 0 and the CACR1.CACREFE bit is 1. On the other hand, when the internal clock is used as a reference (CACR1.CACREFE bit = 0), clock frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is 1.

(2) When the CACREF pin input is used as a reference, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits is input from the CACREF pin after 1 is written to the CFME bit. The valid edge is a rising edge (CACR1.EDGES[1:0] = 00b) in Figure 10.2.
When the internal clock is used as a reference, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits is input based on the clock source selected by the CACR2.RSCS[2:0] bits after 1 is written to the CFME bit. The valid edge is a rising edge (CACR1.EDGES[1:0] = 00b) in Figure 10.2.

(3) When the next valid edge is input, the counter value is transferred in CACNTBR and compared with the values of CAULVR and CALLVR. If both CACNTBR ≤ CAULVR and CACNTBR ≥ CALLVR are satisfied, only the MENDF flag in CASTR is set to 1 because the clock frequency is correct. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.

(4) When the next valid edge is input, the counter value is transferred in CACNTBR and compared with the values of CAULVR and CALLVR. In the case of CACNTBR > CAULVR, the FERRF flag in CASTR is set to 1 because the

clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. Also, the MENDF flag in CASTR is set to 1. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.

(5) When the next valid edge is input, the counter value is transferred in CACNTBR and compared with the values of CAULVR and CALLVR. In the case of CACNTBR < CALLVR, the FERRF flag in CASTR is set to 1 because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. Also, the MENDF flag in CASTR is set to 1. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.

(6) While the CFME bit in CACR0 is 1, the counter value is transferred in CACNTBR and compared with the values of CAULVR and CALLVR every time a valid edge is input. Writing 0 to the CFME bit in CACR0 clears the counter and stops up-counting.

10.3.2 Digital Filtering of Signals on the CACREF Pin

The CACREF pin has a digital filter. Levels on the target pin for sampling are conveyed to the internal circuitry after matching three consecutive times at the selected sampling interval and the same level continues to be conveyed internally until the level on the pin again matches three consecutive times.

Enabling and disabling of the digital filter and its sampling clock are selectable.

The counter value transferred in CACNTBR may be in error by up to one cycle of the sampling clock due to the difference between the phases of the digital filter and the signal input to the CACREF pin.

When a frequency dividing clock is selected as a count source clock, the counter value error is obtained by the following formula:

Counter value error = (One cycle of the count source clock) / (One cycle of the sampling clock)

10.4 Interrupt Requests

The CAC generates three types of interrupt request: frequency error interrupt, measurement end interrupt, and overflow interrupt. When an interrupt source is generated, the corresponding status flag becomes 1. Table 10.3 lists details on the interrupt requests of the clock frequency accuracy measurement circuit.

Table 10.3 Interrupt Requests of Clock Frequency Accuracy Measurement Circuit

Interrupt Request	Interrupt Enable Bit	Status Flag	Interrupt Source
Frequency error interrupt	CAICR.FERRIE	CASTR.FERRF	The result of comparing CACNTBR to CAULVR and CALLVR is either CACNTBR > CAULVR or CACNTBR < CALLVR.
Measurement end interrupt	CAICR.MENDIE	CASTR.MENDF	A valid edge is input from the CACREF pin. Note however that a measurement end interrupt does not occur at the first valid edge after writing 1 to the CACR0.CFME bit.
Overflow interrupt	CAICR.OVFIE	CASTR.OVFF	The counter has overflowed.

10.5 Usage Notes

10.5.1 Module Stop Function Setting

CAC operation can be disabled or enabled using module stop control register C (MSTPCRC). The initial setting is for the CAC to be halted. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

11. Low Power Consumption

11.1 Overview

This MCU has several functions for reducing power consumption, including switching of clock signals to reduce power consumption, BCLK output control, SDCLK output control, stopping modules, functions for low power consumption in normal operation, and transitions to low power consumption states.

Table 11.1 lists the specifications of low power consumption functions, and Table 11.2 lists the conditions to shift to low power consumption modes, states of the CPU and peripheral modules, and the method for release from each mode. After a reset, this LSI enters the normal program execution state, but modules except for the DMAC, DTC, and RAM do not operate.

Table 11.1 Specifications of Low Power Consumption Functions

Item	Specification
Reducing power consumption by switching clock signals	The frequency division ratio is settable independently for the system clock (ICLK), peripheral module clock (PCLKA, PCLKB, PCLKC, PCLKD), external bus clock (BCLK), and flash interface clock (FCLK).*1
BCLK output control function	BCLK output or high-level output can be selected.*1
SDCLK output control function	SDCLK output or high-level output can be selected.*1
Module-stop function	Functions can be stopped independently for each peripheral module.
Function for transition to low power consumption mode	Transition to a low power consumption mode in which the CPU, peripheral modules, or oscillators are stopped is enabled.
Low power consumption modes	• Sleep mode • All-module clock stop mode • Software standby mode • Deep software standby mode
Function for lower operating power consumption	• Power consumption can be reduced in normal operation, sleep mode, and all-module clock stop mode by selecting an appropriate operating power consumption control mode according to the operating frequency and operating voltage range. • Three operating power control modes High-speed operating mode Low-speed operating mode 1 Low-speed operating mode 2

Note 1. For details, see section 9, Clock Generation Circuit.

Table 11.2 Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode

Entering and Exiting Low Power Consumption Modes and Operating States	Sleep Mode	All-Module Clock Stop Mode	Software Standby Mode	Deep Software Standby Mode
Transition condition	Control register + instruction	Control register + instruction	Control register + instruction	Control register + instruction
Method of release other than reset	Interrupt	Interrupt*1	Interrupt*2	Interrupt*3
State after release*4	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (reset processing)
Main clock oscillator	Operating possible	Operating possible	Operating possible*5	Operating possible*5
Sub-clock oscillator	Operating possible	Operating possible	Operating possible*6	Operating possible*6
High-speed on-chip oscillator	Operating possible	Operating possible	Stopped	Stopped
Low-speed on-chip oscillator	Operating possible	Operating possible	Stopped	Stopped
IWDT-dedicated on-chip oscillator	Operating possible*7	Operating possible*7	Operating possible*7	Stopped (Undefined)*7
PLL	Operating possible	Operating possible	Stopped	Stopped
CPU	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
RAM (ECC RAM included)	Operating possible (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
Standby RAM	Operating possible (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Retained/Undefined)*8
Flash memory	Operating	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
USBFS host/function module (USBb)	Operating possible	Stopped*9	Stopped*9	Stopped (Retained/Undefined)*10
USBFS host/function module (USBA)	Operating possible	Stopped*9	Stopped*9	Stopped (Retained/Undefined)*10
Watchdog timer (WDTA)	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
Independent watchdog timer (IWDT)	Operating possible*7	Operating possible*7	Operating possible*7	Stopped (Undefined)*7
Realtime clock (RTC)	Operating possible	Operating possible	Operating possible	Operating possible
8-bit timer (unit 0, unit 1) (TMR)	Operating possible	Operating possible*11	Stopped (Retained)	Stopped (Undefined)
Voltage detection circuit (LVDA)	Operating possible	Operating possible	Operating possible	Operating possible *12, *13
Power-on reset circuit	Operating	Operating	Operating	Operating*13
Peripheral modules	Operating possible	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
I/O ports	Operating	Retained*14	Retained*15	Retained*15

“Operating possible” means that operating or stopped can be controlled by the control register setting.
“Stopped (Retained)” means that internal register values are retained and internal operations are suspended.
“Stopped (Undefined)” means that internal register values are undefined and power is not supplied to the internal circuit.

Note 1. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ15) or any of peripheral interrupts (the 8-bit timer, RTC alarm, RTC periodic, IWDT, USB suspend/resume, voltage monitoring 1, voltage monitoring 2, and main-clock oscillation stop detection).
Note 2. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ15) or any of peripheral interrupts (the RTC alarm, RTC periodic, IWDT, USB suspend/resume, voltage monitoring 1, and voltage monitoring 2 interrupts).
Note 3. “Interrupts” here indicates a certain external pin interrupt source pin (the NMI, IRQ0-DS to IRQ15-DS, SCL2-DS, SDA2-DS, or CRX1-DS) or any of peripheral interrupts (the RTC alarm, RTC periodic, USB suspend/resume, voltage monitoring 1, and voltage monitoring 2 interrupts). However, these interrupts are enabled only when the corresponding bit in the deep standby interrupt enable registers i (DPSIERi) (i = 0 to 3) is set to 1. Pins that have “-DS” appended to their names can be used as triggers for release from deep software standby. USBA is not released from deep software standby mode in response to the change of DM or DP. Also, USBb is not released from deep software standby mode using USB0_OVRCURB multiplexed with pin P22.
Note 4. This does not include release initiated by the RES# pin reset, power-on reset, voltage monitoring reset, or independent watchdog-timer reset. The transition is to the reset state when release is initiated by one of these reset sources.
Note 5. Operation or stopping can be selected by the main clock oscillator forced oscillation bit (MOFXIN) in the main clock oscillator forced oscillation control register (MOFCR).
Note 6. Operation or stopping is selected by the sub-clock oscillator control bit (RTCEN) in the RTC control register 3 (RCR3).
Note 7. Operation or stopping is selected by the setting of the IWDT sleep mode count stop control bit (IWDTSLCSTP) in the option function select register 0 (OFS0) in IWDT auto start mode. If the OFS0.IWDTSLCSTP bit is 0 (disabling stopping of the counter when a transition to low power consumption mode is made), the transition is to software standby mode rather than deep software standby mode. In any mode other than IWDT auto start mode, operation or stopping is selected by the setting of the sleep mode counter stop control bit (SLCSTP) in the IWDT counter stop control register (IWDTCSTPR). If the

IWDTCSTPR.SLCSTP bit is 0 (disabling stopping of the counter when a transition to low power consumption mode is made), the transition is to software standby mode rather than deep software standby mode.

Note 8. Retention or undefined is selectable by the setting of the deep cut bits in the deep standby control register (DPSBYCR.DEEPCUT[1:0]).

Note 9. Detection of USB resumption is possible.

Note 10. Disabling or enabling of detection of USB resumption is controllable by the deep cut bits in the deep standby control register (DPSBYCR.DEEPCUT[1:0]). When detection of USB resumption is enabled, the values of the registers in the USB resume detecting unit are only held even in deep software standby mode.
However, USBA is not released from deep software standby mode in response to the change of DM or DP. Also, USBb is not released from deep software standby mode using USB0_OVRCURB multiplexed with pin P22.

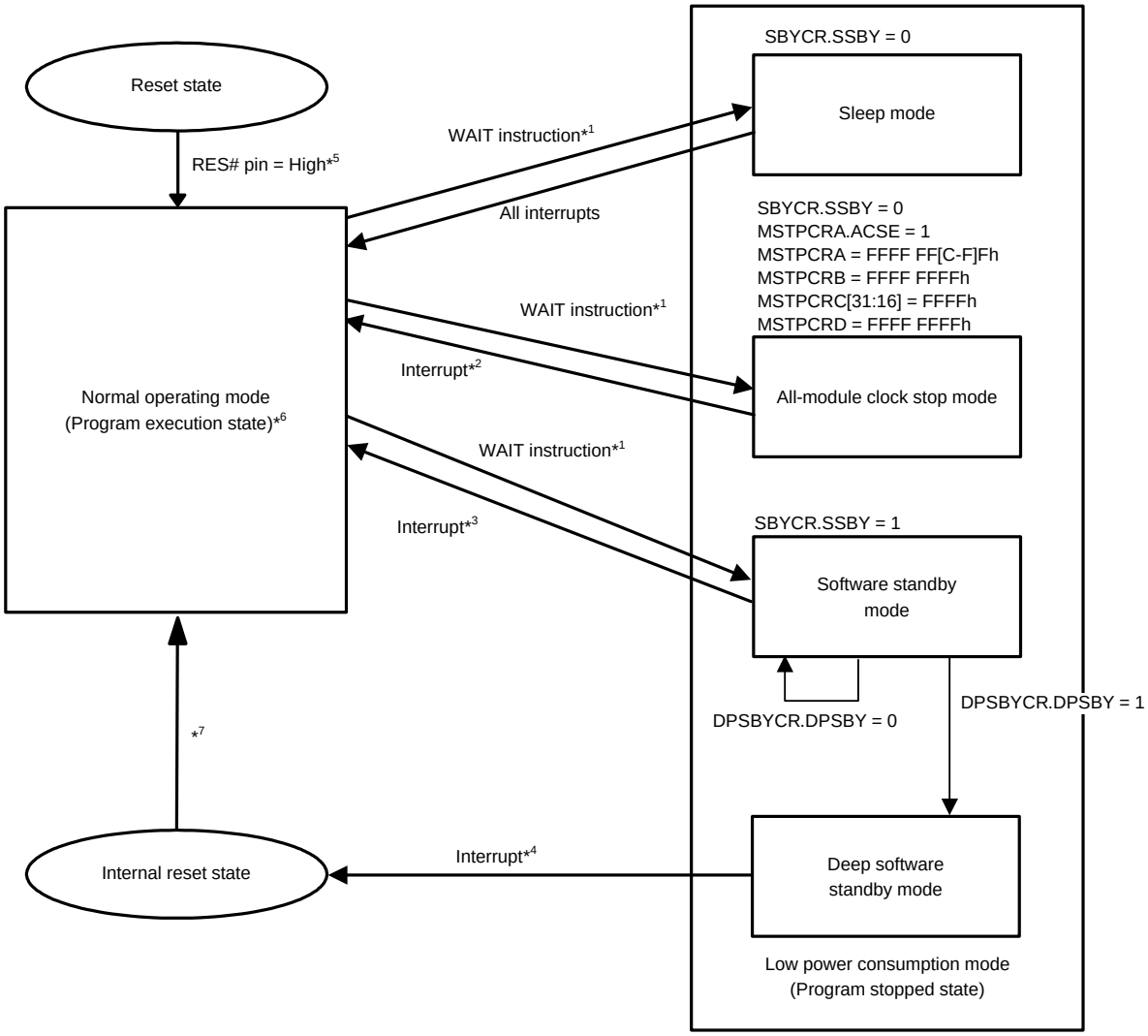
Note 11. Stopping or operation is controlled by the module-stop setting bits (MSTPA4 and MSTPA5, respectively) in module-stop control register A (MSTPCRA) for 8-bit timers 0 and 1 (unit 0) and 2 and 3 (unit 1).

Note 12. If the voltage monitoring 1 circuit mode selection bit in the voltage monitoring 1 circuit control register 0 (LVD1CR0.LVD1RI) or the voltage monitoring 2 circuit mode selection bit in the voltage monitoring 2 circuit control register 0 (LVD2CR0.LVD2RI) is 1, the transition is to software standby mode rather than deep software standby mode.

Note 13. When the deep cut bits in the deep standby control register (DPSBYCR.DEEPCUT[1:0]) are set to 11b and the LSI enters deep software standby mode, the voltage detection circuit stops and the low power consumption function of the power-on reset circuit is enabled.

Note 14. If pin P53 is being used for the BCLK signal, operation continues with as-is output of BCLK. While the 8-bit timer and RTC are operated, the related pins continue operation.

Note 15. Retention of levels or placement in the high-impedance state is selectable for the address bus and bus control signals (CS0# to CS7#, RD#, WR0# to WR3#, WR#, BC0# to BC3#, ALE, CKE, SDCS#, RAS#, CAS#, WE#, and DQM0 to DQM3) by the output port enable bit (OPE) in the standby control register (SBYCR).



Note 1. When an interrupt that acts as a trigger for release is received during the transition to the program-stopped state after the execution of a WAIT instruction, the transition to the program stopped state does not proceed and interrupt exception processing is executed instead.

Note 2. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ15) or any of peripheral interrupts (the 8-bit timer, RTC alarm, RTC periodic, IWDT, USB suspend/resume, voltage-monitoring 1, voltage-monitoring 2, and oscillator-stopped detection interrupts). However, a, 8-bit timer interrupt is only effective if the value of the corresponding module-stop setting bit (MSTPA4 or MSTPA5, respectively) in module-stop control register A (MSTPCRA) for 8-bit timer 0 and 1 (unit 0) or 2 and 3 (unit 1) is 0.

Note 3. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ15) or any of peripheral interrupts (the RTC alarm, RTC periodic, IWDT, USB suspend/resume, voltage-monitoring 1, and voltage-monitoring 2 interrupts).

Note 4. “Interrupts” here indicates a certain external pin interrupt source pin (the NMI, IRQ0-DS to IRQ15-DS, SCL2-DS, SDA2-DS, or CRX1-DS) or any of peripheral interrupts (the RTC alarm, RTC periodic, USB suspend/resume, voltage-monitoring 1, and voltage-monitoring 2 interrupts). However, these interrupts are enabled only when the corresponding bit in the deep standby interrupt enable registers i (DPSIERi) (i = 0 to 3) is set to 1. USBA is not released from deep software standby mode in response to the change of DM or DP. Also, USBb is not released from deep software standby mode using USB0_OVRCURB multiplexed with pin P22.

Note 5. The LOCO is the source of the operating clock following a transition from the reset state to normal operating mode.

Note 6. The transition to normal operating mode is made due to an interrupt from sleep mode, all-module clock-stop mode, or software standby mode. In the case of recovery from sleep mode, the clock source for subsequent use is selectable. For details, refer to section 11.2.7, Sleep Mode Return Clock Source Switching Register (RSTCKCR). Also in the case of recovery from software standby mode, the clock source will be the same before and at the time of the transition.

Note 7. When an interrupt source as indicated in Note 4 above is generated, an internal reset (deep software standby reset) is generated over a fixed period. Release from deep software standby mode accompanies release from the internal reset state. The transition from deep software standby mode is to normal operating mode, and reset exception processing starts with the LOCO as the source of the operating clock.

When the RES# pin reset, power-on reset, voltage-monitoring 0 reset, voltage-monitoring 1 reset, voltage-monitoring 2 reset, watchdog timer reset, independent watchdog timer reset, or software reset is generated in any state, a transition to the reset state is made. The voltage-monitoring 1 reset and voltage-monitoring 2 reset cannot be used for release from deep software standby mode.

Figure 11.1 Mode Transitions

11.2 Register Descriptions

11.2.1 Standby Control Register (SBYCR)

Address(es): 0008 000Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SSBY	OPE	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	OPE	Output Port Enable	0: In software standby mode or deep software standby mode, the address bus and bus control signals are set to the high-impedance state. 1: In software standby mode or deep software standby mode, the address bus and bus control signals retain the output state.	R/W
b15	SSBY	Software Standby	0: Shifts to sleep mode or all-module clock stop mode after the WAIT instruction is executed 1: Shifts to software standby mode after the WAIT instruction is executed	R/W

OPE Bit (Output Port Enable)

The OPE bit specifies whether to retain the output of the address bus and bus control signals (CS0# to CS7#, RD#, WR0# to WR3#, WR#, BC0# to BC3#, ALE, CKE, SDCS#, RAS#, CAS#, WE#, and DQM0 to DQM3) in software standby mode or deep software standby mode, or to set the output to the high-impedance state.

SSBY Bit (Software Standby)

The SSBY bit specifies the transition destination after the WAIT instruction is executed.

When the SSBY bit is set to 1, the LSI enters software standby mode after execution of the WAIT instruction. When the LSI returns to normal operating mode after an interrupt has initiated release from software standby mode, the SSBY bit remains 1. Write 0 to this bit to clear it.

When the oscillation stop detection function enable bit in the oscillation stop detection control register (OSTDCR.OSTDE) is 1, the setting of the SSBY bit is ineffective. Even if the SSBY bit is 1, the LSI will enter sleep mode or all module clock stop mode on execution of the WAIT instruction.

When the code flash P/E mode entry bit in the flash P/E mode entry register (FENTRYR.FENTRYC) is 1 or the data flash P/E mode entry bit (FENTRYR.FENTRYD) is 1, the setting of this bit is ineffective. Sleep mode is entered on execution of the WAIT instruction even if this bit has been set to 1.

11.2.2 Module Stop Control Register A (MSTPCRA)

Address(es): 0008 0010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ACSE	—	MSTPA 29	MSTPA 28	MSTPA 27	—	—	MSTPA 24	—	—	—	—	MSTPA 19	—	MSTPA 17	MSTPA 16
Value after reset:	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSTPA 15	MSTPA 14	MSTPA 13	—	MSTPA 11	MSTPA 10	MSTPA 9	—	MSTPA 7	—	MSTPA 5	MSTPA 4	—	—	MSTPA 1	MSTPA 0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	MSTPA0	Compare Match Timer W (Unit 1) Module Stop	Target module: CMTW unit 1 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b1	MSTPA1	Compare Match Timer W (Unit 0) Module Stop	Target module: CMTW unit 0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b3, b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b4	MSTPA4	8-Bit Timer 3/2 (Unit 1) Module Stop	Target module: TMR3/TMR2 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b5	MSTPA5	8-Bit Timer 1/0 (Unit 0) Module Stop	Target module: TMR1/TMR0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	MSTPA7	General PWM Timer	Target module: GPTA 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b8	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b9	MSTPA9	Multifunction Timer Pulse Unit 3 Module Stop	Target module: MTU3 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b10	MSTPA10	Programmable Pulse Generator (Unit 1) Module Stop	Target module: PPG1 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b11	MSTPA11	Programmable Pulse Generator (Unit 0) Module Stop	Target module: PPG0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b12	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b13	MSTPA13	16-Bit Timer Pulse Unit 0 (Unit 0) Module Stop	Target module: TPU unit 0 (TPU0 to TPU5) 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b14	MSTPA14	Compare Match Timer (Unit 1) Module Stop	Target module: CMT unit 1 (CMT2, CMT3) 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b15	MSTPA15	Compare Match Timer (Unit 0) Module Stop	Target module: CMT unit 0 (CMT0, CMT1) 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b16	MSTPA16	12-bit A/D Converter (Unit 1) Module Stop	Target module: S12AD unit 1 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W

Bit	Symbol	Bit Name	Description	R/W
b17	MSTPA17	12-bit A/D Converter (Unit 0) Module Stop	Target module: S12AD unit 0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b18	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b19	MSTPA19	12-bit D/A Converter Module Stop	Target module: 12-bit D/A 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b23 to b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b24	MSTPA24	Module Stop A24	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b26, b25	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b27	MSTPA27	Module Stop A27	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b28	MSTPA28	DMA Controller/Data Transfer Controller Module Stop	Target module: DMAC/DTC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b29	MSTPA29	EXDMA Controller Module Stop	Target module: EXDMAC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b30	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b31	ACSE	All-Module Clock Stop Mode Enable	0: All-module clock stop mode is disabled 1: All-module clock stop mode is enabled	R/W

ACSE Bit (All-Module Clock Stop Mode Enable)

The ACSE bit enables or disables a transition to all-module clock stop mode.

With the ACSE bit set to 1, when the CPU executes the WAIT instruction with the SBYCR.SSBY bit, MSTPCRA, MSTPCRB, MSTPCRC, and MSTPCRD satisfying specified conditions, the LSI enters all-module clock stop mode. For details, see section 11.6.2, All-Module Clock Stop Mode.

Whether to stop the 8-bit timers or not can be selected by the MSTPA5 and MSTPA4 bits.

When the MSTPCRA.ACSE bit = 0 while the SBYCR.SSBY = 0, a transition to sleep mode is made after the WAIT instruction is executed.

When the code flash P/E mode entry bit in the flash P/E mode entry register (FENTRYR.FENTRYC) is 1 or the data flash P/E mode entry bit (FENTRYR.FENTRYD) is 1, the setting of this bit is ineffective. Sleep mode is entered on execution of the WAIT instruction if this bit has been set to 1.

11.2.3 Module Stop Control Register B (MSTPCRB)

Address(es): 0008 0014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSTPB31	MSTPB30	MSTPB29	MSTPB28	MSTPB27	MSTPB26	MSTPB25	MSTPB24	MSTPB23	MSTPB22	MSTPB21	—	MSTPB19	—	MSTPB17	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSTPB15	MSTPB14	—	MSTPB12	—	—	MSTPB9	MSTPB8	—	MSTPB6	—	MSTPB4	—	MSTPB2	MSTPB1	MSTPB0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	MSTPB0	CAN Module 0 Module Stop*1	Target module: CAN0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b1	MSTPB1	CAN Module 1 Module Stop*1	Target module: CAN1 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b2	MSTPB2	CAN Module 2 Module Stop*1	Target module: CAN2 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b4	MSTPB4	Serial Communication Interface SCIh Module Stop	Target module: SCIh (SCI12) 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b5	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b6	MSTPB6	Data Operation Circuit Module Stop	Target module: DOC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b8	MSTPB8	Temperature Sensor Module Stop	Target module: Temperature sensor 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b9	MSTPB9	Event Link Controller Module Stop	Target module: ELC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b11, b10	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b12	MSTPB12	Universal Serial Bus 2.0 FS Interface Module Stop*2	Target module: USBA 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b13	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b14	MSTPB14 *3	Ethernet Controller and Ethernet Controller DMA Controller (Channel 1) Modules Stop	Target modules: ETHER and EDMAC (channel 1) 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b15	MSTPB15 *3	Ethernet Controller and Ethernet Controller DMA Controller (Channel 0) Modules Stop	Target modules: ETHER and EDMAC (channel 0) 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b16	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b17	MSTPB17	Serial Peripheral Interface 0 Module Stop	Target module: RSPI0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b18	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

Bit	Symbol	Bit Name	Description	R/W
b19	MSTPB19	Universal Serial Bus 2.0 FS Interface Module Stop*2	Target module: USB0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b20	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b21	MSTPB21	I^2C Bus Interface 0 Module Stop	Target module: RIIC0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b22	MSTPB22	Parallel Data Capture Unit Module Stop	Target module: PDC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b23	MSTPB23	CRC Calculator Module Stop	Target module: CRC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b24	MSTPB24	Serial Communication Interface 7 Module Stop	Target module: SCI7 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b25	MSTPB25	Serial Communication Interface 6 Module Stop	Target module: SCI6 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b26	MSTPB26	Serial Communication Interface 5 Module Stop	Target module: SCI5 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b27	MSTPB27	Serial Communication Interface 4 Module Stop	Target module: SCI4 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b28	MSTPB28	Serial Communication Interface 3 Module Stop	Target module: SCI3 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b29	MSTPB29	Serial Communication Interface 2 Module Stop	Target module: SCI2 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b30	MSTPB30	Serial Communication Interface 1 Module Stop	Target module: SCI1 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b31	MSTPB31	Serial Communication Interface 0 Module Stop	Target module: SCI0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W

Note 1. The MSTPBi bit should be rewritten while the oscillation of the clock controlled by the MSTPBi bit is stabilized. For entering software standby mode after writing a new value to the MSTPBi bit, wait for two cycles of the CAN clock (CANMCLK) to elapse after writing the new value, and then execute a WAIT instruction (i = 0 to 2).

Note 2. For entering software standby mode after writing a new value to the MSTPB19 bit, wait for two cycles of the USBb clock (UCLK) to elapse after writing the new value, and then execute a WAIT instruction. For entering software standby mode after writing a new value to the MSTPB12 bit, wait for two cycles of the USBA clock (UCLK) to elapse after writing the new value, and then execute a WAIT instruction, if UCLK is to be used as the operating clock for the USBA module by setting the PHYSET.HSEB bit to 1.

Note 3. Clearing either of the MSTPB15 and MSTPB14 bits initiates supply of the clock signal to the corresponding Ethernet controller (EPTPC) and EDMAC (DMAC for the Ethernet controller). However, some internal registers of the EPTPC are inaccessible unless the corresponding bit is cleared. For details, see section 36.6.1, Restrictions on Access to Registers. If either of the MSTPB15 and MSTPB14 bits is cleared, set the timeout detection enable bit in the bus error monitoring enable register (BEREN.TOEN) to 1. For details of bus timeout, see section 16.7.1.2, Timeout.

11.2.4 Module Stop Control Register C (MSTPCRC)

Address(es): 0008 0018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	MSTPC27	MSTPC26	MSTPC25	MSTPC24	MSTPC23	—	—	—	MSTPC19	—	MSTPC17	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	MSTPC7	MSTPC6	—	—	—	—	—	MSTPC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MSTPC0	RAM0 Module Stop*1	Target module: RAM0 (0000 0000h to 0007 FFFFh) 0: RAM0 operating 1: RAM0 stopped	R/W
b5 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	MSTPC6	ECCRAM Module Stop*2	Target module: ECCRAM 0: ECCRAM operating 1: ECCRAM stopped	R/W
b7	MSTPC7	Standby RAM Module Stop*3	Target module: Standby RAM 0: Standby RAM operating 1: Standby RAM stopped	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b17	MSTPC17	I^2C Bus Interface 2 Module Stop	Target module: RIIC2 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b18	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b19	MSTPC19 *4	CAC Module Stop	Target module: CAC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b22 to b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b23	MSTPC23	Quad Serial Peripheral Interface Module Stop	Target module: QSPI 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b24	MSTPC24	FIFO On-chip Serial Communications Interface 11 Module Stop	Target module: SCIF11 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b25	MSTPC25	FIFO On-chip Serial Communications Interface 10 Module Stop	Target module: SCIF10 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b26	MSTPC26	FIFO On-chip Serial Communications Interface 9 Module Stop	Target module: SCIF9 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b27	MSTPC27	FIFO On-chip Serial Communications Interface 8 Module Stop	Target module: SCIF8 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b31 to b28	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. The MSTPC0 bit should not be set to 1 during access to the corresponding RAM. The corresponding RAM should not be accessed while the MSTPC0 bit is set to 1.

Note 2. The MSTPC6 bit should not be set to 1 during access to the ECCRAM. The ECCRAM should not be accessed while the MSTPC6 bit is set to 1.

Note 3. The MSTPC7 bit should not be set to 1 during access to the standby RAM. The standby RAM should not be accessed while the MSTPC7 bit is set to 1.

Note 4. The MSTPC19 bit should be rewritten while the oscillation of the clock to be controlled by this bit is stable. For entering software standby mode after writing a new value to this bit, wait for two cycles of the slowest clock among the clocks output by the oscillators actually oscillating, and then execute a WAIT instruction.

11.2.5 Module Stop Control Register D (MSTPCRD)

Address(es): 0008 001Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	MSTPD23	—	MSTPD21	—	MSTPD19	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSTPD15	MSTPD14	—	—	—	—	—	—	MSTPD7	MSTPD6	MSTPD5	MSTPD4	MSTPD3	MSTPD2	MSTPD1	MSTPD0
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MSTPD0	Module Stop D0	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b1	MSTPD1	Module Stop D1	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b2	MSTPD2	Module Stop D2	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b3	MSTPD3	Module Stop D3	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b4	MSTPD4	Module Stop D4	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b5	MSTPD5	Module Stop D5	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b6	MSTPD6	Module Stop D6	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b7	MSTPD7	Module Stop D7	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b13 to b8	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b14	MSTPD14	Serial Sound Interface 1 Module Stop	Target module: SSI1 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b15	MSTPD15	Serial Sound Interface 0 Module Stop	Target module: SSI0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b18 to b16	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b19	MSTPD19	SD Host Interface Module Stop	Target module: SDHI 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b20	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b21	MSTPD21	MMC Host Interface Module Stop	Target module: MMCIF 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b22	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b23	MSTPD23	Sampling Rate Converter Module Stop	Target module: SRC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W

Bit	Symbol	Bit Name	Description	R/W
b31 to b24	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

11.2.6 Operating Power Control Register (OPCCR)

Address(es): 0008 00A0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OPCMTSF	—	OPCM[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	OPCM[2:0]	Operating Power Control Mode Select	b2 b0 0 0 0: High-speed operating mode 1 1 0: Low-speed operating mode 1 1 1 1: Low-speed operating mode 2 Settings other than above are prohibited.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	OPCMTSF	Operating Power Control Mode Transition Status Flag	• Read 0: Transition completed 1: During transition • Write The write value should be 0.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Register OPCCR is used to reduce power consumption in normal operating mode, sleep mode, and all-module clock stop mode. Power consumption can be reduced according to the operating frequency and operating voltage to be used by the OPCCR setting.

OPCCR should not be modified in the following cases:

- When the operating power control mode transition status flag (OPCMTSF) is 1 (operating power control mode switching is in progress)
- When the code flash P/E mode entry bit (FENTRYR.FENTRYC) in the flash P/E mode entry register is 1 or the data flash P/E mode entry bit (FENTRYR.FENTRYD) is 1
- Period from the time of WAIT instruction execution for a sleep mode transition, to return from sleep mode to normal operation

For the procedure to use in shifting to operating power control mode, refer to section 11.5, Function for Lower Operating Power Consumption.

On return from software standby, the chip enters the high-speed operating mode. Even if a WAIT instruction is executed, if release from software standby precedes completion of the transition, the mode remains the same as before execution of the WAIT instruction. If this creates a problem, set the OPCCR.OPCM[2:0] bits to 000b during processing of the return interrupt.

OPCM[2:0] Bits (Operating Power Control Mode Select)

The OPCM[2:0] bits select operating power control mode in normal operating mode, sleep mode, and all-module clock stop mode.

Table 11.3 shows the operating power control modes along with the operating frequency ranges, operating voltage ranges, and power consumption.

Table 11.3 Relationship between Operating Power Control Mode, Operating Range, and Power Consumption

Operating Power Control Mode	OPCM [2:0] Bits	Operating Frequency Range							Operating Voltage Range		Power Consumption
		Reading Flash Memory						Programming or Erasing Flash Memory	Reading Flash Memory	Programming or Erasing Flash Memory	
		ICLK	FCLK	PCLKA	PCLKB	PCLKC PCLKD	BCLK	FCLK			
High-speed operating mode	000b	120 MHz max	60 MHz max	120 MHz max	60 MHz max	60 MHz max*1	120 MHz max	4 MHz to 60 MHz	2.7 V to 3.6 V	2.7 V to 3.6 V	High
Low-speed operating mode 1	110b	1 MHz max	1 MHz max	1 MHz max	1 MHz max	1 MHz max*1	1 MHz max	P/E disabled	2.7 V to 3.6 V	P/E disabled	↓
Low-speed operating mode 2	111b	32 kHz to 264 kHz	32 kHz to 264 kHz	264 kHz max	264 kHz max	264 kHz max*1	264 kHz max	P/E disabled	2.7 V to 3.6 V	P/E disabled	Low

Note 1. When the 12-bit A/D converter is used in high-speed operating mode or low-speed operating mode 1, the frequency must be set to at least 1 MHz. The 12-bit A/D converter cannot be used in low-speed operating mode 2.

Each operating power control mode is described below.

- High-speed operating mode

This mode allows high-speed operation.
During reading the flash memory (FLASH), the maximum operating frequency of ICLK, PCLKA, and BCLK is 120 MHz, and that of FCLK and PCLKB is 60 MHz. The S12AD conversion clocks PCLKC and PCLKD can be operated in the operating frequency from 1 MHz to 60 MHz. During Flash memory programming/erasure (P/E), the FCLK can be operated in the operating frequency from 4 MHz to 60 MHz. The operating voltage is in the range of 2.7 to 3.6 V both for Flash memory read and P/E. After release from a reset, this LSI is activated in this mode.

- Low-speed operating mode 1

This mode reduces power consumption for low-speed operation.
During reading the flash memory (FLASH), the maximum operating frequency of ICLK, FCLK, PCLKA, PCLKB and BCLK is 1 MHz. The operating voltage is in the range of 2.7 to 3.6 V.
In low-speed operating mode 1, P/E operation of Flash memory is disabled, and writing to set the PLLCR2.PLLEN bit to 0 (PLL operation) is prohibited.
In this mode, lower power consumption is possible than in high-speed operating mode when the same operation is performed under the same conditions (operating frequency, operating voltage).

- Low-speed operating mode 2

As compare to low-speed operating mode 1, this mode reduces power consumption for low-speed operation.
During reading the flash memory (FLASH), the maximum operating frequency of ICLK, FCLK, PCLKA, PCLKB and BCLK is 264 kHz, and the minimum operating frequency of ICLK and FCLK is 32 kHz. The operating voltage is in the range of 2.7 to 3.6 V.
The following restrictions apply when low-speed operating mode 2 is selected:

- P/E operations for flash memory are prohibited.
- Reading of data flash is prohibited.
- Using the PLL or HOCO is prohibited.
- Using the oscillation stop detection function of the main clock oscillator is prohibited.

In this mode, lower power consumption is possible than in low-speed operating mode 1 when the same operation is performed under the same conditions (operating frequency, operating voltage).

When the clock source select bits in the system clock control register 3 (SCKCR3.CKSEL[2:0]) are 011b (sub-clock oscillator selected) and the system clock select bits (ICK[3:0]) or Flash-IF clock select bits (FCK[3:0]) in the system clock control register (SCKCR) are not 0000b (no frequency dividing), OPCM[2:0] bits cannot be set to 111b.
When the PLL stop control bit (PLLCR2.PLLEN) in PLL control register 2 is 0 (PLL operation), writing 110b (low-speed operating mode 1) and 111b (low-speed operating mode 2) to the OPCM[2:0] bits is not possible.

OPCMTSF Flag (Operating Power Control Mode Transition Status Flag)

The OPCMTSF flag indicates the switching control state when the operating power control mode is switched.
When a write access is attempted to change the operating power control mode, the OPCMTSF flag is set to 1. The flag becomes 0 after a transition to the changed control mode is completed. Make sure that the OPCMTSF flag is 0 (completed operating power control mode transition) before the next processing.

11.2.7 Sleep Mode Return Clock Source Switching Register (RSTCKCR)

Address(es): 0008 00A1h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTCKEN	—	—	—	—	RSTCKSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RSTCKSEL[2:0]	Sleep Mode Return Clock Source Select	b2 b0 0 0 1: HOCO is selected 0 1 0: Main clock oscillator is selected Settings other than above are prohibited while the RSTCKEN bit is 1.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	RSTCKEN	Sleep Mode Return Clock Source Switching Enable	0: Clock source switching on release from sleep mode is disabled 1: Clock source switching on release from sleep mode is enabled	R/W

Register RSTCKCR is used to control clock source switching at the time of release from sleep mode.
When operation is restored from sleep mode by setting RSTCKCR, the main clock oscillator stop bit in the main clock oscillator control register (MOSCCR.MOSTP) and HOCO stop bit in the high-speed on-chip oscillator control register (HOCOCR.HCSTP) corresponding to the clock source to be used on restoration are automatically modified to the operating state. The value of RSTCKSEL[2:0] bits is automatically reloaded to the clock source select bits in the system clock control register 3 (SCKCR3.CKSEL[2:0]).
The sleep mode return clock source switching function and clock source switching function by the ELC cannot be used at the same time. To enable the sleep mode return clock source switching function, write 1 to the RSTCKCR.RSTCKEN bit with the ELC clock source switching function disabled. The ELC clock source switching function should be enabled with the RSTCKCR.RSTCKEN bit being 0.
When the setting of register RSTCKCR is for the HOCO to be used in recovery from sleep mode, the power supply for the HOCO is not automatically switched on. If the HOCO to be used in recovery from sleep mode, the power supply for the HOCO must be on when the transition to sleep mode takes place.
When return from sleep mode is made while clock source switching on release from sleep mode is enabled (RSTCKCR.RSTCKEN is 1), and operating power control mode select bits (OPCCR.OPCM[2:0]) are set so as to select low-speed operating mode 1 (110b) or low-speed operating mode 2 (111b), the OPCCR.OPCM[2:0] bits are automatically switched to high-speed mode (000b).

RSTCKSEL[2:0] Bits (Sleep Mode Return Clock Source Select)

The RSTCKSEL[2:0] bits select the clock source to be used at the time of release from sleep mode.
The clock source selected by the RSTCKSEL[2:0] bits is enabled only when the RSTCKEN bit is 1.

RSTCKEN Bit (Sleep Mode Return Clock Source Switching Enable)

The RSTCKEN bit enables or disables clock source switching at the time of release from sleep mode.
On release from sleep mode, the clock source should be switched only when LOCO or sub clock is selected as a clock for a transition to sleep mode. To make a transition to sleep mode with HOCO, main clock, or PLL selected as the clock source, the RSTCKEN bit should not be set to 1.

11.2.8 Deep Standby Control Register (DPSBYCR)

Address(es): 0008 C280h

	b7	b6	b5	b4	b3	b2	b1	b0
	DPSBY	IOKEEP	—	—	—	—	DEEPCUT[1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DEEPCUT [1:0]	Deep Cut	b1 b0 0 0: Power is supplied to the standby RAM and USB resume detecting unit in deep software standby mode 0 1: Power is not supplied to the standby RAM and USB resume detecting unit in deep software standby mode 1 0: Setting prohibited 1 1: Power is not supplied to the standby RAM and USB resume detecting unit in deep software standby mode. In addition, LVD is stopped and the low power consumption function in a power-on reset circuit is enabled.	R/W
b5 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	IOKEEP	I/O Port Retention	0: Release from deep software standby mode and cancellation of I/O port retention proceed simultaneously. 1: The I/O port state is retained even after release from deep software standby mode. Then, writing 0 to the IOKEEP bit cancels the I/O port retention.	R/W
b7	DPSBY	Deep Software Standby	SSBY b7 0 0: Transition to sleep mode or all-module clock stop mode is made after the WAIT instruction is executed 0 1: Transition to sleep mode or all-module clock stop mode is made after the WAIT instruction is executed 1 0: Transition to software standby mode is made after the WAIT instruction is executed 1 1: Transition to deep software standby mode is made after the WAIT instruction is executed	R/W

Register DPSBYCR is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

DEEPCUT[1:0] Bits (Deep Cut)

The DEEPCUT[1:0] bits control the internal power supply to the standby RAM and USB resume detecting unit in deep software standby mode. In addition, these bits control the state of LVD and power-on reset circuit in deep software standby mode.

The internal power supply of standby RAM and USB resume detecting unit can be controlled by the setting of the DEEPCUT[1:0] bits.

When a USB suspend/resume interrupt*1 is used as a deep software standby mode releasing source, the DEEPCUT[1:0] bits must be set to 00b.

When the LVD is used in deep software standby mode, the DEEPCUT[1:0] bits must be set to 00b or 01b.

For lower power consumption, set the DEEPCUT[1:0] bits to 11b so that the LVD is stopped and the low power consumption function of the power-on reset circuit is enabled.

Note 1. USBA is not released from deep software standby mode in response to the change of DM or DP. Also, USBb is not released from deep software standby mode using USB0_OVRCURB multiplexed with pin P22.

IOKEEP Bit (I/O Port Retention)

In deep software standby mode, I/O ports keep retaining the same states from software standby mode. The IOKEEP bit specifies whether to keep retaining the I/O port states from deep software standby mode even after release from deep software standby mode, or to cancel retention of the I/O port states.

DPSBY Bit (Deep Software Standby)

The DPSBY bit controls transitions to deep software standby mode.
When the WAIT instruction is executed while the SBYCR.SSBY and DPSBY bits are both 1, the LSI enters deep software standby mode through software standby mode.
The DPSBY bit remains 1 when release from deep software standby mode is triggered by certain pins which are sources of external pin interrupts (NMI, IRQ0-DS to IRQ15-DS, SCL2-DS, SDA2-DS, and CRX1-DS) or a peripheral interrupt (RTC alarm, RTC periodic, USB suspend/resume*1, voltage monitoring 1, or voltage monitoring 2). Write 0 to this bit to clear it.
The setting of the DPSBY bit becomes invalid when the IWDT is in auto-start mode and the OFS0.IWDTSLCSTP is 0 (counting continues) or the IWDT is in register start mode and the SLCSTP bit in IWDTCSTPR is 0.
Instead, even when the SBYCR.SSBY bit is 1 and the DPSBY bit 1, the transition after the execution of a WAIT instruction is to software standby mode.
The setting of the DPSBY bit becomes invalid when voltage monitoring 1 reset is enabled by the voltage monitoring 1 circuit mode select bit (LVD1CR0.LVD1RI = 1) or when a voltage monitoring 2 reset is enabled by the voltage monitoring 2 circuit mode bit (LVD2CR0.LVD2RI = 1). In this case, even when the SBYCR.SSBY bit is 1 and the DPSBY bit is 1, the transition after the execution of a WAIT instruction is to software standby mode.

Note 1. USBA is not released from deep software standby mode in response to the change of DM or DP. Also, USBb is not released from deep software standby mode using USB0_OVRCURB multiplexed with pin P22.

11.2.9 Deep Standby Interrupt Enable Register 0 (DPSIER0)

Address(es): 0008 C282h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ7E	DIRQ6E	DIRQ5E	DIRQ4E	DIRQ3E	DIRQ2E	DIRQ1E	DIRQ0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0E	IRQ0-DS Pin Enable	0: Release from deep software standby mode by the IRQ0-DS pin is disabled 1: Release from deep software standby mode by the IRQ0-DS pin is enabled	R/W
b1	DIRQ1E	IRQ1-DS Pin Enable	0: Release from deep software standby mode by the IRQ1-DS pin is disabled 1: Release from deep software standby mode by the IRQ1-DS pin is enabled	R/W
b2	DIRQ2E	IRQ2-DS Pin Enable	0: Release from deep software standby mode by the IRQ2-DS pin is disabled 1: Release from deep software standby mode by the IRQ2-DS pin is enabled	R/W
b3	DIRQ3E	IRQ3-DS Pin Enable	0: Release from deep software standby mode by the IRQ3-DS pin is disabled 1: Release from deep software standby mode by the IRQ3-DS pin is enabled	R/W
b4	DIRQ4E	IRQ4-DS Pin Enable	0: Release from deep software standby mode by the IRQ4-DS pin is disabled 1: Release from deep software standby mode by the IRQ4-DS pin is enabled	R/W
b5	DIRQ5E	IRQ5-DS Pin Enable	0: Release from deep software standby mode by the IRQ5-DS pin is disabled 1: Release from deep software standby mode by the IRQ5-DS pin is enabled	R/W
b6	DIRQ6E	IRQ6-DS Pin Enable	0: Release from deep software standby mode by the IRQ6-DS pin is disabled 1: Release from deep software standby mode by the IRQ6-DS pin is enabled	R/W
b7	DIRQ7E	IRQ7-DS Pin Enable	0: Release from deep software standby mode by the IRQ7-DS pin is disabled 1: Release from deep software standby mode by the IRQ7-DS pin is enabled	R/W

Register DPSIER0 is not initialized by the internal reset signal used as deep software standby mode releasing source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

After the setting of DPSIER0 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR0 being set to 1. Therefore, DPSIFR0 should be set to 0 before a transition to deep software standby mode.

11.2.10 Deep Standby Interrupt Enable Register 1 (DPSIER1)

Address(es): 0008 C283h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ15E	DIRQ14E	DIRQ13E	DIRQ12E	DIRQ11E	DIRQ10E	DIRQ9E	DIRQ8E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ8E	IRQ8-DS Pin Enable	0: Release from deep software standby mode by the IRQ8-DS pin is disabled 1: Release from deep software standby mode by the IRQ8-DS pin is enabled	R/W
b1	DIRQ9E	IRQ9-DS Pin Enable	0: Release from deep software standby mode by the IRQ9-DS pin is disabled 1: Release from deep software standby mode by the IRQ9-DS pin is enabled	R/W
b2	DIRQ10E	IRQ10-DS Pin Enable	0: Release from deep software standby mode by the IRQ10-DS pin is disabled 1: Release from deep software standby mode by the IRQ10-DS pin is enabled	R/W
b3	DIRQ11E	IRQ11-DS Pin Enable	0: Release from deep software standby mode by the IRQ11-DS pin is disabled 1: Release from deep software standby mode by the IRQ11-DS pin is enabled	R/W
b4	DIRQ12E	IRQ12-DS Pin Enable	0: Release from deep software standby mode by the IRQ12-DS pin is disabled 1: Release from deep software standby mode by the IRQ12-DS pin is enabled	R/W
b5	DIRQ13E	IRQ13-DS Pin Enable	0: Release from deep software standby mode by the IRQ13-DS pin is disabled 1: Release from deep software standby mode by the IRQ13-DS pin is enabled	R/W
b6	DIRQ14E	IRQ14-DS Pin Enable	0: Release from deep software standby mode by the IRQ14-DS pin is disabled 1: Release from deep software standby mode by the IRQ14-DS pin is enabled	R/W
b7	DIRQ15E	IRQ15-DS Pin Enable	0: Release from deep software standby mode by the IRQ15-DS pin is disabled 1: Release from deep software standby mode by the IRQ15-DS pin is enabled	R/W

Register DPSIER1 is not initialized by the internal reset signal used as deep software standby mode releasing source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

After the setting of DPSIER1 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR1 being set to 1. Therefore, DPSIFR1 should be set to 0 before a transition to deep software standby mode.

11.2.11 Deep Standby Interrupt Enable Register 2 (DPSIER2)

Address(es): 0008 C284h

	b7	b6	b5	b4	b3	b2	b1	b0
	DUSBIE	DRIICCIE	DRIICDIE	DNMIE	DRTCAIE	DRTCIIE	DLVD2IE	DLVD1IE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DLVD1IE	LVD1 Deep Standby Release Signal Enable	0: Disable release from deep software standby mode by the voltage monitoring 1 signal 1: Enable release from deep software standby mode by the voltage monitoring 1 signal	R/W
b1	DLVD2IE	LVD2 Deep Standby Release Signal Enable	0: Disable release from deep software standby mode by the voltage monitoring 2 signal 1: Enable release from deep software standby mode by the voltage monitoring 2 signal	R/W
b2	DRTCIIE	RTC Periodic Interrupt Deep Standby Release Signal Enable	0: Release from deep software standby mode by the RTC periodic interrupt signal is disabled 1: Release from deep software standby mode by the RTC periodic interrupt signal is enabled	R/W
b3	DRTCAIE	RTC Alarm Interrupt Deep Standby Release Signal Enable	0: Release from deep software standby mode by the RTC alarm interrupt signal is disabled 1: Release from deep software standby mode by the RTC alarm interrupt signal is enabled	R/W
b4	DNMIE	NMI Pin Enable	0: Release from deep software standby mode by the NMI pin is disabled 1: Release from deep software standby mode by the NMI pin is enabled	R/W*1
b5	DRIICDIE	SDA2-DS Deep Standby Release Signal Enable	0: Release from deep software standby mode by the SDA2-DS signal is disabled 1: Release from deep software standby mode by the SDA2-DS signal is enabled	R/W
b6	DRIICCIE	SCL2-DS Deep Standby Release Signal Enable	0: Release from deep software standby mode by the SCL2-DS signal is disabled 1: Release from deep software standby mode by the SCL2-DS signal is enabled	R/W
b7	DUSBIE	USB Suspend/Resume Deep Standby Release Signal Enable	0: Release from deep software standby mode by the USB suspend/resume is disabled 1: Release from deep software standby mode by the USB suspend/resume is enabled	R/W

Note 1. 1 can be written only once. Once 1 is written to this bit, subsequent write accesses are disabled.

Register DPSIER2 is not initialized by the internal reset signal used as deep software standby mode releasing source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.
After the setting of DPSIER2 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR2 being set to 1. Therefore, DPSIFR2 should be set to 0 before a transition to deep software standby mode.

DUSBIE Bit (USB Suspend/Resume Deep Standby Release Signal Enable)

The DUSBIE bit is a common enable bit for USBb and USBA. However, USBA is not released from deep software standby mode in response to the change of DM or DP. Also, USBb is not released from deep software standby mode using USB0_OVRCURB multiplexed with pin P22.

11.2.12 Deep Standby Interrupt Enable Register 3 (DPSIER3)

Address(es): 0008 C285h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DCANIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DCANIE	CRX1-DS Deep Standby Release Signal Enable	0: Release from deep software standby mode by the CRX1-DS pin is disabled 1: Release from deep software standby mode by the CRX1-DS pin is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Register DPSIER3 is not initialized by the internal reset signal used as deep software standby mode releasing source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

After the setting of DPSIER3 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR3 being set to 1. Therefore, DPSIFR3 should be set to 0 before a transition to deep software standby mode.

11.2.13 Deep Standby Interrupt Flag Register 0 (DPSIFR0)

Address(es): 0008 C286h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ7F	DIRQ6F	DIRQ5F	DIRQ4F	DIRQ3F	DIRQ2F	DIRQ1F	DIRQ0F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0F	IRQ0-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ0-DS pin 1: Request for release is being generated on the IRQ0-DS pin	R(/W) *1
b1	DIRQ1F	IRQ1-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ1-DS pin 1: Request for release is being generated on the IRQ1-DS pin	R(/W) *1
b2	DIRQ2F	IRQ2-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ2-DS pin 1: Request for release is being generated on the IRQ2-DS pin	R(/W) *1
b3	DIRQ3F	IRQ3-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ3-DS pin 1: Request for release is being generated on the IRQ3-DS pin	R(/W) *1
b4	DIRQ4F	IRQ4-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ4-DS pin 1: Request for release is being generated on the IRQ4-DS pin	R(/W) *1
b5	DIRQ5F	IRQ5-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ5-DS pin 1: Request for release is being generated on the IRQ5-DS pin	R(/W) *1
b6	DIRQ6F	IRQ6-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ6-DS pin 1: Request for release is being generated on the IRQ6-DS pin	R(/W) *1
b7	DIRQ7F	IRQ7-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ7-DS pin 1: Request for release is being generated on the IRQ7-DS pin	R(/W) *1

Note 1. Only 0 can be written to clear the flag.

Each flag is set to 1 when a request for release specified by the DPSIEGR0 register is generated.
Each flag may be set to 1 when a request for release is generated in any mode (not even deep software standby mode) or when the setting of the DPSIER0 register is modified. Therefore, a transition to deep software standby mode should be made after the DPSIFR0 register is set to 00h.
To set the DPSIFR0 register to 00h after modifying the DPSIER0 register, wait for at least six PCLKB cycles, read the DPSIFR0 register, and then write 0 to the DPSIFR0 register. Six or more PCLKB cycles can be secured, for example, by reading the DPSIER0 register.
The DPSIFR0 register is not initialized by the internal reset signal used as deep software standby mode releasing source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

DIRQnF Flags (IRQn Deep Standby Release Flag) (n = 0 to 7)

These flags indicate that a request for release has been generated on the IRQn-DS pin.
[Setting condition]

- A request for release is generated on the IRQn-DS pin specified by the DPSIEGR0 register

[Clearing condition]

- Each bit is read as 1 and then written by 0

11.2.14 Deep Standby Interrupt Flag Register 1 (DPSIFR1)

Address(es): 0008 C287h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ15F	DIRQ14F	DIRQ13F	DIRQ12F	DIRQ11F	DIRQ10F	DIRQ9F	DIRQ8F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ8F	IRQ8-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ8-DS pin 1: Request for release is being generated on the IRQ8-DS pin	R(/W) *1
b1	DIRQ9F	IRQ9-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ9-DS pin 1: Request for release is being generated on the IRQ9-DS pin	R(/W) *1
b2	DIRQ10F	IRQ10-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ10-DS pin 1: Request for release is being generated on the IRQ10-DS pin	R(/W) *1
b3	DIRQ11F	IRQ11-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ11-DS pin 1: Request for release is being generated on the IRQ11-DS pin	R(/W) *1
b4	DIRQ12F	IRQ12-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ12-DS pin 1: Request for release is being generated on the IRQ12-DS pin	R(/W) *1
b5	DIRQ13F	IRQ13-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ13-DS pin 1: Request for release is being generated on the IRQ13-DS pin	R(/W) *1
b6	DIRQ14F	IRQ14-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ14-DS pin 1: Request for release is being generated on the IRQ14-DS pin	R(/W) *1
b7	DIRQ15F	IRQ15-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ15-DS pin 1: Request for release is being generated on the IRQ15-DS pin	R(/W) *1

Note 1. Only 0 can be written to clear the flag.

Each flag is set to 1 when a request for release specified by the DPSIEGR1 register is generated.
Each flag may be set to 1 when a request for release is generated in any mode (not even deep software standby mode) or when the setting of the DPSIER1 register is modified. Therefore, a transition to deep software standby mode should be made after the DPSIFR1 register is set to 00h.
To set the DPSIFR1 register to 00h after modifying the DPSIER1 register, wait for at least six PCLKB cycles, read the DPSIFR1 register, and then write 0 to the DPSIFR1 register. Six or more PCLKB cycles can be secured, for example, by reading the DPSIER1 register.
The DPSIFR1 register is not initialized by the internal reset signal used as deep software standby mode releasing source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

DIRQnF Flags (IRQn Deep Standby Release Flag) (n = 8 to 15)

These flags indicate that a request for release has been generated on the IRQn-DS pin.
[Setting condition]

- A request for release is generated on the IRQn-DS pin specified by the DPSIEGR1 register

[Clearing condition]

- Each bit is read as 1 and then written by 0

11.2.15 Deep Standby Interrupt Flag Register 2 (DPSIFR2)

Address(es): 0008 C288h

	b7	b6	b5	b4	b3	b2	b1	b0
	DUSBIF	DRIICCIF	DRIICDIF	DNMIF	DRTCAIF	DRTCIIF	DLVD2IF	DLVD1IF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DLVD1IF	LVD1 Deep Standby Release Flag	0: Request for release by the voltage monitor 1 signal is not being generated 1: Request for release by the voltage monitor 1 signal is being generated	R(/W) *1
b1	DLVD2IF	LVD2 Deep Standby Release Flag	0: Request for release by the voltage monitor 2 signal is not being generated 1: Request for release by the voltage monitor 2 signal is being generated	R(/W) *1
b2	DRTCIIF	RTC Periodic Interrupt Deep Standby Release Flag	0: Request for release by the RTC periodic interrupt signal is not being generated 1: Request for release by the RTC periodic interrupt signal is being generated	R(/W) *1
b3	DRTCAIF	RTC Alarm Interrupt Deep Standby Release Flag	0: Request for release by the RTC alarm interrupt signal is not being generated 1: Request for release by the RTC alarm interrupt signal is being generated	R(/W) *1
b4	DNMIF	NMI Deep Standby Release Flag	0: Request for release is not being generated on the NMI pin 1: Request for release is being generated on the NMI pin	R(/W) *1
b5	DRIICDIF	SDA2-DS Deep Standby Release Flag	0: Request for release by the SDA2-DS signal is not being generated 1: Request for release by the SDA2-DS signal is being generated	R(/W) *1
b6	DRIICCIF	SCL2-DS Deep Standby Release Flag	0: Request for release by the SCL2-DS signal is not being generated 1: Request for release by the SCL2-DS signal is being generated	R(/W) *1
b7	DUSBIF	USB Suspend/Resume Deep Standby Release Flag	0: Request for release by the USB suspend/resume is not being generated 1: Request for release by the USB suspend/resume is being generated	R(/W) *1

Note 1. Only 0 can be written to clear the flag.

Each flag is set to 1 when a request for release specified by the DPSIEGR2 register is generated.
Each flag may be set to 1 when a request for release is generated in any mode (not even deep software standby mode) or when the setting of the DPSIER2 register is modified. Therefore, a transition to deep software standby mode should be made after the DPSIFR2 register is set to 00h.
To set the DPSIFR2 register to 00h after modifying the DPSIER2 register, wait for at least six PCLKB cycles, read the DPSIFR2 register, and then write 0 to the DPSIFR2 register. Six or more PCLKB cycles can be secured, for example, by reading the DPSIER2 register.
The DPSIFR2 register is not initialized by the internal reset signal used as deep software standby mode releasing source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

DLVDmIF Flag (LVDm Deep Standby Release Flag) (m = 1 or 2)

This flag indicates that a request for release by the voltage monitor m signal has been generated.
[Setting condition]

- A request for release is generated by the voltage monitoring m signal that is selected in the DPSIEGR2 register

[Clearing condition]

- This bit is read as 1 and then written by 0

DRTCIIF Flag (RTC Periodic Interrupt Deep Standby Release Flag)

This flag indicates that a request for release by the RTC periodic interrupt signal has been generated.
[Setting condition]

- A request for release by the RTC periodic interrupt signal is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

DRTCAIF Flag (RTC Alarm Interrupt Deep Standby Release Flag)

This flag indicates that a request for release by the RTC alarm interrupt signal has been generated.
[Setting condition]

- A request for release by the RTC alarm interrupt signal is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

DNMIF Flag (NMI Deep Standby Release Flag)

This flag indicates that a request for release has been generated on the NMI pin.
[Setting condition]

- A request for release is generated on the NMI pin specified by DPSIEGR2

[Clearing condition]

- This bit is read as 1 and then written by 0

DRIICDIF Flag (SDA2-DS Deep Standby Release Flag)

This flag indicates that a request for release by the SDA2-DS interrupt signal has been generated.
[Setting condition]

- A request for release is generated on the SDA2-DS pin specified by DPSIEGR2

[Clearing condition]

- This bit is read as 1 and then written by 0

DRIICCIF Flag (SCL2-DS Deep Standby Release Flag)

This flag indicates that a request for release by the SCL2-DS interrupt signal has been generated.
[Setting condition]

- A request for release is generated on by the SCL2-DS pin specified by DPSIEGR2

[Clearing condition]

- This bit is read as 1 and then written by 0

DUSBIF Flag (USB Suspend/Resume Deep Standby Release Flag)

This flag indicates that a request for release by the USB suspend/resume has been generated.
The DUSBIF flag is a common flag for USBb and USBA. However, USBA is not released from deep software standby mode in response to the change of DM or DP. Also, USBb is not released from deep software standby mode using USB0_OVRCURB multiplexed with pin P22.
Figure 11.2 shows the configuration of the DUSBIF flag.
[Setting condition]

- A request for release by the USB suspend/resume is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

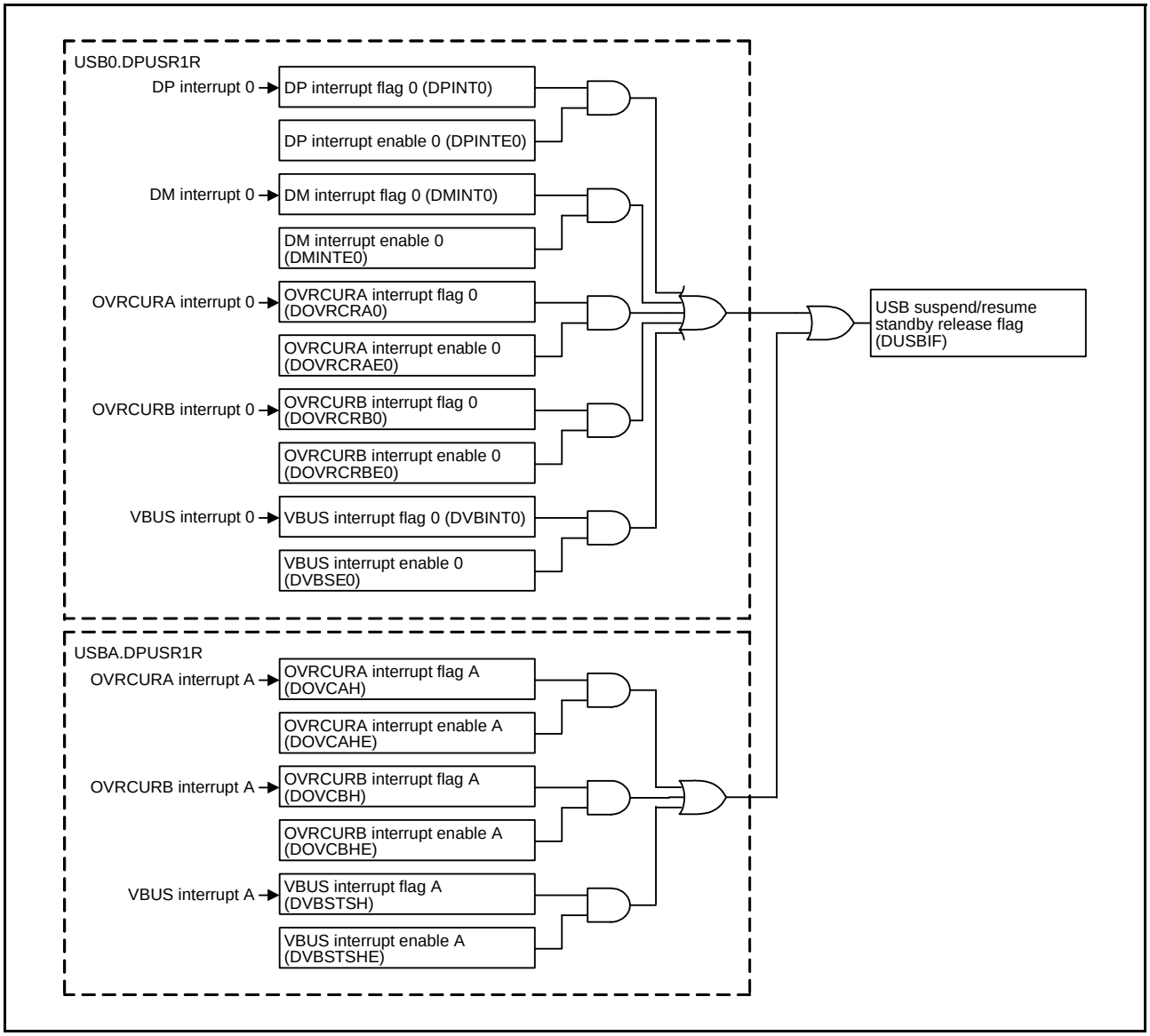


Figure 11.2 Configuration of the USB Suspend/Resume Deep Standby Release Flag (DUSBIF)

11.2.16 Deep Standby Interrupt Flag Register 3 (DPSIFR3)

Address(es): 0008 C289h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DCANIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DCANIF	CRX1-DS Deep Standby Release Flag	0: Request for release is not being generated on the CRX1-DS pin 1: Request for release is being generated on the CRX1-DS pin	R(/W) *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to clear the flag.

Each flag is set to 1 when a request for release specified by the DPSIEGR3 register is generated.
Each flag may be set to 1 when a request for release is generated in any mode (not even deep software standby mode) or when the setting of the DPSIER3 register is modified. Therefore, a transition to deep software standby mode should be made after the DPSIFR3 register is set to 00h.
To set the DPSIFR3 register to 00h after modifying the DPSIER3 register, wait for at least six PCLKB cycles, read the DPSIFR3 register, and then write 0 to the DPSIFR3 register. Six or more PCLKB cycles can be secured, for example, by reading the DPSIER3 register.
The DPSIFR3 register is not initialized by the internal reset signal used as deep software standby mode releasing source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

DCANIF Flag (CRX1-DS Deep Standby Release Flag)

This flag indicates that a request for release has been generated on the CRX1-DS pin.
[Setting condition]

- A request for release is generated on the CRX1-DS pin specified by the DPSIEGR3 register

[Clearing condition]

- This bit is read as 1 and then written by 0

11.2.17 Deep Standby Interrupt Edge Register 0 (DPSIEGR0)

Address(es): 0008 C28Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ7EG	DIRQ6EG	DIRQ5EG	DIRQ4EG	DIRQ3EG	DIRQ2EG	DIRQ1EG	DIRQ0EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0EG	IRQ0-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b1	DIRQ1EG	IRQ1-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b2	DIRQ2EG	IRQ2-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b3	DIRQ3EG	IRQ3-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b4	DIRQ4EG	IRQ4-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b5	DIRQ5EG	IRQ5-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b6	DIRQ6EG	IRQ6-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b7	DIRQ7EG	IRQ7-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W

Register DPSIEGR0 is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

11.2.18 Deep Standby Interrupt Edge Register 1 (DPSIEGR1)

Address(es): 0008 C28Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ15EG	DIRQ14EG	DIRQ13EG	DIRQ12EG	DIRQ11EG	DIRQ10EG	DIRQ9EG	DIRQ8EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ8EG	IRQ8-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b1	DIRQ9EG	IRQ9-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b2	DIRQ10EG	IRQ10-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b3	DIRQ11EG	IRQ11-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b4	DIRQ12EG	IRQ12-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b5	DIRQ13EG	IRQ13-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b6	DIRQ14EG	IRQ14-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b7	DIRQ15EG	IRQ15-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W

Register DPSIEGR1 is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

11.2.19 Deep Standby Interrupt Edge Register 2 (DPSIEGR2)

Address(es): 0008 C28Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DRIICCEG	DRIICDEG	DNMIEG	—	—	DLVD2EG	DLVD1EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DLVD1EG	LVD1 Edge Select	0: A request for release is generated when VCC < Vdet1 (fall) is detected 1: A request for release is generated when VCC ≥ Vdet1 (rise) is detected	R/W
b1	DLVD2EG	LVD2 Edge Select	0: A request for release is generated when VCC < Vdet2 (fall) is detected 1: A request for release is generated when VCC ≥ Vdet2 (rise) is detected	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DNMIEG	NMI Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b5	DRIICDEG	SDA2-DS Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b6	DRIICCEG	SCL2-DS Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Register DPSIEGR2 is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

11.2.20 Deep Standby Interrupt Edge Register 3 (DPSIEGR3)

Address(es): 0008 C28Dh

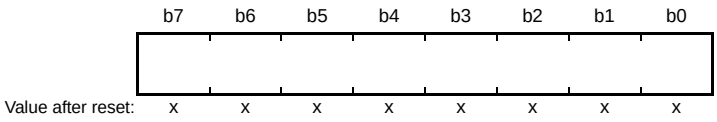
	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DCANIEG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DCANIEG	CRX1-DS Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Register DPSIEGR3 is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

11.2.21 Deep Standby Backup Register (DPSBKRy) (y = 0 to 31)

Address(es): 0008 C2A0h to 0008 C2BFh



x: Undefined

Register DPSBKRy is a 32-byte readable/writable register to store data during deep software standby mode. The value of this register is retained even in deep software standby mode where RAM data is not retained. DPSBKRy is not initialized, and the register value is undefined immediately after power-on.

11.3 Reducing Power Consumption by Switching Clock Signals

When the SCKCR.FCK[3:0], ICK[3:0], BCK[3:0], PCKA[3:0], PCKB[3:0], PCKC[3:0], and PCKD[3:0] bits are set, the clock frequency changes. The CPU, DMAC, DTC, code flash memory, and RAM operate on the operating clock specified by the ICK[3:0] bits.
Peripheral modules operate on the operating clock specified by the PCKA[3:0], PCKB[3:0], PCKC[3:0], and PCKD[3:0] bits.
The data flash memory operates on the operating clock specified by the FCK[3:0] bits.
The external bus operates on the operating clock specified by the BCK[3:0] bits. For details, see section 9, Clock Generation Circuit.

11.4 Module-Stop Function

The module-stop function can be set for each on-chip peripheral module.
When the MSTPmi bit (m = A to D, i = 31 to 0) in registers MSTPCRA to MSTPCRD is set to 1, the specified module stops operating and enters the module-stop state, but the CPU continues to operate independently. When the corresponding MSTPmi bit is set to 0, the module is released from the module-stop state and restarts operating at the end of the bus cycle. The internal states of modules are retained in the module-stop state.
After release from a reset, all modules except for the DMAC, DTC, EXDMAC, RAM, ECCRAM, and standby RAM are placed in the module-stop state. Though read/write access cannot be made to the registers of the module that are in the module-stop state, some registers may be written to directly after the setting to the module-stop state. Therefore, care should be paid.

11.5 Function for Lower Operating Power Consumption

By selecting an appropriate operating power consumption control mode according to the operating frequency and operating voltage, power consumption can be reduced in normal operation, sleep mode and all-module clock stop mode.

11.5.1 Setting Operating Power Consumption Control Mode

Examples of the procedures for switching operating power consumption control modes are shown below:

(1) Switching from Normal Power Consumption Mode to Low Power Consumption Mode

Example: From high-speed operating mode to low-speed operating mode 1

(High-speed operation in the operating power consumption control mode used before mode-switching)
↓
Set to switch from the HOCO clock to the LOCO clock (clock source and frequency division ratio)
↓
Write to register OPCCR (high-speed operating mode → low-speed operating mode 1)
↓
Confirm that the OPCCR.OPCMTSF flag is 0
↓
(Low-speed operation in the switched operating power consumption control mode)

(2) Switching from Low Power Consumption Mode to Normal Power Consumption Mode

Example: From low-speed operating mode 2 to high-speed operating mode

(Low-speed operation in the operating power consumption control mode used before mode-switching)
↓
Write to register OPCCR (low-speed operating mode 2 → high-speed operating mode)
↓
Confirm that the OPCCR.OPCMTSF flag is 0
↓
Set to switch from the LOCO clock to the HOCO clock (clock source and frequency division ratio)
↓
(High-speed operation in the switched operating power consumption control mode)

11.6 Low Power Consumption Modes

11.6.1 Sleep Mode

11.6.1.1 Transition to Sleep Mode

When the WAIT instruction is executed while the SBYCR.SSBY bit is 0, the CPU enters sleep mode. In sleep mode, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop.
When the WDT is used, the WDT stops counting when sleep mode is entered.
Counting by the IWDT stops if a transition to sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1 (stopping counting by the IWDT at transitions to low-power-consumption modes). In the same way, counting by the IWDT stops if a transition to sleep mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 1.
Furthermore, counting by the IWDT continues if a transition to sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low-power-consumption modes). In the same way, counting by the IWDT continues if a transition to sleep mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 0.
To use sleep mode, make the following settings and then execute a WAIT instruction.

(1) Clear the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt request destination*2 to be used for recovery from sleep mode to the CPU.
(3) Set the priority*3 of the interrupt to be used for recovery from sleep mode to a level higher than the setting of the PSW.IPL[3:0] bits*1 of the CPU.
(4) Set the IERm.IENj bit*3 for that interrupt to 1.
(5) For the last I/O register to which writing proceeded, read the register to confirm that the value written has been reflected.
(6) Execute the WAIT instruction (this automatically sets the I bit*1 in the PSW of the CPU to 1).

Note 1. For details, see section 2, CPU.
Note 2. For details, see section 15.7.3, Selecting Interrupt Request Destination.
Note 3. For details, see section 15, Interrupt Controller (ICUA).

11.6.1.2 Release from Sleep Mode

Release from sleep mode is initiated by a non-maskable interrupt, an interrupt, the RES# pin reset, a power-on reset, a voltage monitoring reset, or a reset caused by an IWDT underflow.

- Release triggered by an interrupt signal
  Generation of an interrupt triggers release from sleep mode and the interrupt exception processing starts. If a maskable interrupt has been masked by the CPU (the priority level*1 of the interrupt has been set to a value lower than that of the PSW.IPL[3:0] bits*2 of the CPU), release from sleep mode does not proceed.
- Release due to a reset on the RES# pin
  When the RES# pin is driven low, the LSI enters the reset state. When the RES# pin is driven high after the reset signal is input for a predetermined time period, the CPU starts the reset exception processing.
- Release due to a power-on reset
  Release from sleep mode is initiated by a power-on reset.
- Release due to a voltage monitoring reset
  Release from sleep mode is initiated by a voltage monitoring reset from the voltage detection circuit.
- Release due to the independent watchdog timer reset
  Release from sleep mode is initiated by an internal reset generated by an IWDT underflow. However, when such conditions are set that stop IWDT counting in sleep mode (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1), the IWDT is stopped and release from sleep mode is not initiated by the independent watchdog timer reset.

Note 1. For details, see section 15, Interrupt Controller (ICUA).
Note 2. For details, see section 2, CPU.

11.6.1.3 Sleep Mode Return Clock Source Switching Function

To switch the clock source used on return from sleep mode, the clock used after return needs to be set by the sleep mode return clock source switching register (RSTCKCR) and the wait control register needs to be set for each clock source. When the return interrupt is generated, after oscillation settling of the oscillator specified as the return clock, the clock source is automatically switched, and then operation returns from sleep mode. At this time, the registers related to clock source switching are automatically rewritten.
For details, see section 11.2.7, Sleep Mode Return Clock Source Switching Register (RSTCKCR). For setting a waiting time for oscillation stabilization, see section 9.2.16, Main Clock Oscillator Wait Control Register (MOSCWTCR).

11.6.2 All-Module Clock Stop Mode

11.6.2.1 Transition to All-Module Clock Stop Mode

After setting the MSTPCRA.ACSE bit to 1 and placing modules controlled by MSTPCRA, MSTPCRB, MSTPCRC, and MSTPCRD registers in the module-stop state (MSTPCRA = FFFF FF[C-F]Fh, MSTPCRB = FFFF FFFFh, MSTPCRC[31:16] = FFFFh, MSTPCRD = FFFF FFFFh), executing a WAIT instruction while the SBYCR.SSBY bit is 0 stops the bus controller, I/O ports, and all modules except for the 8-bit timers*1, POE*2, IWDT, RTC, power-on reset circuit, voltage detection circuit at the end of the current bus cycle, and the chip enters all-module clock stop mode*3. When the WDT is used, the WDT stops counting when all-module clock stop mode is entered.

Counting by the IWDT stops if a transition to all-module clock-stop mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1 (stopping counting by the IWDT at transitions to low-power-consumption modes). In the same way, counting by the IWDT stops if a transition to all-module clock-stop mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 1.

Furthermore, counting by the IWDT continues if a transition to all-module clock-stop mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low-power-consumption modes). In the same way, counting by the IWDT continues if a transition to all-module clock-stop mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 0.

To use all-module clock-stop mode, make the following settings and then execute a WAIT instruction.

(1) Clear the PSW.I bit*4 of the CPU to 0.
(2) Set the interrupt request destination*5 to be used for recovery from all-module clock stop mode to the CPU.
(3) Set the priority*6 of the interrupt to be used for recovery from all-module clock stop mode to a level higher than the setting of the PSW.IPL[3:0] bits*4 of the CPU.
(4) Set the IERm.IENj bit*6 for the interrupt to be used for recovery from all-module clock stop mode to 1.
(5) Read the last I/O register to have been written and confirm that its value reflects the value written.
(6) Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the PSW.I bit*4 of the CPU to 1).

Note 1. The MSTPCRA.MSTPA4 and MSTPA5 bits select operation or stop of these modules.
Note 2. When a POE interrupt source condition is satisfied while the setting to enable POE interrupts is in place, recovery from all-module clock stop mode does not proceed but the flag to indicate satisfaction of the source condition is retained. If a different source leads to recovery from all-module clock stop mode in this situation, a POE interrupt is generated after recovery.
Note 3. Transitions to all-module clock stop mode are not to be made in some states of DTC or DMAC operations. Before setting the MSTPCRA.MSTPA28 bit to 1, clear the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC so that the DTC and DMAC are not activated.
Note 4. For details, see section 2, CPU.
Note 5. For details, see section 15.7.3, Selecting Interrupt Request Destination.
Note 6. For details, see section 15, Interrupt Controller (ICUA).

11.6.2.2 Release from All-Module Clock Stop Mode

Release from all-module clock-stop mode is initiated by an external pin interrupt (the NMI or IRQ0 to IRQ15), a peripheral interrupt (8-bit timer*1, RTC alarm, RTC periodic, IWDT*2, USB suspend/resume, voltage monitoring 1, voltage monitoring 2, or oscillator-stopped detection interrupt), a RES# pin reset, a power-on reset, a voltage monitoring reset, or an independent watchdog timer reset, and the transition to the normal program execution state proceeds via exception processing for the given interrupt or reset.

However, note that in cases where a maskable interrupt has been masked by the CPU (priority level*3 of the interrupt has been set to a value lower than that of the PSW.IPL[3:0] bits*4 of the CPU) or a maskable interrupt has been set up as a trigger to start the DTC or DMA transfer, release from all-module clock stop mode will not proceed.

Note 1. The MSTPA4 and MSTPA5 bits of MSTPCRA select operation or stopping of these modules.
Note 2. If a condition for the independent watchdog timer to stop counting applied (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1) at the time of a transition to all-module clock-stop mode, using a reset from the independent watchdog timer to release the chip from all-module clock-stop mode is impossible because the independent watchdog timer is stopped.
Note 3. For details, see section 15, Interrupt Controller (ICUA).
Note 4. For details, see section 2, CPU.

11.6.3 Software Standby Mode

11.6.3.1 Transition to Software Standby Mode

When a WAIT instruction is executed with the SBYCR.SSBY bit set to 1 and the DPSBYCR.DPSBY bit set to 0, a transition to software standby mode is made. In this mode, the CPU, on-chip peripheral functions, and the oscillator functions stop. However, the contents of the CPU internal registers, RAM data, on-chip peripheral functions, and the states of the I/O ports are retained. Whether the address bus and bus control signals are placed in the high-impedance or the output state is retained can be specified by the SBYCR.OPE bit. Software standby mode allows significant reduction in power consumption because the oscillator stops in this mode. Yet, the main and sub clock oscillators can be operated or stopped. For details, see Table 11.2, Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode.

Clear the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0 before executing the WAIT instruction.

When the WDT is used, the WDT stops counting when software standby mode is entered because the oscillator stops. Counting by the IWDT stops if a transition to software standby mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1 (stopping counting by the IWDT at transitions to low-power-consumption modes). In the same way, counting by the IWDT stops if a transition to software standby mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 1.

Furthermore, counting by the IWDT continues if a transition to software standby mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low-power-consumption modes). In the same way, counting by the IWDT continues if a transition to software standby mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 0.

When the oscillation stop detection function is enabled (OSTDCR.OSTDE = 1), software standby mode cannot be entered. To make a transition to software standby mode, execute a WAIT instruction after disabling the oscillation stop detection function (OSTDCR.OSTDE = 0).

To use software standby mode, make the following settings and then execute a WAIT instruction.

(1) When the ethernet controller is in use, set bits ETHERC0.ECMR.TE and ETHERC0.ECMR.RE to 0. After checking that bits EDMAC0.EDTRR.TR and EDMAC0.EDRRR.RR have been set to 0, wait for 100 ns, and then set the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt request destination*2 to be used for recovery from software standby mode to the CPU.
(3) Set the priority*3 of the interrupt to be used for recovery from software standby mode to a level higher than the setting of the PSW.IPL[3:0] bits*1 of the CPU.
(4) Set the IERm.IENj bit*3 for the interrupt to be used for recovery from software standby mode to 1.
(5) For the last I/O register to which writing proceeded, read the register to confirm that the value written has been reflected.
(6) Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the PSW.I bit*1 of the CPU to 1).

Note 1. For details, see section 2, CPU.
Note 2. For details, see section 15.7.3, Selecting Interrupt Request Destination.
Note 3. For details, see section 15, Interrupt Controller (ICUA).

11.6.3.2 Release from Software Standby Mode

Release from software standby mode is initiated by an external pin interrupt (the NMI or IRQ0 to IRQ15), peripheral interrupts (the RTC alarm, RTC periodic, IWDT, USB suspend/resume, voltage monitoring 1, and voltage monitoring 2 interrupts), an RES# pin reset, a power-on reset, a voltage monitoring reset, or an independent watchdog timer reset. When an interrupt initiates release from software standby, the oscillators which were stopped by the transition to software standby are restarted. After the oscillation of all these oscillators has become stable, operation returns from software standby.
Note that the oscillators are not stopped by the transition to software standby under the following two conditions, but the return from software standby still follows the period for stabilization of oscillation by the oscillators if they had been stopped.

- MOFCR.MOFXIN = 1 and MOSCCR.MOSTP = 0
- RCR3.RTCEN = 1 and SOSCCR.SOSTP = 0

(1) Release due to an interrupt
When an interrupt request from the NMI, IRQ0 to IRQ15, RTC alarm, RTC periodic, IWDT, USB suspend/resume, voltage monitoring 1, or voltage monitoring 2 interrupt is generated, each oscillator which was operating before a transition to software standby mode resumes oscillation. After the time for return from software standby has elapsed, the chip is released from software standby and starts interrupt exception processing.
The time for return after release from software standby is the oscillation stabilization waiting time plus the time required for operations by the software standby release sequencer.

tSBYi = tSBYOSCWT + tSBYSEQ
tSBYi (i = MC, EX, PC, PE, PH, SC, HO, LO): Time for return after release from software standby
tSBYOSCWT: Oscillation stabilization waiting time
tSBYSEQ: Time required for operations by the software standby release sequencer

For the oscillation stabilization waiting time to be used in calculating the time for return after release from software standby, use the greatest value of the oscillation stabilization waiting time of the oscillators which are to be started. For the oscillation stabilization waiting times of the oscillators, see section 64, Electrical Characteristics.

(2) Release due to a reset on the RES# pin
Clock oscillation starts when the low level is applied to the RES# pin. Clock supply for the LSI starts at the same time. Keep the level on the RES# pin low over the time required for oscillation of the clocks to become stable. Reset exception processing starts when the high level is applied to the RES# pin.

(3) Release due to a power-on reset
Release from software standby mode proceeds after a fall in the power-supply voltage leads to the generation of a power-on reset.

(4) Release due to a voltage monitoring reset
Release from software standby mode proceeds after a fall in the power-supply voltage leads to the generation of a voltage monitoring reset.

(5) Release due to an independent watchdog timer reset
An internal reset due to an underflow of the IWDT leads to release from software standby mode.
However, if a condition for the independent watchdog timer to stop counting applied (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1) at the time of a transition to software standby, using a reset from the independent watchdog timer to release the chip from software standby is impossible because the independent watchdog timer is stopped.

11.6.3.3 Example of Software Standby Mode Application

Figure 11.3 shows an example where a transition to software standby mode is made at the falling edge of the IRQn pin, and release from software standby mode is initiated at the rising edge of the IRQn pin.
In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 01b (falling edge), and then the IRQCRi.IRQMD[1:0] bits are set to 10b (rising edge). After that, the SBYCR.SSBY bit is set to 1 and the WAIT instruction is executed. Thus a transition to software standby mode is made. After that, release from software standby mode is initiated at the rising edge of the IRQn pin.
To return from software standby mode, settings of the interrupt controller (ICU) are also necessary. For details, see section 15, Interrupt Controller (ICUA).

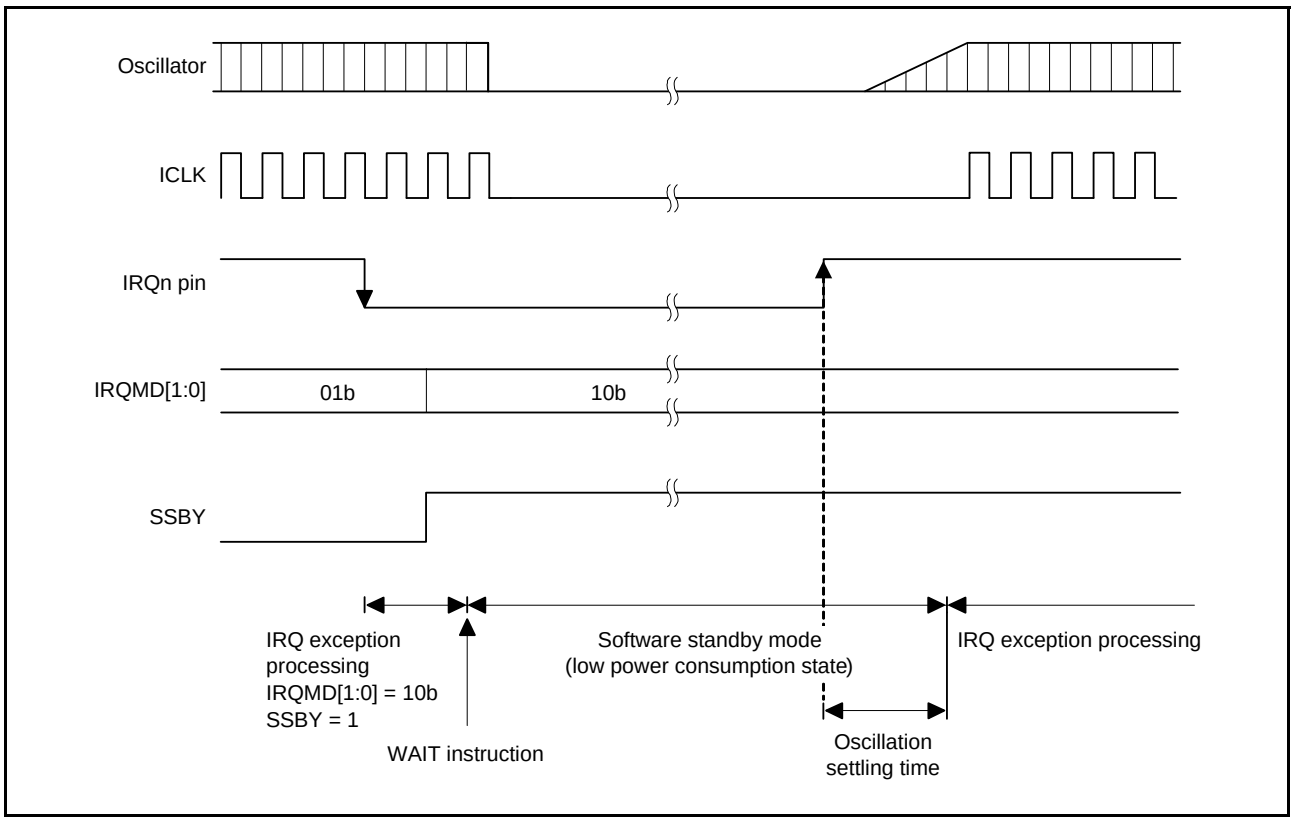


Figure 11.3 Example of Software Standby Mode Application

11.6.4 Deep Software Standby Mode

11.6.4.1 Transition to Deep Software Standby Mode

When the WAIT instruction is executed with the SBYCR.SSBY bit set to 1, a transition to software standby mode*1 is made. At this time, when the DPSBYCR.DPSBY bit is set to 1, a transition to deep software standby mode is made. On deep software standby mode, the CPU, internal peripheral modules (except for parts of the RTC alarm, RTC periodic, SCL2-DS, SDA2-DS, CRX1-DS, and USB suspend/resume detecting unit), RAM (except for standby RAM), and functions of the oscillators are stopped; furthermore, since the internal supply of power for these modules is stopped, power consumption is markedly reduced. Yet, the main and sub clock oscillators can be operated or stopped. For details, see Table 11.2, Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode. At this time, the contents of all the registers of the CPU and internal peripheral modules (except for parts of the RTC alarm, RTC periodic, SCL2-DS, SDA2-DS, CRX1-DS, and USB suspend/resume detecting unit) become undefined.
Data in the standby RAM are preserved if the setting of the DEEPCUT[1:0] bits is 00b. If the setting of the DEEPCUT[1:0] bits is 01b, the internal supply of power to the standby RAM and the USB resume detecting unit is cut off, reducing power consumption. Data in the standby RAM become undefined at this time. If the setting of the DEEPCUT[1:0] bits is 11b, the internal supply of power to the standby RAM, and the USB resume detecting unit is cut off, the LVD is stopped, and the low-power-consumption function of the power-on reset circuit is enabled, so power consumption is further reduced. For details, see section 64, Electrical Characteristics.
When the WDT is in use, since the oscillators and power supply to the WDT are stopped by the transition to deep software standby mode, counting also stops.
Power supply to the IWDT-dedicated clock and the IWDT is stopped and counting by the IWDT stops if a transition to deep software standby mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1 (stopping counting by the IWDT at transitions to low-power-consumption modes). In the same way, power supply to the IWDT-dedicated clock and the IWDT is stopped and counting by the IWDT stops if a transition to deep software standby mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 1.
Furthermore, counting by the IWDT continues if a transition to software standby mode is made but not to deep software standby mode while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low-power-consumption modes). In the same way, counting by the IWDT continues if a transition to software standby mode is made but not to deep software standby mode while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 0.
When the voltage monitoring 1 reset function (LVD1CR0.LVD1RI = 1) or voltage monitoring 2 reset function (LVD2CR0.LVD2RI = 1) is selected for the voltage detection circuit, a transition to deep software standby mode cannot be made, but to software standby.
The I/O port states remain unchanged from software standby mode.

Note 1. Conditions on the DTC, DMAC, and IWDT for transition to software standby mode should be met before the WAIT instruction is executed. For details, see section 11.6.3, Software Standby Mode.

11.6.4.2 Release from Deep Software Standby Mode

Release from deep software standby mode is initiated by any of the external pin interrupt source pins (the NMI, IRQ0-DS to IRQ15-DS, SCL2-DS, SDA2-DS, or CRX1-DS), peripheral interrupts (the RTC alarm, RTC periodic, USB suspend/resume*1, voltage monitoring 1, and voltage monitoring 2 interrupts), an RES# pin reset, a power-on reset, or a voltage monitoring 0 reset.

Note 1. However, USBA is not released from deep software standby mode in response to the change of DM or DP. Also, USBb is not released from deep software standby mode using USB0_OVRCURB multiplexed with pin P22.

(1) Release triggered by an external interrupt pin or internal interrupt signal
Release from deep software standby mode is controlled by registers DPSIERn (n = 0 to 3) and DPSIFRn (n = 0 to 3). When a deep software standby release interrupt is generated, the corresponding flag in DPSIFRn is set to 1. At this time, if the releasing source is enabled in DPSIERn, release from deep software standby mode proceeds. Rising edge or falling edge can be selected by DPSIEGRn (n = 0 to 3) registers. The interrupts for which an edge can be selected are the NMI, IRQ0-DS to IRQ15-DS, SCL2-DS, SDA2-DS, CRX1-DS, voltage monitoring 1, and voltage monitoring 2 interrupts.
When a deep software standby mode releasing source is generated, the internal power supply and LOCO clock oscillation begin, and then a deep software standby reset is generated for the entire LSI.
A stable LOCO clock is then supplied to the entire LSI, which is released from deep software standby reset. This is accompanied by release from deep software standby, and reset exception processing then starts.
When release from deep software standby is triggered by an external interrupt pin or internal interrupt signal, the RSTSR0.DPSRSTF flag is set to 1.

(2) Release due to a reset on the RES# pin
The low level being applied to the RES# pin triggers release from deep software standby.
At this time, the RES# pin should be held low according to the specifications described in section 64, Electrical Characteristics. Reset exception processing starts when the high level is applied to the RES# pin.

(3) Release due to a power-on reset
Release from deep software standby mode proceeds after a fall in the power-supply voltage leads to the generation of a power-on reset.

(4) Release due to a voltage monitoring 0 reset
Release from deep software standby mode proceeds after a fall in the power-supply voltage leads to the generation of a voltage monitoring 0 reset.

11.6.4.3 Pin States at the Time of Release from Deep Software Standby Mode

In deep software standby mode, the I/O ports retain the same states from software standby mode. The inside of the LSI is initialized by an internal reset generated on release from deep software standby mode. Upon release from deep software standby mode, the reset exception processing starts. The following shows the states of I/O ports at this time.
Whether to initialize the I/O ports or to retain the I/O port states at the time of software standby mode can be selected by the DPSBYCR.IOKEEP bit.

- When the DPSBYCR.IOKEEP bit = 0
  I/O ports are initialized by an internal reset generated on release from deep software standby mode.
- When the DPSBYCR.IOKEEP bit = 1
  Although the inside of the LSI is initialized by an internal reset generated on release from deep software standby mode, I/O ports retain their states from software standby mode regardless of the LSI internal state. At this time, the I/O port states remain unchanged from software standby mode even if settings of I/O ports or peripheral modules are made. Then, the retained I/O port states are released by clearing the DPSBYCR.IOKEEP bit to 0, and the LSI operates according to the internal state.

The DPSBYCR.IOKEEP bit is not initialized by an internal reset generated on release from deep software standby mode.

11.6.4.4 Example of Deep Software Standby Mode Application

Figure 11.4 shows an example where a transition to deep software standby mode is made at the falling edge of the IRQn-DS pin, and release from deep software standby mode is initiated at the rising edge of the IRQn-DS pin.
In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 01b (falling edge). Then, after the DPSIEGRy.DIRQnEG (y = 0 or 1, n = 0 to 15) bit is set to 1 (rising edge) and the SBYCR.SSBY bit and DPSBYCR.DPSBY bit are both set to 1, the WAIT instruction is executed. Thus a transition to deep software standby mode is made.
After that, release from deep software standby mode is initiated at the rising edge of the IRQ-DS pin.

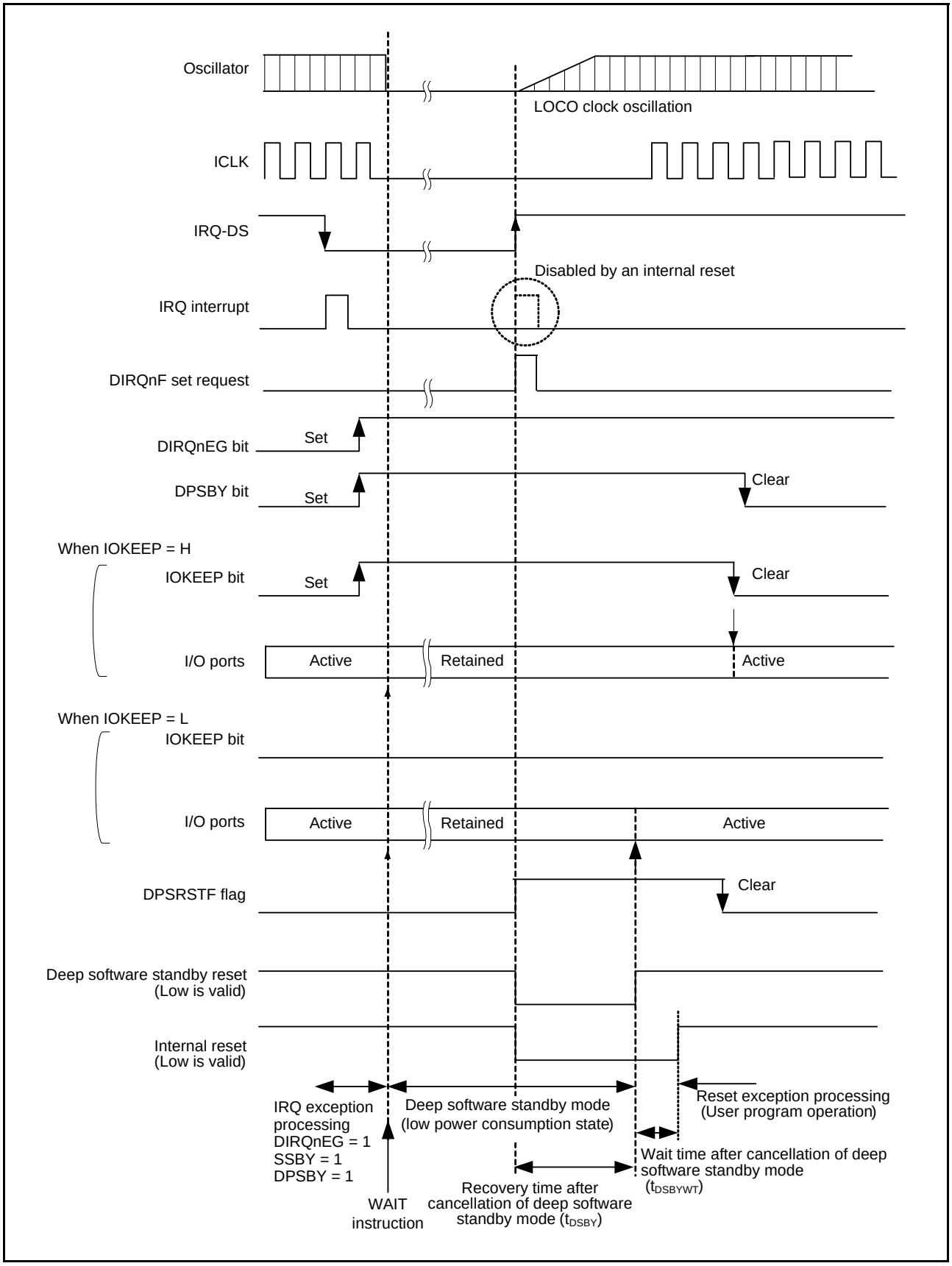


Figure 11.4 Example of Deep Software Standby Mode Application

11.6.4.5 Flowchart to Use Deep Software Standby Mode

Figure 11.5 shows an example of a flowchart to use deep software standby mode.
In this example, the RSTSR0.DPSRSTF flag of the reset function is read after the reset exception processing to determine whether a reset was generated by the RES# pin or by release from deep software standby mode.
In the case of a reset by the RES# pin, a transition to deep software standby mode is made after the required register settings have been made.
In the case of a reset by release from deep software standby mode, the DPSBYCR.IOKEEP bit is set to 0 after the I/O port settings have been made.

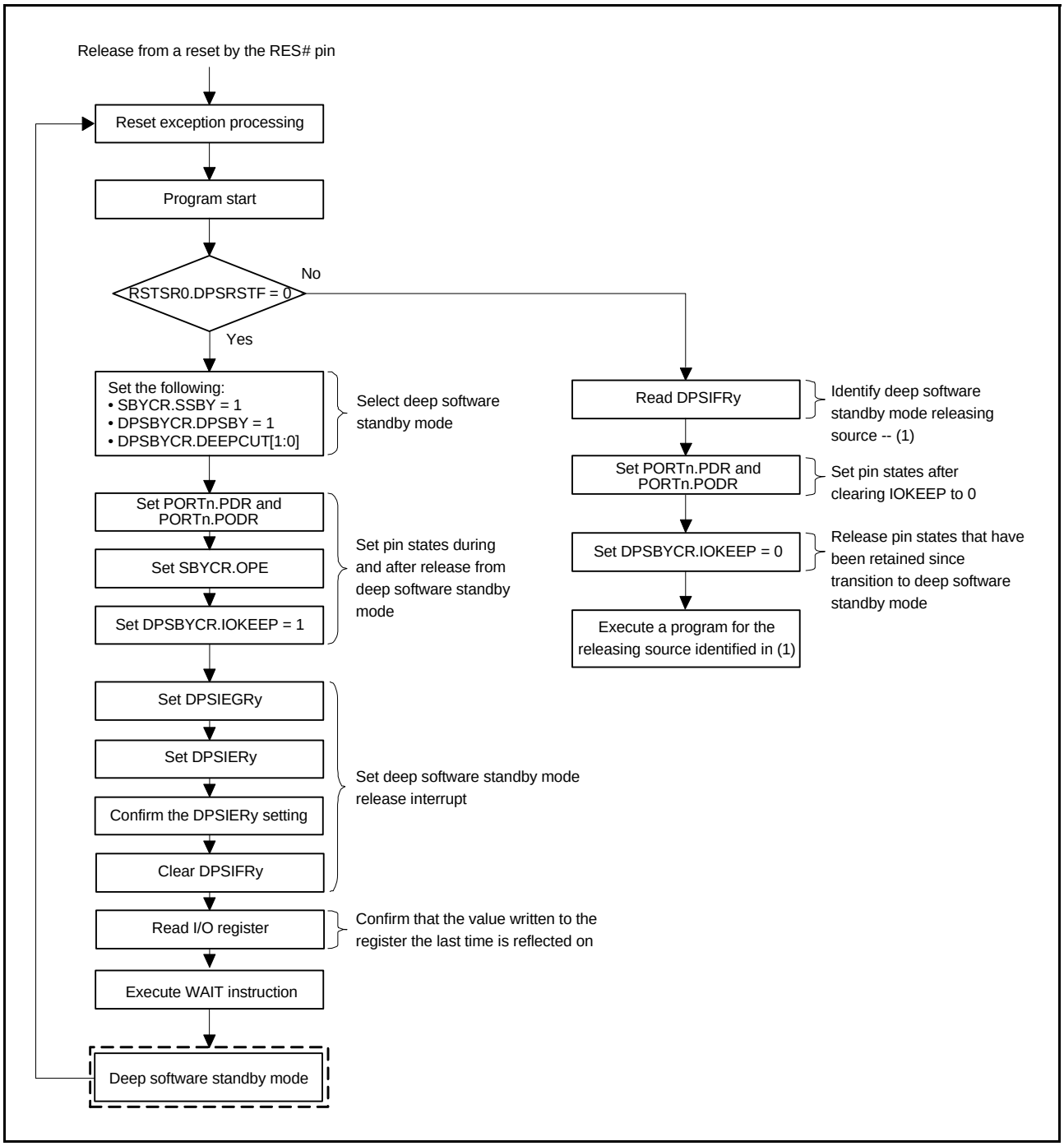


Figure 11.5 Example of Flowchart to Use Deep Software Standby Mode

11.7 Usage Notes

11.7.1 I/O Port States

I/O port states are retained in software standby mode and deep software standby mode. Therefore, the supply current is not reduced while output signals are held high.

11.7.2 Module-Stop State of DMAC and DTC

Before setting the MSTPCRA.MSTPA28 bit to 1, clear the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0 so that the DMAC and DTC are not activated.
For details, see section 18, DMA Controller (DMACAa) and section 20, Data Transfer Controller (DTCa).

11.7.3 On-Chip Peripheral Module Interrupts

These interrupts do not operate in the module-stop state. Therefore, if the module-stop state is entered after an interrupt request is generated, a CPU interrupt source or a DMAC or DTC startup source cannot be cleared. For this reason, disable interrupts before entering the module-stop state.

11.7.4 Write Access to MSTPCRA, MSTPCRB, MSTPCRC, and MSTPCRD

Write accesses to registers MSTPCRA, MSTPCRB, MSTPCRC and MSTPCRD should be made only by the CPU.

11.7.5 Input Buffer Control by DIRQnE Bit (n = 0 to 15)

Setting the DPSIERy.DIRQnE (y = 0 or 1, n = 0 to 15) bit to 1 enables the input buffer of the IRQ0-DS to IRQ15-DS pins. Therefore, note that, although inputs to these pins are sent to the DPSIFRy.DIRQnF (y = 0 or 1, n = 0 to 15) bits, they are not sent to the interrupt controller, peripheral modules, and I/O ports.

11.7.6 Timing of Wait Instructions

A WAIT instruction that follows register writing may be executed before the writing is completed. Accordingly, the WAIT instruction may be executed before the change to the setting of an I/O register is reflected, in which case operation may not be as intended. To avoid this, always execute the WAIT instruction after confirming that the last writing to the register has completed.

11.7.7 Rewriting the Register by DMAC and DTC in Sleep Mode

The WDT stops in sleep mode. Do not set up the DMACA and DTC to rewrite any registers related to the WDT while the chip is in sleep mode.
According to the settings of the OFS0.IWDTSLCSTP bit and IWDTCSTPR.SLCSTP bit, the IWDT may also stop in sleep mode. If that is the case, do not set up the DMAC and DTC to rewrite any registers related to the IWDT in sleep mode.
The RSTCKCR register can be set so that the clock source is switched on recovery from sleep mode. For this reason, rewriting the register while the chip is in sleep mode may lead to unintended operation, so do not allow rewriting of the RSTCKCR register in sleep mode.

11.7.8 Point for Caution when Shifting from Low-Speed Operating Mode to Software Standby Mode

On return from software standby, the chip enters high-speed operating mode. Even if a WAIT instruction is executed in low-speed operating mode, if generation of the return interrupt precedes completion of the transition to software standby and processing for the transition is canceled, the chip does not return to the mode before execution of the WAIT instruction. If this creates a problem, set the OPCCR.OPCM[2:0] bits to 000b during processing of the return interrupt.

12. Battery Backup Function

12.1 Overview

When the voltage at the VCC pin is dropped, power can be supplied to the realtime clock (RTC) and sub-clock oscillator from the dedicated battery backup power pin (VBATT pin). When the voltage drop at the VCC pin is detected, connection to power is switched to the VBATT pin.

Figure 12.1 shows the configuration of the battery backup function.

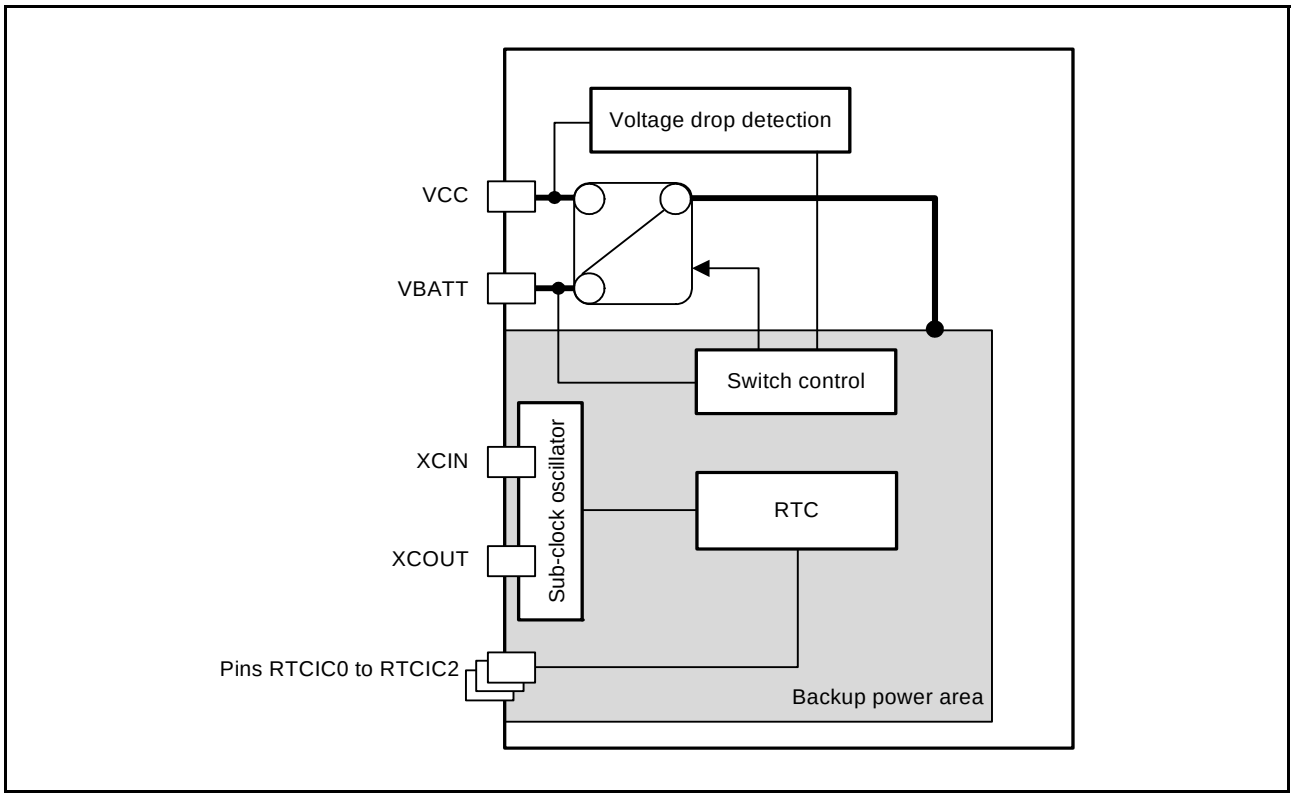


Figure 12.1 Configuration of Battery Backup Function

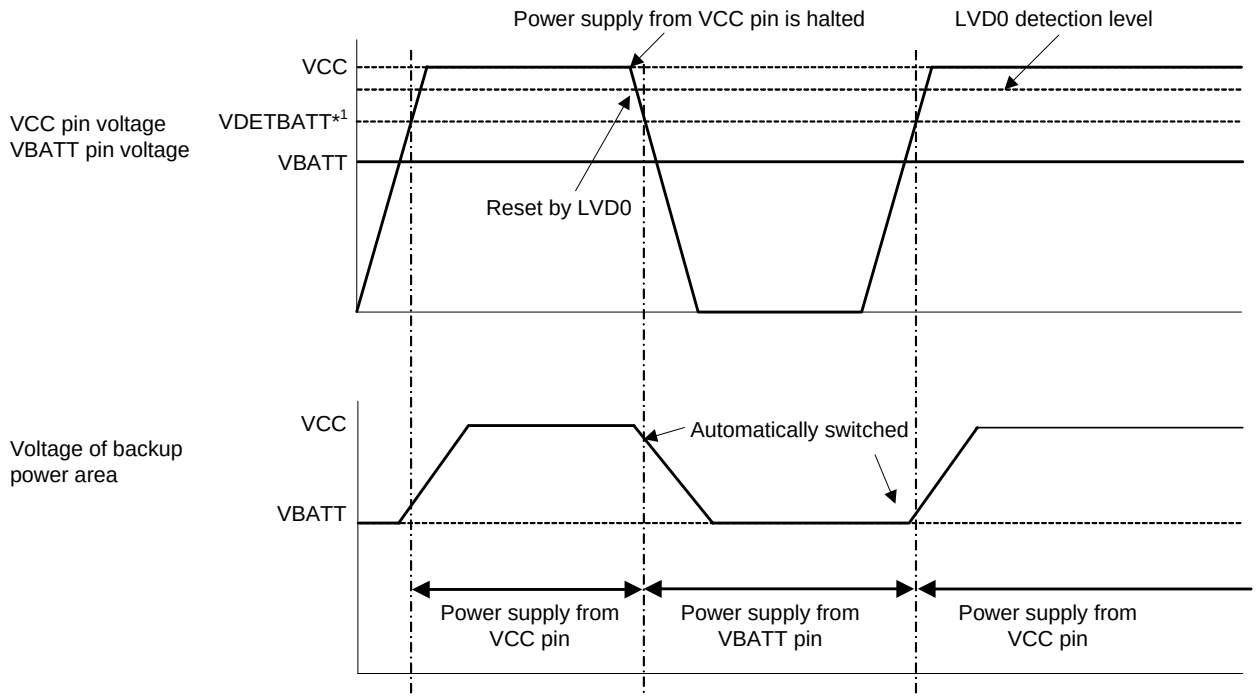
12.2 Operation

12.2.1 Battery Backup Function

When the voltage at the VCC pin is dropped, power can be supplied to the RTC and sub-clock oscillator from the VBATT pin. When the power supply reduction from the VCC pin is detected, connection to power is switched to the power supply from the VBATT pin. The power supply from the VCC pin is resumed when the voltage at the VCC pin exceeds VDETBATT while the RTC is operating on the power supply from the VBATT pin. This power supply change does not affect the RTC operation. When the voltage level at the VBATT pin voltage falls below the operation guaranteed voltage, operation of the RTC cannot be guaranteed.
In addition, the battery backup function should be used after the voltage monitoring 0 reset is enabled.
The power is supplied to the following modules from the VBATT pin.

- RTC
- Sub-clock oscillator (including XCIN and XCOUT pins)
- P30, P31, and P32 input pins

Figure 12.2 shows the operation for switching to the battery backup function.



Note: For details on electrical characteristics, see section 64, Electrical Characteristics.
Note 1. VDETBATT indicates the judgment level of the power supply change between the VCC pin and the VBATT pin.

Figure 12.2 Operation for Switching to Battery Backup Function

12.3 Usage Notes

1. When the VBATT pin is not in use, connect the VBATT pin to the VCC pin.
2. When the voltage level at the VBATT is lower than the guaranteed operation range, operation of the sub-clock and RTC cannot be guaranteed. The RTC must be initialized to restart power supply after the VBATT pin falls below the operation guaranteed voltage.
3. Writing to the RTC registers should be performed while power is being supplied from the VCC pin.
4. When VCC is higher than VDETBATT, the VCC pin and VBATT pin are not connected by means of circuitry. When VCC is lower than VDETBATT and the switch is connected to the VBATT pin, if the voltage at the VBATT pin becomes lower than (the range from VCC to 0.6 V), current may flow into the VBATT pin through the parasitic diode between the VCC and VBATT pins.
5. During RTC operation using the voltage from the VBATT pin, I/O ports (P30, P31, and P32) within the backup power supply area can only be used as time capture event input pins for the RTC.

13. Register Write Protection Function

The register write protection function protects important registers from being overwritten for in case a program runs out of control. The registers to be protected are set with the protect register (PRCR).

Table 13.1 lists the association between the PRCR bits and the registers to be protected.

Table 13.1 Association between PRCR Bits and Registers to be Protected

PRCR Bit	Register to be Protected
PRC0	• Registers related to the clock generation circuit: SCKCR, SCKCR2, SCKCR3, PLLCR, PLLCR2, BCKCR, MOSCCR, SOSCCR, LOCOCR, ILOCOCR, HOCOCR, HOCOCR2, OSTDCR, OSTDSR
PRC1	• Registers related to the operating modes: SYSCR0, SYSCR1 • Registers related to the low power consumption functions: SBYCR, MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD, OPCCR, RSTCKCR, DPSBYCR, DPSIER0 to DPSIER3, DPSIFR0 to DPSIFR3, DPSIEGR0 to DPSIEGR3 • Registers related to clock generation circuit: MOSCWTCR, SOSCWTCR, MOFCR, HOCOPCR • Software reset register: SWRR
PRC3	• Registers related to the LVD: LVCMPCR, LVDLVLR, LVD1CR0, LVD1CR1, LVD1SR, LVD2CR0, LVD2CR1, LVD2SR

13.1 Register Descriptions

13.1.1 Protect Register (PRCR)

Address(es): 0008 03FEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRKEY[7:0]								—	—	—	—	PRC3	—	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
b0	PRC0	Protect Bit 0	Enables writing to the registers related to the clock generation circuit. 0: Write disabled 1: Write enabled	R/W
b1	PRC1	Protect Bit 1	Enables writing to the registers related to operating modes, low power consumption, and software reset. 0: Write disabled 1: Write enabled	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	PRC3	Protect Bit 3	Enables writing to the registers related to the LVD. 0: Write disabled 1: Write enabled	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	PRKEY[7:0]	PRC Key Code	These bits control permission and prohibition of writing to the PRCR register. To modify the PRCR register, write A5h to the eight higher-order bits and the desired value to the eight lower-order bits as a 16-bit unit.	R/(W)*1

Note 1. Written values are not retained. These bits are read as 00h.

PRCi Bits (Protect Bit i) (i = 0, 1, 3)

These bits enable or disable writing to the corresponding registers to be protected.

Setting the PRCi bits to 1 and 0 enables and disables writing to the corresponding registers to be protected, respectively.

14. Exception Handling

14.1 Exception Events

During execution of a program by the CPU, the occurrence of a certain event may cause execution of that program to be suspended and execution of another program to be started. Such kinds of events are called exception events.
The RXv2 CPU supports eight types of exceptions. The types of exception events are shown in Figure 14.1.
The occurrence of an exception causes the processor mode to shift to supervisor mode.

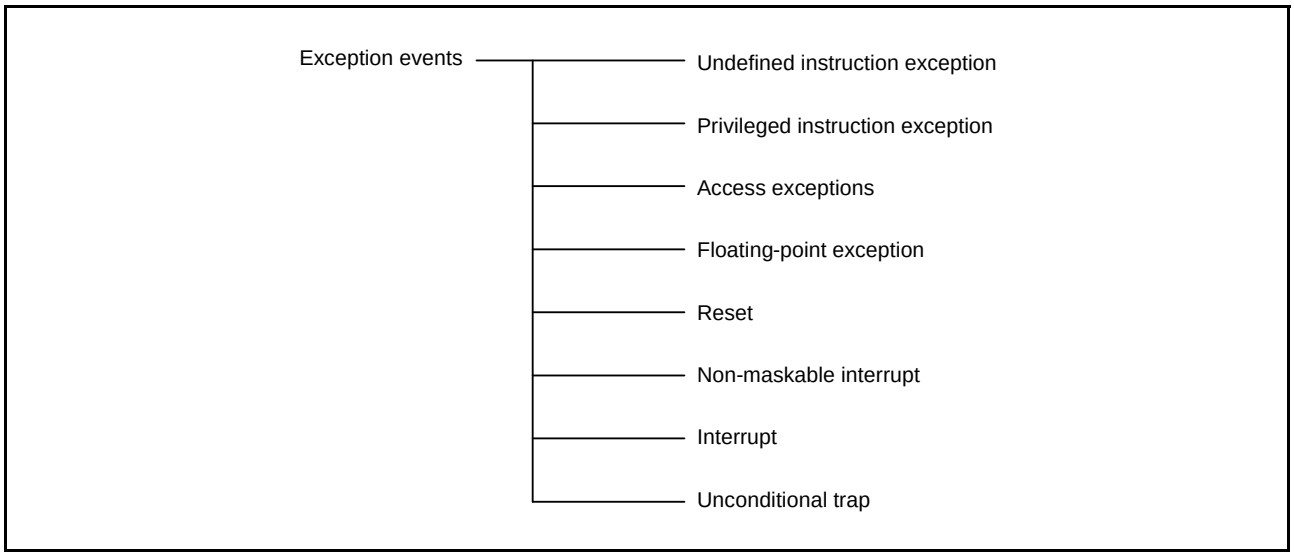


Figure 14.1 Types of Exception Events

14.1.1 Undefined Instruction Exception

An undefined instruction exception occurs when execution of an undefined instruction (an instruction not implemented) is detected.

14.1.2 Privileged Instruction Exception

A privileged instruction exception occurs when execution of a privileged instruction is detected in user mode. Privileged instructions can be executed only in supervisor mode.

14.1.3 Access Exceptions

An access exception occurs when an error is detected in access to memory by the CPU. If the memory-protection unit detects an instruction memory-protection error, an instruction-access exception occurs, and if the unit detects a data memory protection error, an operand-access exception occurs.

14.1.4 Floating-Point Exception

Floating-point exceptions include the five exception events (overflow, underflow, inexact, division-by-zero, and invalid operation) specified in the IEEE754 standard and another floating-point exception that is generated on detection of unimplemented processing. The exception handling of floating-point exceptions is prohibited when the EX, EU, EZ, EO, or EV bit in FPSW is 0.

14.1.5 Reset

A reset is generated by input of a reset signal to the CPU. This has the highest priority of any exception and is always accepted.

14.1.6 Non-Maskable Interrupt

The non-maskable interrupt is generated by input of a non-maskable interrupt signal to the CPU and is only used when a fatal fault is considered to have occurred in the system. Never use the non-maskable interrupt with an attempt to return to the program that was being executed at the time of interrupt generation after the exception handling routine is ended.

14.1.7 Interrupt

Interrupts are generated by the input of interrupt signals to the CPU. A fast interrupt can be selected as the interrupt with the highest priority. In the case of the fast interrupt, hardware pre-processing and hardware post-processing are handled fast. The priority level of the fast interrupt is 15 (the highest). The exception handling of interrupts is masked when the I bit in PSW is 0.

14.1.8 Unconditional Trap

An unconditional trap is generated when the INT or BRK instruction is executed.

14.2 Exception Handling Procedure

In the exception handling, part of the processing is handled automatically by hardware and part of it is handled by a program (exception handling routine) that has been written by the user. Figure 14.2 shows the processing procedure when an exception other than a reset is accepted.

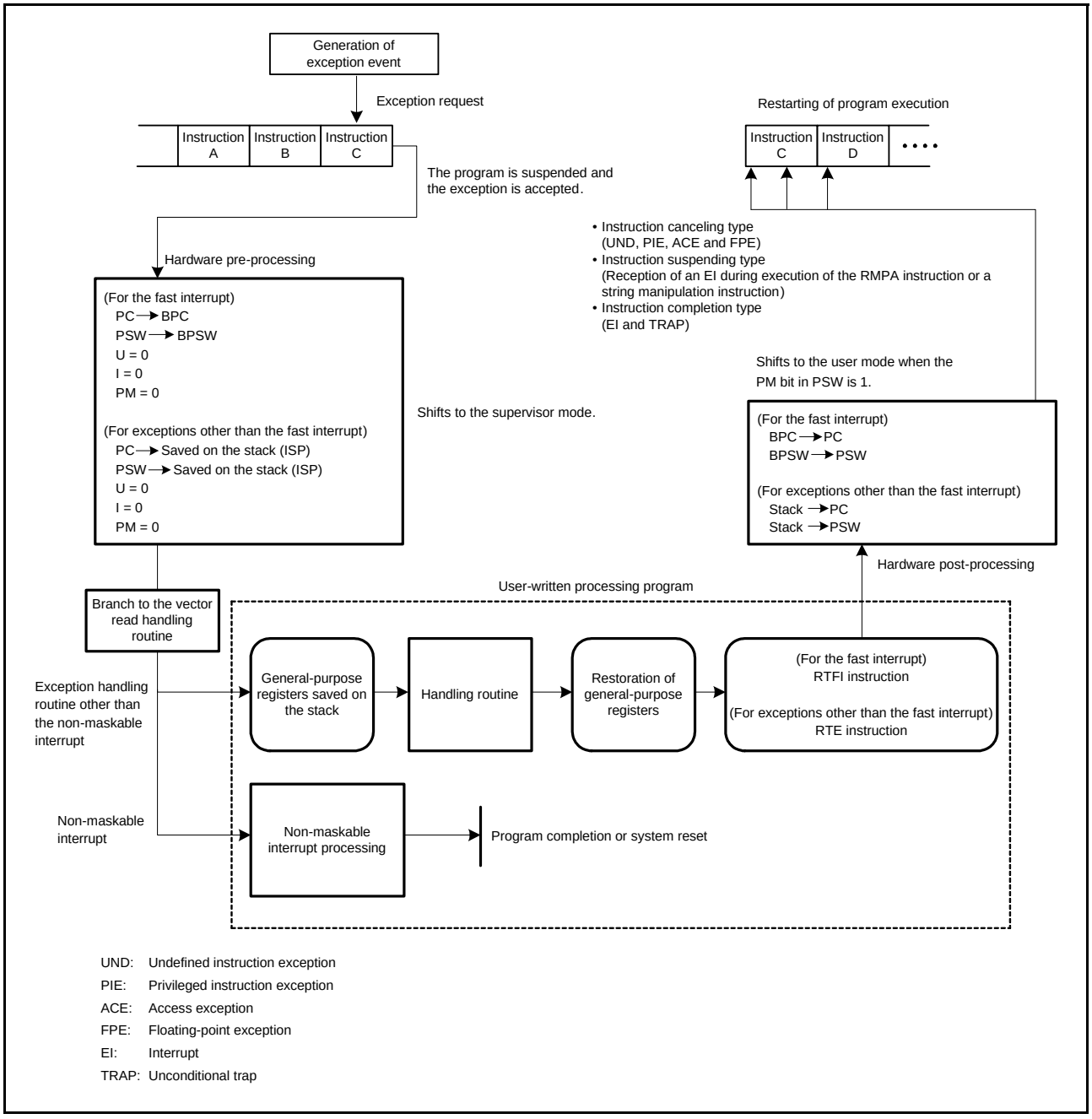


Figure 14.2 Outline of Exception Handling Procedure

When an exception is accepted, hardware processing by the RXv2 CPU is followed by access to the vector to acquire the address of the branch destination. In the vector, a vector address is allocated to each exception, and the branch destination address of the exception handling routine is written to each vector address.
Hardware pre-processing by the RXv2 CPU handles saving of the contents of the program counter (PC) and processor status word (PSW). In the case of a fast interrupt, the contents are saved in the backup PC (BPC) and the backup PSW (BPSW), respectively. In the case of exceptions other than a fast interrupt, the contents are saved in the stack area.
General purpose registers and control registers other than the PC and PSW that are to be used within the exception handling routine must be saved on the stack by a user program at the start of the exception handling routine.
On completion of processing by an exception handling routine, registers saved on the stack are restored and the RTE instruction is executed to restore execution from the exception handling routine to the original program. For return from a fast interrupt, the RTFI instruction is used instead. In the case of a non-maskable interrupt, however, finish the program or reset the system without returning to the original program.
Hardware post-processing by the RXv2 CPU handles restoration of the contents of PC and PSW. In the case of a fast interrupt, the values of BPC and BPSW are restored to PC and PSW, respectively. In the case of exceptions other than a fast interrupt, the values are restored from the stack to PC and PSW.

14.3 Acceptance of Exception Events

When an exception occurs, the CPU suspends the execution of the program and processing branches to the exception handling routine.

14.3.1 Acceptance Timing and Saved PC Value

Table 14.1 lists the timing of acceptance and the program counter (PC) value to be saved for each exception event.

Table 14.1 Acceptance Timing and Saved PC Value

Exception Event		Type of Handling	Acceptance Timing	Value Saved in BPC or on the Stack
Undefined instruction exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Privileged instruction exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Access exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Floating-point exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Reset		Instruction abandonment type	Any machine cycle	None
Non-maskable interrupt	During execution of the RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions	Instruction suspending type	During instruction execution	PC value of the instruction being executed
	Other than above	Instruction completion type	At the next break between instructions	PC value of the next instruction
Interrupt	During execution of the RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions	Instruction suspending type	During instruction execution	PC value of the instruction being executed
	Other than above	Instruction completion type	At the next break between instructions	PC value of the next instruction
Unconditional trap		Instruction completion type	At the next break between instructions	PC value of the next instruction

14.3.2 Vector and Site for Saving the Values in the PC and PSW

The vector for each type of exception and the site for saving the values of the program counter (PC) and processor status word (PSW) are listed in Table 14.2. The addresses where the exception vector table and interrupt vector table start must be set. For details, see section 2.6, Vector Table.

Table 14.2 Vector and Site for Saving the Values in the PC and PSW

Exception		Vector	Site for Saving the Values in the PC and PSW
Undefined instruction exception		Exception vector table (EXTB)	Stack
Privileged instruction exception		Exception vector table (EXTB)	Stack
Access exception		Exception vector table (EXTB)	Stack
Floating-point exception		Exception vector table (EXTB)	Stack
Reset		Exception vector table (EXTB)	Nowhere
Non-maskable interrupt		Exception vector table (EXTB)	Stack
Interrupt	Fast interrupt	FINTV	BPC and BPSW
	Other than above	Interrupt vector table (INTB)	Stack
Unconditional trap		Interrupt vector table (INTB)	Stack

14.4 Hardware Processing for Accepting and Returning from Exceptions

This section describes the hardware processing for accepting and returning from exceptions other than a reset.

(1) Hardware Pre-Processing for Accepting an Exception

(a) Saving PSW

- For a fast interrupt
  PSW → BPSW
- For exceptions other than a fast interrupt
  PSW → Stack

Note: The values in FPSW are not saved by hardware pre-processing. Therefore, if floating-point instructions are to be used within an exception handling routine, the user must save these values on the stack within the exception handling routine.

(b) Updating PM, U, and I Bits in PSW

I: Set to 0
U: Set to 0
PM: Set to 0

(c) Saving PC

- For a fast interrupt
  PC → BPC
- For exceptions other than a fast interrupt
  PC → Stack

(d) Setting Branch Destination Address of Exception Handling Routine in PC

Processing is shifted to the exception handling routine by acquiring the vector corresponding to the exception and then branching accordingly.

(2) Hardware Post-Processing for Execution of RTE and RTFI Instructions

(a) Restoring PSW

- For a fast interrupt
  BPSW → PSW
- For exceptions other than a fast interrupt
  Stack → PSW

(b) Restoring PC

- For a fast interrupt
  BPC → PC
- For exceptions other than a fast interrupt
  Stack → PC

14.5 Hardware Pre-Processing

The hardware pre-processing from reception of each exception request to execution of the associated exception handling routine are explained below.

14.5.1 Undefined Instruction Exception

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 005Ch.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.2 Privileged Instruction Exception

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 0050h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.3 Access Exceptions

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 0054h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.4 Floating-Point Exception

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 0064h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.5 Reset

1. The control registers are initialized.
2. The vector is fetched from address FFFF FFFCh.
3. The fetched vector is set to the PC.

14.5.6 Non-Maskable Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved on the stack (ISP). For other instructions, the PC value of the next instruction is saved.
4. The processor interrupt priority level bits (IPL[3:0]) in PSW are set to Fh.
5. The vector is fetched from the value of EXTB + address 0000 0078h.
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.7 Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP) or, for the fast interrupt, in the backup PSW (BPSW).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved. For other instructions, the PC value of the next instruction is saved. Saving of the PC is in the backup PC (BPC) for fast interrupts.
4. The processor interrupt priority level bits (IPL[3:0]) in PSW indicate the interrupt priority level of the interrupt.
5. The vector for an interrupt source other than the fast interrupt is fetched from the interrupt vector table. For the fast interrupt, the address is fetched from the fast interrupt vector register (FINTV).
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.8 Unconditional Trap

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) for the next instruction is saved on the stack (ISP).
4. For the INT instruction, the value at the vector corresponding to the INT instruction number is fetched from the interrupt vector table.
   For the BRK instruction, the value at the vector from the start address is fetched from the interrupt vector table.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.6 Return from Exception Handling Routine

Executing the instruction listed in Table 14.3 at the end of the corresponding exception handling routine restores the values of the program counter (PC) and processor status word (PSW) that were saved on the stack or in the control registers (BPC and BPSW) immediately before the exception handling sequence.

Table 14.3 Return from Exception Handling Routine

Exception		Instruction for Return
Undefined instruction exception		RTE
Privileged instruction exception		RTE
Access exception		RTE
Floating-point exception		RTE
Reset		Return is impossible
Non-maskable interrupt		Prohibited
Interrupt	Fast interrupt	RTFI
	Other than above	RTE
Unconditional trap		RTE

14.7 Priority of Exception Events

The priority of exception events is listed in Table 14.4. When multiple exceptions are generated at the same time, the exception with the highest priority is accepted first.

Table 14.4 Priority of Exception Events

Priority		Exception Event
High	1	Reset
↑	2	Non-maskable interrupt
	3	Interrupt
	4	Instruction access exception
	5	Undefined instruction exception Privileged instruction exception
	6	Unconditional trap
	7	Operand access exception
Low	8	Floating-point exception

15. Interrupt Controller (ICUA)

15.1 Overview

The interrupt controller (ICU) controls various interrupt requests from the peripheral modules and the IRQi pin (i = 0 to 15), and generates an interrupt request to the CPU and a transfer request to the DTC and DMAC.
Table 15.1 lists the ICU specifications, and Figure 15.1 shows a block diagram of the interrupt controller.

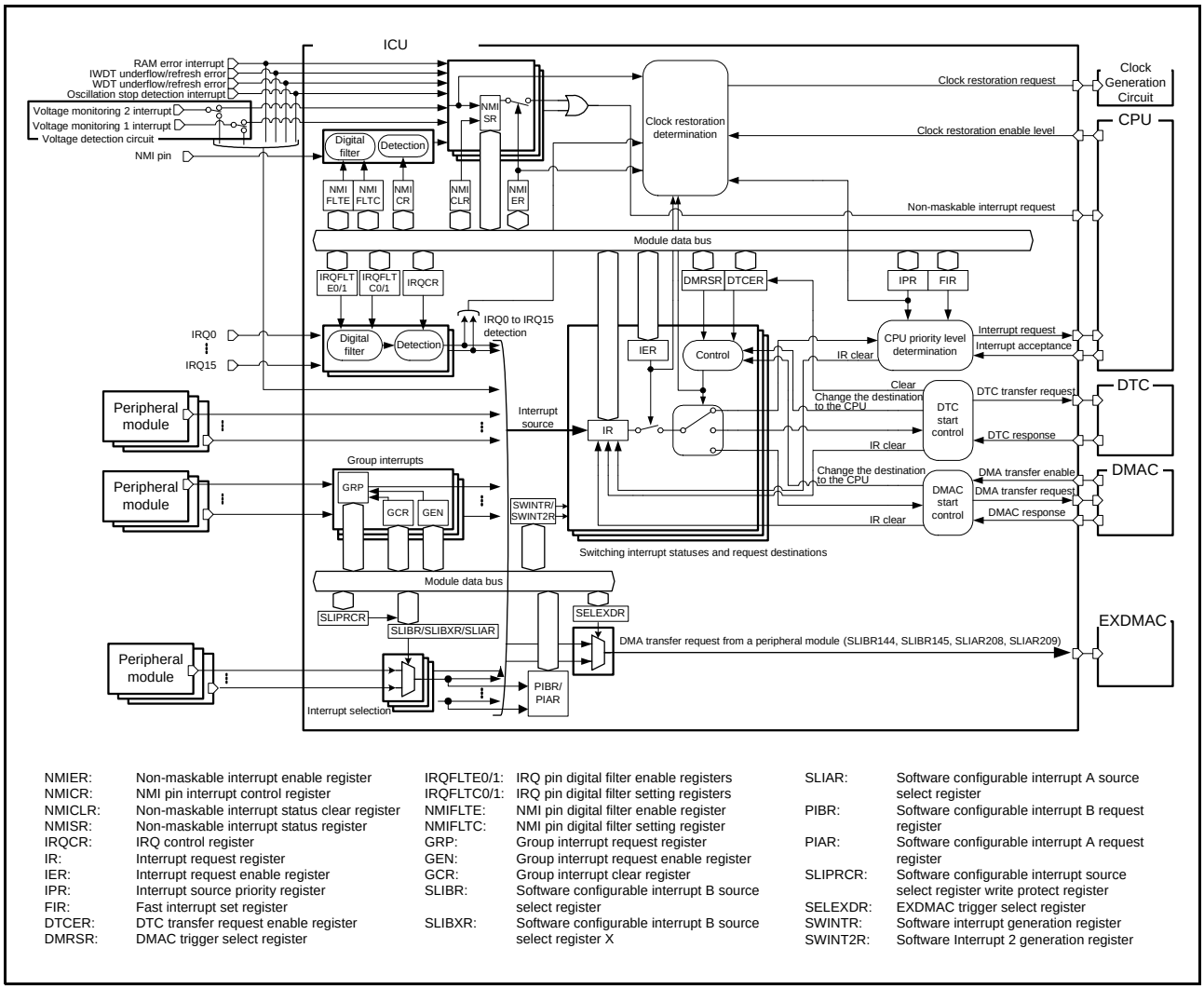
Table 15.1 ICU Specifications (1/2)

Item		Description
Interrupts	Peripheral interrupts	Interrupts from peripheral modules • Interrupt detection method: Edge detection/level detection (fixed for each interrupt source) • Group interrupt: Multiple interrupt sources are grouped together and treated as an interrupt source. Group BE0 interrupt: Interrupt sources of peripheral modules that use PCLKB as the operating clock (edge detection) Group BL0/BL1 interrupt: Interrupt sources of peripheral modules that use PCLKB as the operating clock (level detection) Group AL0/AL1 interrupt: Interrupt sources of peripheral modules that use PCLKA as the operating clock (level detection) • Software configurable interrupt B: Any of the interrupt sources for peripheral modules that use PCLKB as the operating clock can be assigned to interrupt vector numbers 128 to 207. • Software configurable interrupt A: Any of the interrupt sources for peripheral modules that use PCLKA as the operating clock can be assigned to interrupt vector numbers 208 to 255.
	External pin interrupt	Interrupt by the input signal to the IRQi pin (i = 0 to 15) • Interrupt detection method: Detection of low level, falling edge, rising edge, rising and falling edges One of these detection methods can be set for each source. • Digital filter can be used to remove noise.
	Software interrupt	• Interrupt request can be generated by writing to a register. • Two interrupt sources
	Interrupt priority	Priority level can be set with interrupt source priority register r (IPRr) (r = 000 to 255).
	Fast interrupt function	CPU interrupt response time can be reduced. This function can be used for only one interrupt source.
	DTC/DMAC control	Interrupt sources can be used to start the DTC and DMAC.*1
	EXDMAC control	Interrupt selected by software configurable interrupt B source select register 144 or software configurable interrupt A source select register 208 can be used to start EXDMAC0. Interrupt selected by software configurable interrupt B source select register 145 or software configurable interrupt A source select register 209 can be used to start EXDMAC1.
Non-maskable interrupts *2	NMI pin interrupt	Interrupt by the input signal to the NMI pin • Interrupt detection: Falling edge/rising edge • Digital filter can be used to remove noise.
	Oscillation stop detection interrupt *3	This interrupt occurs when the main clock oscillator stop is detected.
	WDT underflow/refresh error interrupt *3	This interrupt occurs when the watchdog timer (WDT) underflows or a refresh error occurs.
	IWDT underflow/refresh error interrupt *3	This interrupt occurs when the independent watchdog timer (IWDT) underflows or a refresh error occurs.
	Voltage monitoring 1 interrupt *3	Interrupt from voltage detection circuit 1 (LVD1)
	Voltage monitoring 2 interrupt *3	Interrupt from voltage detection circuit 2 (LVD2)
	RAM error interrupt *3	This interrupt occurs when a parity check error is detected in the RAM or an ECC error is detected in the ECCRAM.

Table 15.1 ICU Specifications (2/2)

Item		Description
Return from low power consumption state	Sleep mode	• Exit sleep mode by any interrupt source.
	All-module clock stop mode	• Exit all-module clock stop mode by the NMI pin interrupt, external pin interrupt, or peripheral interrupt (voltage monitoring 1, voltage monitoring 2, oscillation stop detection interrupt, USB resume, RTC alarm, RTC period, USBA resume, IWDT, software configurable interrupt 146 to 157).
	Software standby mode	• Exit all-module clock stop mode by the NMI pin interrupt, external pin interrupt, or peripheral interrupt (voltage monitoring 1, voltage monitoring 2, USB resume, RTC alarm, RTC period, USBA resume, IWDT).
	Deep software standby mode	• Exit all-module clock stop mode by the NMI pin interrupt, specific external pin interrupt, or peripheral interrupt (voltage monitoring 1, voltage monitoring 2, USB resume, RTC alarm, RTC period, USBA resume).

Note 1. For the DTC and DMAC triggers, refer to Table 15.5, Interrupt Vector Table.
Note 2. Once non-maskable interrupts are enabled, they cannot be disabled.
Note 3. Each source for these non-maskable interrupts can be used for maskable interrupts. When using for maskable interrupts, do not change the NMIER register value from the value after reset. To enable the voltage monitoring 1 interrupt, set the LVD1CR1.LVD1IRQSEL bit to 1, and to enable the voltage monitoring 2 interrupt, set the LVD2CR1.LVD2IRQSEL bit to 1.



NMIER:	Non-maskable interrupt enable register	IRQFLTE0/1:	IRQ pin digital filter enable registers	SLIAR:	Software configurable interrupt A source select register
NMICR:	NMI pin interrupt control register	IRQFLTC0/1:	IRQ pin digital filter setting registers	PIBR:	Software configurable interrupt B request register
NMICLR:	Non-maskable interrupt status clear register	NMIFLTE:	NMI pin digital filter enable register	PIAR:	Software configurable interrupt A request register
NMISR:	Non-maskable interrupt status register	NMIFLTC:	NMI pin digital filter setting register	SLIPRCR:	Software configurable interrupt source select register write protect register
IRQCR:	IRQ control register	GRP:	Group interrupt request register	SELEXDR:	EXDMAC trigger select register
IR:	Interrupt request register	GEN:	Group interrupt request enable register	SWINTR:	Software interrupt generation register
IER:	Interrupt request enable register	GCR:	Group interrupt clear register	SWINT2R:	Software Interrupt 2 generation register
IPR:	Interrupt source priority register	SLIBR:	Software configurable interrupt B source select register		
FIR:	Fast interrupt set register				
DTCER:	DTC transfer request enable register	SLIBXR:	Software configurable interrupt B source select register X		
DMRSR:	DMAC trigger select register				

Figure 15.1 Block Diagram of the ICU

Table 15.2 lists I/O pins used for the ICU.

Table 15.2 ICU I/O Pins

Pin Name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQ0 to IRQ15	Input	External interrupt request pins

15.2 Register Descriptions

15.2.1 Interrupt Request Register n (IRn) (n = 016 to 255)

Address(es): 0008 7010h to 0008 70FFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IR	Interrupt Status Flag	0: No interrupt request is generated. 1: An interrupt request is generated.	R/(W) *1
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note 1. For an edge detection interrupt source, only 0 can be written to this bit; do not write 1.
For a level detection interrupt source, neither 0 nor 1 can be written.

The IRn register indicates whether an interrupt request has been generated.
This register is provided for each interrupt vector number, and n matches the interrupt vector number.
For the correspondence between interrupt sources and interrupt vector numbers, refer to Table 15.5, Interrupt Vector Table.

IR Flag (Interrupt Status Flag)

The IR flag is a status flag indicating whether an interrupt request has been generated. This flag becomes 1 when an interrupt request is generated. To detect an interrupt request, set the interrupt enable bit of the peripheral module to enable output of the interrupt request.
An interrupt request can be detected by edge detection or level detection. For interrupts from peripheral modules, the detection method (edge detection or level detection) is determined depending on the source. Refer to Table 15.5, Interrupt Vector Table for details on the detection method for each source. For interrupts from the IRQi pin (i = 0 to 15), edge detection or level detection can be selected by setting the IRQCRi.IRQMD[1:0] bits.
The interrupt status flag for group interrupts is the ISj flag (j = 0 to 31) in the group interrupt request register (GRPBE0, GRPBL0, GRPBL1, GRPAL0, GRPAL1). When any of the ISj flags becomes 1, the IRn.IR flag corresponding to each group interrupt becomes 1. Group interrupts are detected by level detection.
Refer to section 15.4.4, Group Interrupts for details on group interrupts.

(1) Edge detection

This flag becomes 1 under the following condition:

- The IR flag becomes 1 when an external pin interrupt request is generated. For interrupt requests for peripheral modules, refer to the corresponding sections.

This flag becomes 0 under any of the following conditions:

- The IR flag becomes 0 when the interrupt request destination accepts an interrupt request.
- The IR flag becomes 0 by writing 0 to the IR flag. Note that when the interrupt request destination is the DTC or DMAC, do not write 0 to the IR flag.

(2) Level detection

This flag becomes 1 under any of the following conditions:

- The IR flag is 1 while an interrupt request for peripheral interrupts or external pin interrupts is generated. For interrupt requests for peripheral modules, refer to the corresponding sections.
- For group interrupts, the IR flag becomes 1 when the ISj flag in the group interrupt request register is 1 (interrupt request is generated) while the ENj bit in the group interrupt request enable register is 1 (enabled) (j = 0 to 31).

This flag becomes 0 under any of the following conditions:

- The IR flag becomes 0 by clearing output of the peripheral module interrupt request. The IR flag does not become 0 when the interrupt request destination accepts the interrupt request. For interrupt requests for peripheral modules, refer to the corresponding sections.
- For group interrupts, the IR flag becomes 1 when the ENj bit in the group interrupt request enable register is 0 (disabled) or when the ISj flag in the group interrupt request register is 0 (interrupt request is not generated).

When level detection is selected for detecting external pin interrupts, set the input level of the IRQi pin to high (i = 0 to 15) to cancel the external pin interrupt that has occurred. When level detection is selected, do not write to the IR flag.

15.2.2 Interrupt Request Enable Register m (IERm) (m = 02h to 1Fh)

Address(es): 0008 7202h to 0008 721Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IEN0	Interrupt Request Enable 0	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b1	IEN1	Interrupt Request Enable 1		R/W
b2	IEN2	Interrupt Request Enable 2		R/W
b3	IEN3	Interrupt Request Enable 3		R/W
b4	IEN4	Interrupt Request Enable 4		R/W
b5	IEN5	Interrupt Request Enable 5		R/W
b6	IEN6	Interrupt Request Enable 6		R/W
b7	IEN7	Interrupt Request Enable 7		R/W

Note: When the interrupt source of the interrupt vector number is reserved, set the corresponding bit to 0. The read value is 0.

The IERm register enables or disables output of the interrupt request to the interrupt request destination.

IENj Bit (Interrupt Request Enable j) (j = 0 to 7)

When the IENj bit is 1, an interrupt request is output to the destination. When the IENj bit is 0, an interrupt request is not output to the destination.

The IRn.IR flag (n = 016 to 255) is not affected by the IENj bit setting. Even when the IENj bit is 0, the IR flag changes according to the conditions described in section 15.2.1, Interrupt Request Register n (IRn) (n = 016 to 255).

The IERm.IENj bit is provided for each interrupt vector number.

Refer to Table 15.5, Interrupt Vector Table for the correspondence between interrupt sources and the IERm.IENj bit.

Note that m and j can be calculated by the following formula:

m = quotient when n divided by 8

n = remainder when n divided by 8

Refer to section 15.7.3.1, Interrupt Request Destination Setting Procedure for the procedure to set the IERm.IENj bit to select the interrupt request destination.

15.2.3 Interrupt Source Priority Register r (IPRr) (r = 000 to 255)

Address(es): 0008 7300h to 0008 73FFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	IPR[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IPR[3:0]	Interrupt Priority Level Select	b3 b0 0 0 0 0: Level 0 (interrupt disabled) *1 0 0 0 1: Level 1 0 0 1 0: Level 2 0 0 1 1: Level 3 0 1 0 0: Level 4 0 1 0 1: Level 5 0 1 1 0: Level 6 0 1 1 1: Level 7 1 0 0 0: Level 8 1 0 0 1: Level 9 1 0 1 0: Level 10 1 0 1 1: Level 11 1 1 0 0: Level 12 1 1 0 1: Level 13 1 1 1 0: Level 14 1 1 1 1: Level 15 (highest)	R/W
b7 to b4	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note 1. For an interrupt source of a fast interrupt, even when the IPR[3:0] bits are set to level 0, the priority level is level 15.

The IPRr register sets the interrupt priority level of an interrupt source that is assigned to the corresponding interrupt vector number.

IPR[3:0] Bits (Interrupt Priority Level Select)

The IPR[3:0] bits select the interrupt priority level of the corresponding interrupt source.
The priority level selected by the IPR[3:0] bits is used for only determining the priority level of interrupt requests to the CPU. It does not affect transfer requests to the DTC and DMAC.
The CPU accepts only interrupt requests that have the higher priority level than the processor interrupt priority level indicated by the PSW.IPL[3:0] bits.
When multiple interrupt requests are concurrently generated, the priority levels selected by the corresponding IPR[3:0] bits are compared. When multiple interrupt requests that have the same priority level are concurrently generated, the interrupt request that has the smallest interrupt vector number has priority.
Write to this register while the corresponding IERm.IENj bit is 0 (interrupt request is disabled) (j = 0 to 7; m = 02h to 1Fh).
Refer to Table 15.5, Interrupt Vector Table for the correspondence between interrupt vectors and the IPRr register. Note that r matches the vector number when the interrupt vector number is 32 or greater.

15.2.4 Fast Interrupt Set Register (FIR)

Address(es): 0008 72F0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FIEN	—	—	—	—	—	—	—	FVCT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	FVCT[7:0]	Fast Interrupt Vector Number	Set the vector number of an interrupt source that is assigned to a fast interrupt.	R/W
b14 to b8	—	Reserved	The read value is 0. The write value should be 0.	R/W
b15	FIEN	Fast Interrupt Enable	0: Fast interrupt is disabled 1: Fast interrupt is enabled	R/W

The FIR register sets an interrupt source that is handled as the fast interrupt.
The fast interrupt is enabled only when the destination is the CPU. When the destination is the DTC or DMAC, the DTC or DMA transfer request is not affected by setting the interrupt vector number as the fast interrupt.
Write to this register while the corresponding IERm.IENj bit is 0 (j = 0 to 7; m = 02h to 1Fh).
Refer to section 15.9, Fast interrupt for details on the fast interrupt.

FVCT[7:0] Bits (Fast Interrupt Vector Number)

The FVCT[7:0] bits set the interrupt vector number of an interrupt source for the fast interrupt.
Refer to Table 15.5, Interrupt Vector Table for interrupt vector numbers that can be set in the FVCT[7:0] bits. Do not set an interrupt vector number that is reserved.

FIEN Bit (Fast Interrupt Enable)

The FIEN bit enables the fast interrupt to be used.
When the FIEN bit is 1, the interrupt source that is assigned to the interrupt vector number set by the FVCT[7:0] bits is handled as the fast interrupt.
When an interrupt request of the interrupt vector number set by the FVCT[7:0] bits is generated to the CPU while the FIEN bit is 1, an interrupt request is output to the CPU as the fast interrupt regardless of the IPRr register setting (r = 000 to 255). Note that the IPRr register setting is required when using the fast interrupt to exit software standby mode. Refer to section 15.10.3, Exiting Software Standby Mode for details.

15.2.5 Software Interrupt Generation Register (SWINTR)

Address(es): 0008 72E0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SWINT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SWINT	Software Interrupt Generation	The read value is 0. When writing 1 to this bit, a software interrupt request is generated. Writing 0 to this bit has no effect.	R/(W)
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

The SWINTR register controls generation of a software interrupt request.

SWINT Bit (Software Interrupt Generation)

When the SWINT bit is set to 1, a software interrupt request (SWINT) is generated, and the IR027.IR flag becomes 1. A software interrupt request (SWINT) can be set as a DTC trigger, but it cannot be set as a DMAC trigger.

15.2.6 Software Interrupt 2 Generation Register (SWINT2R)

Address(es): 0008 72E1h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SWINT2
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SWINT2	Software Interrupt 2 Generation	The read value is 0. When writing 1 to this bit, a software interrupt request is generated. Writing 0 to this bit has no effect.	R/(W)
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

The SWINT2R register controls generation of software interrupt request 2.

SWINT2 Bit (Software Interrupt 2 Generation)

When the SWINT2 bit is set to 1, software interrupt request 2 (SWINT2) is generated, and the IR026.IR flag becomes 1. Software interrupt request 2 (SWINT2) can be set as the DTC trigger, but it cannot be set as the DMAC trigger.

15.2.7 DTC Transfer Request Enable Register n (DTCERn) (n = 026 to 255)

Address(es): 0008 711Ah to 0008 71FFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTCE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DTCE	DTC Transfer Request Enable	0: The corresponding interrupt source is not selected as the DTC trigger. 1: The corresponding interrupt source is selected as the DTC trigger.	R/W
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

The DTCERn register selects the interrupt source corresponding to interrupt vector number n as the DTC trigger.
Do not set the same interrupt source for both the DTC trigger and DMAC trigger. Refer to Table 15.5, Interrupt Vector Table for the correspondence between interrupt sources and interrupt vector numbers and interrupt sources that can be used as the DTC trigger.

DTCE Bit (DTC Transfer Request Enable)

When the DTCE bit is set to 1, the corresponding interrupt source is selected as the DTC trigger.
This bit becomes 1 under the following condition:

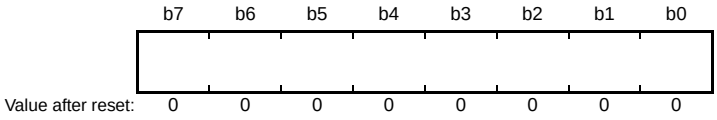
- 1 is written to the DTCE bit

This bit becomes 0 under any of the following conditions:

- The specified number of transfers is completed (for the chain transfer, the number of transfers for the last chain transfer is completed)
- 0 is written to the DTCE bit

15.2.8 DMAC trigger Select Register m (DMRSRm) (m = DMAC channel number)

Address(es): DMRSR0 0008 7400h, DMRSR1 0008 7404h, DMRSR2 0008 7408h, DMRSR3 0008 740Ch, DMRSR4 0008 7410h, DMRSR5 0008 7414h, DMRSR6 0008 7418h, DMRSR7 0008 741Ch



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	These bits set the interrupt vector number of the interrupt source as the DMAC trigger.	R/W

The DMRSRm register sets an interrupt source as the DMACm trigger.

Do not set the same vector number for multiple DMRSRm registers. Do not set the same interrupt source for both the DTC trigger and DMAC trigger. Otherwise, the operation is not guaranteed.

Set the interrupt vector number of an interrupt source used as the DMAC trigger in the DMRSRm register. Do not set vector numbers of interrupt sources that cannot be used as the DMAC trigger.

Refer to Table 15.5, Interrupt Vector Table for interrupt vector numbers of interrupt sources.

Write the DMRSRm register while the DMACm.DMCNT.DTE bit is 0.

15.2.9 IRQ Control Register i (IRQCRi) (i = 0 to 15)

Address(es): 0008 7500h to 0008 750Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	IRQMD[1:0]		—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	The read value is 0. The write value should be 0.	R/W
b3, b2	IRQMD[1:0]	IRQ Detection Select	b3 b2 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Rising and falling edges	R/W
b7 to b4	—	Reserved	The read value is 0. The write value should be 0.	R/W

The IRQCRi register selects the detection method for external pin interrupts.
Write to this register while the corresponding IERm.IENj bit is 0 (j = 0 to 7; m = 02h to 1Fh). After writing to this register, set the IRn.IR flag to 0, and then set the IENj bit to 1 (n = 016 to 255). Note that setting the IR flag to 0 is not required when changing the detection method to level detection.

IRQMD[1:0] Bits (IRQ Detection Select)

The IRQMD[1:0] bits set the detection method for the IRQi pin interrupt (i = 0 to 15).
Refer to section 15.7.4, Setting the External Pin Interrupt for the procedure to set the external pin interrupts.

15.2.10 IRQ Pin Digital Filter Enable Register 0 (IRQFLTE0)

Address(es): 0008 7520h

	b7	b6	b5	b4	b3	b2	b1	b0
	FLTEN7	FLTEN6	FLTEN5	FLTEN4	FLTEN3	FLTEN2	FLTEN1	FLTEN0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FLTEN0	IRQ0 Digital Filter Enable	0: Digital filter is disabled. 1: Digital filter is enabled.	R/W
b1	FLTEN1	IRQ1 Digital Filter Enable		R/W
b2	FLTEN2	IRQ2 Digital Filter Enable		R/W
b3	FLTEN3	IRQ3 Digital Filter Enable		R/W
b4	FLTEN4	IRQ4 Digital Filter Enable		R/W
b5	FLTEN5	IRQ5 Digital Filter Enable		R/W
b6	FLTEN6	IRQ6 Digital Filter Enable		R/W
b7	FLTEN7	IRQ7 Digital Filter Enable		R/W

The IRQFLTE0 register enables and disables digital filters for pins IRQ0 to IRQ7.

FLTENi Bits (IRQi Digital Filter Enable) (i = 0 to 7)

When the FLTENi bit is 1, the digital filter for the IRQi pin is enabled. When the FLTENi bit is 0, the digital filter for the IRQi pin is disabled.

The signal input to the IRQi pin is sampled at the sampling clock set by the IRQFLTC0.FCLKSELi[1:0] bits. When the sampled signal level does not match three times in a row, the input signal is removed.

Refer to section 15.7.6, Digital Filter for details on the digital filter.

15.2.11 IRQ Pin Digital Filter Enable Register 1 (IRQFLTE1)

Address(es): 0008 7521h

	b7	b6	b5	b4	b3	b2	b1	b0
	FLTEN15	FLTEN14	FLTEN13	FLTEN12	FLTEN11	FLTEN10	FLTEN9	FLTEN8
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FLTEN8	IRQ8 Digital Filter Enable	0: Digital filter is disabled. 1: Digital filter is enabled.	R/W
b1	FLTEN9	IRQ9 Digital Filter Enable		R/W
b2	FLTEN10	IRQ10 Digital Filter Enable		R/W
b3	FLTEN11	IRQ11 Digital Filter Enable		R/W
b4	FLTEN12	IRQ12 Digital Filter Enable		R/W
b5	FLTEN13	IRQ13 Digital Filter Enable		R/W
b6	FLTEN14	IRQ14 Digital Filter Enable		R/W
b7	FLTEN15	IRQ15 Digital Filter Enable		R/W

The IRQFLTE1 register enables or disables digital filters for pins IRQ8 to IRQ15.

FLTENi Bit (IRQi Digital Filter Enable) (i = 8 to 15)

When the FLTENi bit is 1, the digital filter for the IRQi pin is enabled. When the FLTENi bit is 0, the digital filter for the IRQi pin is disabled.

The signal input to the IRQi pin is sampled at the sampling clock set by the IRQFLTC1.FCLKSELi[1:0] bits. When the sampled signal level does not match three times in a row, the input signal is removed.

Refer to section 15.7.6, Digital Filter for details on the digital filter.

15.2.12 IRQ Pin Digital Filter Setting Register 0 (IRQFLTC0)

Address(es): 0008 7528h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FCLKSEL7[1:0]		FCLKSEL6[1:0]		FCLKSEL5[1:0]		FCLKSEL4[1:0]		FCLKSEL3[1:0]		FCLKSEL2[1:0]		FCLKSEL1[1:0]		FCLKSEL0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	FCLKSEL0[1:0]	IRQ0 Digital Filter Sampling Clock	0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
b3, b2	FCLKSEL1[1:0]	IRQ1 Digital Filter Sampling Clock		R/W
b5, b4	FCLKSEL2[1:0]	IRQ2 Digital Filter Sampling Clock		R/W
b7, b6	FCLKSEL3[1:0]	IRQ3 Digital Filter Sampling Clock		R/W
b9, b8	FCLKSEL4[1:0]	IRQ4 Digital Filter Sampling Clock		R/W
b11, b10	FCLKSEL5[1:0]	IRQ5 Digital Filter Sampling Clock		R/W
b13, b12	FCLKSEL6[1:0]	IRQ6 Digital Filter Sampling Clock		R/W
b15, b14	FCLKSEL7[1:0]	IRQ7 Digital Filter Sampling Clock		R/W

The IRQFLTC0 register sets the sampling clock of the digital filter for pins IRQ0 to IRQ7.

FCLKSELi[1:0] Bits (IRQi Digital Filter Sampling Clock) (i = 0 to 7)

The FCLKSELi[1:0] bits set the sampling clock of the digital filter for the IRQi pin.

Refer to section 15.7.6, Digital Filter for details on the digital filter.

15.2.13 IRQ Pin Digital Filter Setting Register 1 (IRQFLTC1)

Address(es): 0008 752Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FCLKSEL15[1:0]		FCLKSEL14[1:0]		FCLKSEL13[1:0]		FCLKSEL12[1:0]		FCLKSEL11[1:0]		FCLKSEL10[1:0]		FCLKSEL9[1:0]		FCLKSEL8[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	FCLKSEL8[1:0]	IRQ8 Digital Filter Sampling Clock	0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
b3, b2	FCLKSEL9[1:0]	IRQ9 Digital Filter Sampling Clock		R/W
b5, b4	FCLKSEL10[1:0]	IRQ10 Digital Filter Sampling Clock		R/W
b7, b6	FCLKSEL11[1:0]	IRQ11 Digital Filter Sampling Clock		R/W
b9, b8	FCLKSEL12[1:0]	IRQ12 Digital Filter Sampling Clock		R/W
b11, b10	FCLKSEL13[1:0]	IRQ13 Digital Filter Sampling Clock		R/W
b13, b12	FCLKSEL14[1:0]	IRQ14 Digital Filter Sampling Clock		R/W
b15, b14	FCLKSEL15[1:0]	IRQ15 Digital Filter Sampling Clock		R/W

The IRQFLTC1 register sets the sampling clock of the digital filter for pins IRQ8 to IRQ15.

FCLKSELi[1:0] Bits (IRQi Digital Filter Sampling Clock) (i = 8 to 15)

The FCLKSELi[1:0] bits set the sampling clock of the digital filter for the IRQi pin.

Refer to section 15.7.6, Digital Filter for details on the digital filter.

15.2.14 Non-Maskable Interrupt Status Register (NMISR)

Address(es): 0008 7580h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ECCRAMST	LVD2ST	LVD1ST	IWDTST	WDTST	OSTST	NMIST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NMIST	NMI Status Flag	0: NMI pin interrupt is not requested. 1: NMI pin interrupt is requested.	R
b1	OSTST	Oscillation Stop Detection Interrupt Status Flag	0: Oscillation stop detection interrupt is not requested. 1: Oscillation stop detection interrupt is requested.	R
b2	WDTST	WDT Underflow/Refresh Error Status Flag	0: WDT underflow/refresh error interrupt is not requested. 1: WDT underflow/refresh error interrupt is requested.	R
b3	IWDTST	IWDT Underflow/Refresh Error Status Flag	0: IWDT underflow/refresh error interrupt is not requested. 1: IWDT underflow/refresh error interrupt is requested.	R
b4	LVD1ST	Voltage Monitoring 1 Interrupt Status Flag	0: Voltage monitoring 1 interrupt is not requested. 1: Voltage monitoring 1 interrupt is requested.	R
b5	LVD2ST	Voltage Monitoring 2 Interrupt Status Flag	0: Voltage monitoring 2 interrupt is not requested. 1: Voltage monitoring 2 interrupt is requested.	R
b6	ECCRAMST	RAM Error Interrupt Status Flag	0: RAM error interrupt is not requested. 1: RAM error interrupt is requested.	R
b7	—	Reserved	This bit is read as 0 and cannot be modified.	R

The NMISR register indicates whether a non-maskable interrupt request has occurred.
Each flag in the NMISR register is not affected by the setting of the corresponding bit in the NMIER register.
In the non-maskable interrupt handler, read the NMISR register to check if the other non-maskable interrupt has occurred. Confirm that all the status flags are 0 before exiting the interrupt handler.

NMIST Flag (NMI Status Flag)

The NMIST flag indicates whether an NMI pin interrupt request has been generated.
This flag is read-only. To set the NMIST flag to 0, set the NMICLR.NMICLR bit to 1.
This flag becomes 1 under the following condition:

- An edge set in the NMICR.NMIMD bit is input to the NMI pin

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.NMICLR bit

OSTST Flag (Oscillation Stop Detection Interrupt Status Flag)

The OSTST flag indicates whether an oscillation stop detection interrupt request has been generated.
The OSTST flag is read-only. To set the OSTST flag to 0, set the NMICLR.OSTCLR bit to 1.
This flag becomes 1 under the following condition:

- An oscillation stop detection interrupt occurs

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.OSTCLR bit

WDTST Flag (WDT Underflow/Refresh Error Status Flag)

The WDTST flag indicates whether an WDT underflow/refresh error interrupt request is generated.
The WDTST flag is read-only. To set the WDTST flag to 0, set the NMICLR.WDTCLR bit to 1.
This flag becomes 1 under the following condition:

- A WDT underflow/refresh error interrupt occurs while the WDTRCR.RSTIRQS bit is 0

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.WDTCLR bit

IWDTST Flag (IWDT Underflow/Refresh Error Status Flag)

The IWDTST flag indicates whether an IWDT underflow/refresh error interrupt request is generated.
The IWDTST flag is read-only. To set the IWDTST flag to 0, set the NMICLR.IWDTCLR bit to 1.
This flag becomes 1 under the following condition:

- An IWDT underflow/refresh error interrupt occurs while the IWDTRCR.RSTIRQS bit is 0

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.IWDTCLR bit

LVD1ST Flag (Voltage Monitoring 1 Interrupt Status Flag)

The LVD1ST flag indicates whether a voltage monitoring 1 interrupt request is generated.
The LVD1ST flag is read-only. To set the LVD1ST flag to 0, set the NMICLR.LVD1CLR bit to 1.
This flag becomes 1 under the following condition:

- A voltage monitoring 1 interrupt occurs while the LVD1CR1.LVD1IRQSEL bit is 0

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.LVD1CLR bit

LVD2ST Flag (Voltage Monitoring 2 Interrupt Status Flag)

The LVD2ST flag indicates whether a voltage monitoring 2 interrupt request is generated.
The LVD2ST flag is read-only. To set the LVD2ST flag to 0, set the NMICLR.LVD2CLR bit to 1.
This flag becomes 1 under the following condition:

- A voltage monitoring 2 interrupt occurs while the LVD2CR1.LVD2IRQSEL bit is 0

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.LVD2CLR bit

ECCRAMST Flag (RAM Error Interrupt Status Flag)

The ECCRAMST flag indicates whether an RAM error interrupt request is generated from the RAM.
The ECCRAMST flag is read-only. Clear the error status flag of the RAM or the ECCRAM. Refer to section 61.3.4, RAM Error Interrupt Function for details.
This flag becomes 1 under the following condition:

- A parity check error interrupt occurs
- An ECC error interrupt occurs

This flag becomes 0 under the following condition:

- 0 is written to the RAMSTS.RAMERR flag
- 0 is written to the ECCRAM2STS.ECC2ERR flag
- 0 is written to the ECCRAM1STS.ECC1ERR flag

15.2.15 Non-Maskable Interrupt Enable Register (NMIER)

Address(es): 0008 7581h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ECCRAMEN	LVD2EN	LVD1EN	IWDTEN	WDTEN	OSTEN	NMIEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NMIEN	NMI Pin Interrupt Enable	0: NMI pin interrupt is disabled. 1: NMI pin interrupt is enabled.	R/(W) *1
b1	OSTEN	Oscillation Stop Detection Interrupt Enable	0: Oscillation stop detection interrupt is disabled. 1: Oscillation stop detection interrupt is enabled.	R/(W) *1
b2	WDTEN	WDT Underflow/Refresh Error Enable	0: WDT underflow/refresh error interrupt is disabled. 1: WDT underflow/refresh error interrupt is enabled.	R/(W) *1
b3	IWDTEN	IWDT Underflow/Refresh Error Enable	0: IWDT underflow/refresh error interrupt is disabled. 1: IWDT underflow/refresh error interrupt is enabled.	R/(W) *1
b4	LVD1EN	Voltage Monitoring 1 Interrupt Enable	0: Voltage monitoring 1 interrupt is disabled. 1: Voltage monitoring 1 interrupt is enabled.	R/(W) *1
b5	LVD2EN	Voltage Monitoring 2 Interrupt Enable	0: Voltage monitoring 2 interrupt is disabled. 1: Voltage monitoring 2 interrupt is enabled.	R/(W) *1
b6	ECCRAMEN	RAM Error Interrupt Enable	0: RAM error interrupt is disabled. 1: RAM error interrupt is enabled.	R/(W) *1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Once this bit is set to 1, it cannot be set to 0 by software.

The NMIER register enables and disables generation of non-maskable interrupt requests. When each bit is 1, the corresponding interrupt source is used as a non-maskable interrupt.

NMIEN Bit (NMI Pin Interrupt Enable)

The NMIEN bit enables and disables using the NMI pin interrupt.

OSTEN Bit (Oscillation Stop Detection Interrupt Enable)

The OSTEN bit enables and disables generation of a non-maskable interrupt by the oscillation stop detection interrupt. When the oscillation stop detection interrupt is used as a maskable interrupt, leave this bit set to 0.

WDTEN Bit (WDT Underflow/Refresh Error Enable)

The WDTEN bit enables and disables generation of a non-maskable interrupt by the WDT underflow/refresh error interrupt.

When the WDT underflow/refresh error interrupt is used as a maskable interrupt, leave this bit set to 0.

IWDTEN Bit (IWDT Underflow/Refresh Error Enable)

The IWDTEN bit enables and disables generation of a non-maskable interrupt by the IWDT underflow/refresh error interrupt.

When the IWDT underflow/refresh error interrupt is used as a maskable interrupt, leave this bit set to 0.

LVD1EN Bit (Voltage Monitoring 1 Interrupt Enable)

The LVD1EN bit enables and disables generation of a non-maskable interrupt by the voltage monitoring 1 interrupt.

When the voltage monitoring 1 interrupt is used as a maskable interrupt, leave this bit set to 0.

LVD2EN Bit (Voltage Monitoring 2 Interrupt Enable)

The LVD2EN bit enables and disables generation of a non-maskable interrupt by the voltage monitoring 2 interrupt. When the voltage monitoring 2 interrupt is used as a maskable interrupt, leave this bit set to 0.

ECCRAMEN Bit (RAM Error Interrupt Enable)

The ECCRAMEN bit enables and disables generation of a non-maskable interrupt by the RAM error interrupt. When the RAM error interrupt is used as a maskable interrupt, leave this bit set to 0.

15.2.16 Non-Maskable Interrupt Status Clear Register (NMICLR)

Address(es): 0008 7582h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LVD2CLR	LVD1CLR	IWDTCLR	WDTCLR	OSTCLR	NMICLR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NMICLR	NMI Clear	The read value is 0. When 1 is written to this bit, the NMISR.NMIST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b1	OSTCLR	OST Clear	The read value is 0. When 1 is written to this bit, the NMISR.OSTST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b2	WDTCLR	WDT Clear	The read value is 0. When 1 is written to this bit, the NMISR.WDTST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b3	IWDTCLR	IWDT Clear	The read value is 0. When 1 is written to this bit, the NMISR.IWDTST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b4	LVD1CLR	LVD1 Clear	The read value is 0. When 1 is written to this bit, the NMISR.LVD1ST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b5	LVD2CLR	LVD2 Clear	The read value is 0. When 1 is written to this bit, the NMISR.LVD2ST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b7-b6	—	Reserved	The read value is 0. The write value should be 0.	R/W

The NMICLR register clears each flag in the NMISR register.
When writing 1 to a bit in this register, the corresponding status flag becomes 0.

15.2.17 NMI Pin Interrupt Control Register (NMICR)

Address(es): 0008 7583h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	NMIMD	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	The read value is 0. The write value should be 0.	R/W
b3	NMIMD	NMI Detection Set	0: Falling edge 1: Rising edge	R/W
b7 to b4	—	Reserved	The read value is 0. The write value should be 0.	R/W

The NMICR register selects a detection method for the NMI pin interrupt.
Write to the NMICR register while the NMIER.NMIEN bit is 0.

15.2.18 NMI Pin Digital Filter Enable Register (NMIFLTE)

Address(es): 0008 7590h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	NFLTEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFLTEN	NMI Digital Filter Enable	0: Digital filter is disabled. 1: Digital filter is enabled.	R/W
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

The NMIFLTE register enables or disables the digital filter for the NMI pin.

NFLTEN Bit (NMI Digital Filter Enable)

When the NFLTEN bit is 1, the digital filter is enabled. When the NFLTEN bit is 0, the digital filter is disabled.
The signal input to the NMI pin is sampled at the sampling clock set by the NMIFLTC.NFCLKSEL[1:0] bits. When the sampled signal level does not match three times in a row, the input signal is removed.
Refer to section 15.7.6, Digital Filter for details on the digital filter.

15.2.19 NMI Pin Digital Filter Setting Register (NMIFLTC)

Address(es): 0008 7594h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	NFCLKSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	NFCLKSEL[1:0]	NMI Digital Filter Sampling Clock	b1 b0 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
b7 to b2	—	Reserved	The read value is 0. The write value should be 0.	R/W

The NMIFLTC register sets the sampling clock of the digital filter for the NMI pin.

NFCLKSEL[1:0] Bits (NMI Digital Filter Sampling Clock)

The NFCLKSEL[1:0] bits select the sampling clock of the digital filter for the NMI pin.
Refer to section 15.7.6, Digital Filter for details on the digital filter.

15.2.20 Group BE0 Interrupt Request Register (GRPBE0)
Group BL0/BL1 Interrupt Request Register (GRPBL0/GRPBL1)
Group AL0/AL1 Interrupt Request Register (GRPAL0/GRPAL1)

Address(es): GRPBE0 0008 7600h, GRPBL0 0008 7630h, GRPBL1 0008 7634h, GRPAL0 0008 7830h, GRPAL1 0008 7834h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IS31	IS30	IS29	IS28	IS27	IS26	IS25	IS24	IS23	IS22	IS21	IS20	IS19	IS18	IS17	IS16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IS15	IS14	IS13	IS12	IS11	IS10	IS9	IS8	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IS0	Interrupt Status Flag 0	0: Interrupt is not requested. 1: Interrupt is requested.	R
b1	IS1	Interrupt Status Flag 1		R
b2	IS2	Interrupt Status Flag 2		R
b3	IS3	Interrupt Status Flag 3		R
b4	IS4	Interrupt Status Flag 4		R
b5	IS5	Interrupt Status Flag 5		R
b6	IS6	Interrupt Status Flag 6		R
b7	IS7	Interrupt Status Flag 7		R
b8	IS8	Interrupt Status Flag 8		R
b9	IS9	Interrupt Status Flag 9		R
b10	IS10	Interrupt Status Flag 10		R
b11	IS11	Interrupt Status Flag 11		R
b12	IS12	Interrupt Status Flag 12		R
b13	IS13	Interrupt Status Flag 13		R
b14	IS14	Interrupt Status Flag 14		R
b15	IS15	Interrupt Status Flag 15		R
b16	IS16	Interrupt Status Flag 16		R
b17	IS17	Interrupt Status Flag 17		R
b18	IS18	Interrupt Status Flag 18		R
b19	IS19	Interrupt Status Flag 19		R
b20	IS20	Interrupt Status Flag 20		R
b21	IS21	Interrupt Status Flag 21		R
b22	IS22	Interrupt Status Flag 22		R
b23	IS23	Interrupt Status Flag 23		R
b24	IS24	Interrupt Status Flag 24		R
b25	IS25	Interrupt Status Flag 25		R
b26	IS26	Interrupt Status Flag 26		R
b27	IS27	Interrupt Status Flag 27		R
b28	IS28	Interrupt Status Flag 28		R
b29	IS29	Interrupt Status Flag 29		R
b30	IS30	Interrupt Status Flag 30		R
b31	IS31	Interrupt Status Flag 31		R

These registers indicate the status of each interrupt request of group interrupt sources.
The GRPBE0 register contains the statuses of interrupt sources that are detected by edge detection and use PCLKB as the operating clock.
Registers GRPBL0 and GRPBL1 contain the statuses of interrupt sources that are detected by level detection and use PCLKB as the operating clock.
Registers GRPAL0 and GRPAL1 contain the statuses of interrupt sources that are detected by level detection and use PCLKB as the operating clock.
These five registers are collectively referred to as the “group interrupt request register”.
Refer to section 15.4.4, Group Interrupts for details on group interrupts.

ISj Flag (Interrupt Status Flag j) (j = 0 to 31)

The ISj flag indicates the status of interrupt sources assigned to group interrupts.
The ISj flag becomes 1 only when the corresponding ENj bit in the group interrupt request enable register is 1. When any of the ISj flags becomes 1, the IRn.IR flag corresponding to the group interrupt becomes 1 (n = 016 to 255).

(1) Group BE0

This flag becomes 1 under the following condition:

- The GRPBE0.ISj flag becomes 1 when the corresponding peripheral module interrupt request is generated while the GENBE0.ENj bit is 1 (j = 0 to 31).

This flag becomes 0 under the following condition:

- The GRPBE0.ISj flag becomes 0 when the GCRBE0.CLRj bit is set to 1 (j = 0 to 31).

(2) Group BL0/BL1

This flag becomes 1 under the following condition:

- The GRPBL0/GRPBL1.ISj flag becomes 1 while the corresponding peripheral module interrupt request is generated when the GENBL0/GENBL1.ENj bit is 1 (j = 0 to 31).

This flag becomes 0 under any of the following conditions:

- The GRPBL0/GRPBL1.ISj flag becomes 0 when the corresponding peripheral module interrupt request is cleared.
- The GRPBL0/GRPBL1.ISj flag becomes 0 when the GENBL0/GENBL1.ENj bit is set to 0.

(3) Group AL0/AL1

This flag becomes 1 under the following condition:

- The GRPAL0/GRPAL1.ISj flag becomes 1 while the corresponding peripheral module interrupt request is generated when the GENAL0/GENAL1.ENj bit is 1 (j = 0 to 31).

This flag becomes 0 under any of the following conditions:

- The GRPAL0/GRPAL1.ISj flag becomes 0 when the corresponding peripheral module interrupt request is cleared.
- The GRPAL0/GRPAL1.ISj flag becomes 0 when the GENAL0/GENAL1.ENj bit is set to 0.

15.2.21 Group BE0 Interrupt Request Enable Register (GENBE0)
Group BL0/BL1 Interrupt Request Enable Register (GENBL0/GENBL1)
Group AL0/AL1 Interrupt Request Enable Register (GENAL0/GENAL1)

Address(es): GENBE0 0008 7640h, GENBL0 0008 7670h, GENBL1 0008 7674h, GENAL0 0008 7870h, GENAL1 0008 7874h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	EN31	EN30	EN29	EN28	EN27	EN26	EN25	EN24	EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EN0	Interrupt Request Enable 0	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b1	EN1	Interrupt Request Enable 1		R/W
b2	EN2	Interrupt Request Enable 2		R/W
b3	EN3	Interrupt Request Enable 3		R/W
b4	EN4	Interrupt Request Enable 4		R/W
b5	EN5	Interrupt Request Enable 5		R/W
b6	EN6	Interrupt Request Enable 6		R/W
b7	EN7	Interrupt Request Enable 7		R/W
b8	EN8	Interrupt Request Enable 8		R/W
b9	EN9	Interrupt Request Enable 9		R/W
b10	EN10	Interrupt Request Enable 10		R/W
b11	EN11	Interrupt Request Enable 11		R/W
b12	EN12	Interrupt Request Enable 12		R/W
b13	EN13	Interrupt Request Enable 13		R/W
b14	EN14	Interrupt Request Enable 14		R/W
b15	EN15	Interrupt Request Enable 15		R/W
b16	EN16	Interrupt Request Enable 16		R/W
b17	EN17	Interrupt Request Enable 17		R/W
b18	EN18	Interrupt Request Enable 18		R/W
b19	EN19	Interrupt Request Enable 19		R/W
b20	EN20	Interrupt Request Enable 20		R/W
b21	EN21	Interrupt Request Enable 21		R/W
b22	EN22	Interrupt Request Enable 22		R/W
b23	EN23	Interrupt Request Enable 23		R/W
b24	EN24	Interrupt Request Enable 24		R/W
b25	EN25	Interrupt Request Enable 25		R/W
b26	EN26	Interrupt Request Enable 26		R/W
b27	EN27	Interrupt Request Enable 27		R/W
b28	EN28	Interrupt Request Enable 28		R/W
b29	EN29	Interrupt Request Enable 29		R/W
b30	EN30	Interrupt Request Enable 30		R/W
b31	EN31	Interrupt Request Enable 31		R/W

Note: When a bit has no corresponding interrupt source (bit is reserved), set the bit to 0.

These registers select whether the ISj flag in the group interrupt request register is set to 1 when each interrupt request for group interrupt sources is generated. These five registers are collectively referred to as the “group interrupt request enable register”.
Registers GENBE0, GENBL0/GENBL1, and GENAL0/GENAL1 control the ISj flag in registers GRPBE0, GRPBL0/GRPBL1, and GRPAL0/GRPAL1 register, respectively.
Refer to section 15.4.4, Group Interrupts for details on group interrupts.

ENj Bits (Interrupt Request Enable j) (j = 0 to 31)

The ENj bits select whether the corresponding ISj flag in the group interrupt request register is set to 1 when an interrupt request assigned to group interrupts is generated.

(1) Group BE0

When a peripheral module interrupt request is generated while the corresponding GENBE0.ENj bit is 1, the GRPBE0.ISj flag becomes 1 (j = 0 to 31). When the ENj bit is 0, the ISj flag does not become 1.
Even when the ENj bit is set to 0, the ISj flag does not change.

(2) Group BL0/BL1

When a peripheral module interrupt request is generated while the corresponding GENBL0/GENBL1.ENj bit is 1, the GRPBL0/GRPBL1.ISj flag becomes 1 (j = 0 to 31). When the ENj bit is 0, the ISj flag does not become 1.
When the ENj bit is set to 0, the ISj flag becomes 0.

(3) Group AL0/AL1

When a peripheral module interrupt request is generated while the corresponding GENAL0/GENAL1.ENj bit is 1, the GRPAL0/GRPAL1.ISj flag becomes 1 (j = 0 to 31). When the ENj bit is 0, the ISj flag does not become 1.
When the ENj bit is set to 0, the ISj flag becomes 0.

15.2.22 Group BE0 Interrupt Clear Register (GCRBE0)

Address(es): 0008 7680h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CLR31	CLR30	CLR29	CLR28	CLR27	CLR26	CLR25	CLR24	CLR23	CLR22	CLR21	CLR20	CLR19	CLR18	CLR17	CLR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CLR15	CLR14	CLR13	CLR12	CLR11	CLR10	CLR9	CLR8	CLR7	CLR6	CLR5	CLR4	CLR3	CLR2	CLR1	CLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CLR0	Interrupt Source Clear 0	The read value is 0. When the CLRj bit is set to 1, the corresponding interrupt status flag (GRPBE0.ISj) becomes 0 (j = 0 to 31). Writing 0 has no effect.	R/(W)
b1	CLR1	Interrupt Source Clear 1		R/(W)
b2	CLR2	Interrupt Source Clear 2		R/(W)
b3	CLR3	Interrupt Source Clear 3		R/(W)
b4	CLR4	Interrupt Source Clear 4		R/(W)
b5	CLR5	Interrupt Source Clear 5		R/(W)
b6	CLR6	Interrupt Source Clear 6		R/(W)
b7	CLR7	Interrupt Source Clear 7		R/(W)
b8	CLR8	Interrupt Source Clear 8		R/(W)
b9	CLR9	Interrupt Source Clear 9		R/(W)
b10	CLR10	Interrupt Source Clear 10		R/(W)
b11	CLR11	Interrupt Source Clear 11		R/(W)
b12	CLR12	Interrupt Source Clear 12		R/(W)
b13	CLR13	Interrupt Source Clear 13		R/(W)
b14	CLR14	Interrupt Source Clear 14		R/(W)
b15	CLR15	Interrupt Source Clear 15		R/(W)
b16	CLR16	Interrupt Source Clear 16		R/(W)
b17	CLR17	Interrupt Source Clear 17		R/(W)
b18	CLR18	Interrupt Source Clear 18		R/(W)
b19	CLR19	Interrupt Source Clear 19		R/(W)
b20	CLR20	Interrupt Source Clear 20		R/(W)
b21	CLR21	Interrupt Source Clear 21		R/(W)
b22	CLR22	Interrupt Source Clear 22		R/(W)
b23	CLR23	Interrupt Source Clear 23		R/(W)
b24	CLR24	Interrupt Source Clear 24		R/(W)
b25	CLR25	Interrupt Source Clear 25		R/(W)
b26	CLR26	Interrupt Source Clear 26		R/(W)
b27	CLR27	Interrupt Source Clear 27		R/(W)
b28	CLR28	Interrupt Source Clear 28		R/(W)
b29	CLR29	Interrupt Source Clear 29		R/(W)
b30	CLR30	Interrupt Source Clear 30		R/(W)
b31	CLR31	Interrupt Source Clear 31		R/(W)

Note: Write 1 to only the bit corresponding to the flag that is cleared, and write 0 to the other bits.

The GCRBE0 register clears the GRPBE0.ISj flag (j = 0 to 31).
Refer to section 15.4.4, Group Interrupts for details on group interrupts.

CLRj Bit (Interrupt Source Clear j) (j = 0 to 31)

When the GCRBE0.CLRj bit is set to 1, the GRPBE0.ISj flag becomes 0.

15.2.23 Software Configurable Interrupt B Request Register k (PIBRk) (k = 0h to Ah)

Address(es): 0008 7700h to 0008 770Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	PIR7	PIR6	PIR5	PIR4	PIR3	PIR2	PIR1	PIR0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIR0	Software Configurable Interrupt B Status Flag 0	When reading 0: Interrupt is not requested. 1: Interrupt is requested. When writing *1 0: Ignored. 1: The software configurable interrupt B status flag is cleared.	R/W
b1	PIR1	Software Configurable Interrupt B Status Flag 1		R/W
b2	PIR2	Software Configurable Interrupt B Status Flag 2		R/W
b3	PIR3	Software Configurable Interrupt B Status Flag 3		R/W
b4	PIR4	Software Configurable Interrupt B Status Flag 4		R/W
b5	PIR5	Software Configurable Interrupt B Status Flag 5		R/W
b6	PIR6	Software Configurable Interrupt B Status Flag 6		R/W
b7	PIR7	Software Configurable Interrupt B Status Flag 7		R/W

Note 1. Do not use bit manipulation instructions. If a bit manipulation instruction is used, multiple status flags may be cleared. Write 1 to the flag that is cleared and write 0 to the other flags (write to this register in bytes).

The PIBRk register is used for polling of interrupt requests of interrupt sources that is assigned to software configurable interrupt B by software. For an interrupt request of software configurable interrupt B set in the SLIBXRn or SLIBRn register, use the corresponding IRn.IR flag for polling (n = 128 to 207).
Refer to Table 15.3, Interrupt Sources for Software Configurable Interrupt B for correspondence between interrupt source numbers and interrupt sources for software configurable interrupt B.

PIRj Flag (Software Configurable Interrupt B Status Flag j) (j = 0 to 7)

When an interrupt request of interrupt sources assigned to software configurable interrupt B is generated, the corresponding PIBRk.PIRj flag becomes 1 regardless of whether the interrupt source is selected in the SLIBXRn or SLIBRn register.
Although the PIRj flag does not become 0 after the interrupt request is accepted by the destination (CPU, DTC, DMAC), generation of interrupt requests is not affected.
When using the PIRj flag for polling, set the PIRj flag to 0 by writing 1 to the flag in advance.
This flag becomes 1 under the following condition:

- An interrupt request is generated

This flag becomes 0 under the following condition:

- 1 is written to the PIBRk.PIRj flag

15.2.24 Software Configurable Interrupt A Request Register k (PIARk) (k = 0h to Bh)

Address(es): 0008 7900h to 0008 790Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	PIR7	PIR6	PIR5	PIR4	PIR3	PIR2	PIR1	PIR0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIR0	Software Configurable Interrupt A Status Flag 0	When reading 0: Interrupt is not requested. 1: Interrupt is requested. When writing *1 0: Ignored. 1: The Software Configurable interrupt A status flag is cleared.	R/W
b1	PIR1	Software Configurable Interrupt A Status Flag 1		R/W
b2	PIR2	Software Configurable Interrupt A Status Flag 2		R/W
b3	PIR3	Software Configurable Interrupt A Status Flag 3		R/W
b4	PIR4	Software Configurable Interrupt A Status Flag 4		R/W
b5	PIR5	Software Configurable Interrupt A Status Flag 5		R/W
b6	PIR6	Software Configurable Interrupt A Status Flag 6		R/W
b7	PIR7	Software Configurable Interrupt A Status Flag 7		R/W

Note 1. Do not use bit manipulation instructions. If a bit manipulation instruction is used, multiple status flags may be cleared. Write 1 to the flag that is cleared and write 0 to the other flags (write to this register in bytes).

The PIARk register is used for polling interrupt requests of interrupt sources that is assigned to software configurable interrupt A by software. For an interrupt request of software configurable interrupt A set in the SLIARn register, use the corresponding IRn.IR flag for polling.
Refer to Table 15.4, Interrupt Sources for Software Configurable Interrupt A for correspondence between interrupt source numbers and interrupt sources for software configurable interrupt A.

PIRj Flag (Software Configurable Interrupt A Status Flag j) (j = 0 to 7)

When an interrupt request of interrupt sources assigned to software configurable interrupt A is generated, the corresponding PIARk.PIRj flag becomes 1 regardless of whether the interrupt source is selected in the SLIARn register (n = 208 to 255).
Although the PIRj flag does not become 0 after the interrupt request is accepted by the destination (CPU, DTC, DMAC), generation of interrupt requests is not affected.
When using the PIRj flag for polling, set the PIRj flag to 0 by writing 1 to the flag in advance.
This flag becomes 1 under the following condition:

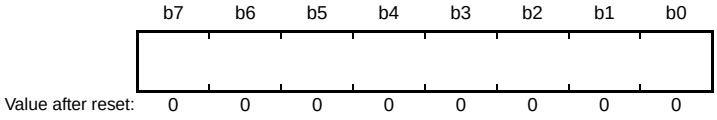
- An interrupt request is generated

This flag becomes 0 under the following condition:

- 1 is written to the PIARk.PIRj flag

15.2.25 Software Configurable Interrupt B Source Select Register Xn (SLIBXRn) (n = 128 to 143)

Address(es): 0008 7780h to 0008 778Fh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	00h: No interrupt source selected. 01h: Interrupt source number 1 : FEh: Interrupt source number 254 FFh: Interrupt source number 255	R/W*1

Note 1. Writing to these bits is ignored while the SLIPRCR.WPRC bit is set to 1.

The SLIBXRn register is used to assign interrupt vector numbers 128 and 143 to interrupt sources assigned to software configurable interrupt B.

Table 15.3, Interrupt Sources for Software Configurable Interrupt B lists interrupt sources assigned to software configurable interrupt B. Set the SLIBXRn register to an interrupt source number that is not reserved. When 00h or FFh are set, no interrupt source is assigned to interrupt vector number n.

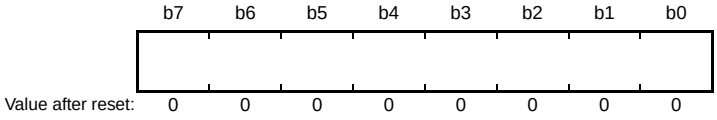
Do not assign the same interrupt source to multiple registers from registers SLIBXRn and SLIBRn.

Some interrupt sources can be used to start the DTC or DMAC. Refer to Table 15.3, Interrupt Sources for Software Configurable Interrupt B for details on which sources can be used to start the DTC or DMAC.

Refer to section 15.7.7, Setting Software Configurable Interrupts for the procedure to set software configurable interrupts.

15.2.26 Software Configurable Interrupt B Source Select Register n (SLIBRn) (n = 144 to 207)

Address(es): 0008 7790h to 0008 77CFh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	00h: No interrupt source selected. 01h: Interrupt source number 1 : FEh: Interrupt source number 254 FFh: No interrupt selected.	R/W*1

Note 1. Writing to these bits is ignored while the SLIPRCR.WPRC bit is set to 1.

The SLIBRn register is used to assign interrupt vector numbers 144 and 207 to interrupt sources for software configurable interrupt B.

Table 15.3, Interrupt Sources for Software Configurable Interrupt B lists interrupt sources for software configurable interrupt B. Set the SLIBRn register to an interrupt source number that is not reserved. When 00h or FFh are set, no interrupt source is assigned to interrupt vector number n.

Do not assign the same interrupt source to multiple registers from registers SLIBXRn and SLIBRn.

Some interrupt sources can be used to start the DTC or DMAC. Refer to Table 15.3, Interrupt Sources for Software Configurable Interrupt B for details on which sources can be used to start the DTC or DMAC.

Although interrupt sources of interrupt vector numbers 144 and 145 can be used to start EXDMAC0 and EXDMAC1, respectively, only interrupt source number 20, TPU1.TGI1A (TGRA input capture/compare match) can be set as a source to start the EXDMAC. Set only one of either the SLIBR144 register or SLIBR145 register to 20, and set the corresponding bit (SELEXD0 or SELEXD1 bit) in the SELEXDR register to 0.

Refer to section 15.7.7, Setting Software Configurable Interrupts for the procedure to set software configurable interrupts.

Table 15.3 Interrupt Sources for Software Configurable Interrupt B (1/2)

Interrupt Source No.	Category	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
0	—	None	No interrupt selected (initial value)	N/A	N/A	PIBR0.PIR0
1	Edge	CMT2	CMI2 (CMCOR compare match)	✓	✓	PIBR0.PIR1
2		CMT3	CMI3 (CMCOR compare match)	✓	✓	PIBR0.PIR2
3		TMR0	CMIA0 (TCORA compare match)	✓	N/A	PIBR0.PIR3
4			CMIB0 (TCORB compare match)	✓	N/A	PIBR0.PIR4
5			OVI0 (TCNT overflow)	N/A	N/A	PIBR0.PIR5
6		TMR1	CMIA1 (TCORA compare match)	✓	N/A	PIBR0.PIR6
7			CMIB1 (TCORB compare match)	✓	N/A	PIBR0.PIR7
8			OVI1 (TCNT overflow)	N/A	N/A	PIBR1.PIR0
9		TMR2	CMIA2 (TCORA compare match)	✓	N/A	PIBR1.PIR1
10			CMIB2 (TCORB compare match)	✓	N/A	PIBR1.PIR2
11			OVI2 (TCNT overflow)	N/A	N/A	PIBR1.PIR3
12		TMR3	CMIA3 (TCORA compare match)	✓	N/A	PIBR1.PIR4
13			CMIB3 (TCORB compare match)	✓	N/A	PIBR1.PIR5
14			OVI3 (TCNT overflow)	N/A	N/A	PIBR1.PIR6
15		TPU0	TGI0A (TGRA input capture/compare match)	✓	✓	PIBR1.PIR7
16			TGI0B (TGRB input capture/compare match)	✓	N/A	PIBR2.PIR0
17			TGI0C (TGRC input capture/compare match)	✓	N/A	PIBR2.PIR1
18			TGI0D (TGRD input capture/compare match)	✓	N/A	PIBR2.PIR2
19			TCI0V (TCNT overflow)	N/A	N/A	PIBR2.PIR3
20 *1		TPU1	TGI1A (TGRA input capture/compare match)	✓	✓	PIBR2.PIR4
21			TGI1B (TGRB input capture/compare match)	✓	N/A	PIBR2.PIR5
22			TCI1V (TCNT overflow)	N/A	N/A	PIBR2.PIR6
23			TCI1U (TCNT underflow)	N/A	N/A	PIBR2.PIR7
24		TPU2	TGI2A (TGRA input capture/compare match)	✓	✓	PIBR3.PIR0
25			TGI2B (TGRB input capture/compare match)	✓	N/A	PIBR3.PIR1
26			TCI2V (TCNT overflow)	N/A	N/A	PIBR3.PIR2
27			TCI2U (TCNT underflow)	N/A	N/A	PIBR3.PIR3
28		TPU3	TGI3A (TGRA input capture/compare match)	✓	✓	PIBR3.PIR4
29			TGI3B (TGRB input capture/compare match)	✓	N/A	PIBR3.PIR5
30			TGI3C (TGRC input capture/compare match)	✓	N/A	PIBR3.PIR6
31			TGI3D (TGRD input capture/compare match)	✓	N/A	PIBR3.PIR7
32			TCI3V (TCNT overflow)	N/A	N/A	PIBR4.PIR0
33		TPU4	TGI4A (TGRA input capture/compare match)	✓	✓	PIBR4.PIR1
34			TGI4B (TGRB input capture/compare match)	✓	N/A	PIBR4.PIR2
35			TCI4V (TCNT overflow)	N/A	N/A	PIBR4.PIR3
36			TCI4U (TCNT underflow)	N/A	N/A	PIBR4.PIR4
37		TPU5	TGI5A (TGRA input capture/compare match)	✓	✓	PIBR4.PIR5
38			TGI5B (TGRB input capture/compare match)	✓	N/A	PIBR4.PIR6
39			TCI5V (TCNT overflow)	N/A	N/A	PIBR4.PIR7
40			TCI5U (TCNT underflow)	N/A	N/A	PIBR5.PIR0

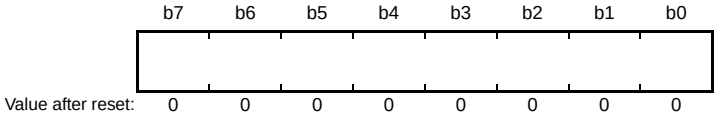
Table 15.3 Interrupt Sources for Software Configurable Interrupt B (2/2)

Interrupt Source No.	Category	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
41	Edge	CMTW0	IC0I0 (input capture of the CMWICR0 register)	✓	✓	PIBR5.PIR1
42			IC1I0 (input capture of the CMWICR1 register)	✓	✓	PIBR5.PIR2
43			OC0I0 (output compare of the CMWOCR0 register)	✓	✓	PIBR5.PIR3
44			OC1I0 (output compare of the CMWOCR1 register)	✓	✓	PIBR5.PIR4
45		CMTW1	IC0I1 (input capture of the CMWICR0 register)	✓	✓	PIBR5.PIR5
46			IC1I1 (input capture of the CMWICR1 register)	✓	✓	PIBR5.PIR6
47			OC0I1 (output compare of the CMWOCR0 register)	✓	✓	PIBR5.PIR7
48			OC1I1 (output compare of the CMWOCR1 register)	✓	✓	PIBR6.PIR0
49		RTC	CUP (carry interrupt)	N/A	N/A	PIBR6.PIR1
50		CAN0	RXF0 (receive FIFO interrupt)	N/A	N/A	PIBR6.PIR2
51			TXF0 (transmit FIFO interrupt)	N/A	N/A	PIBR6.PIR3
52			RXM0 (mailboxes 0 to 31 message reception completed)	N/A	N/A	PIBR6.PIR4
53			TXM0 (mailboxes 0 to 31 message transmission completed)	N/A	N/A	PIBR6.PIR5
54		CAN1	RXF1 (receive FIFO interrupt)	N/A	N/A	PIBR6.PIR6
55			TXF1 (transmit FIFO interrupt)	N/A	N/A	PIBR6.PIR7
56			RXM1 (mailboxes 0 to 31 message reception completed)	N/A	N/A	PIBR7.PIR0
57			TXM1 (mailboxes 0 to 31 message transmission completed)	N/A	N/A	PIBR7.PIR1
58		CAN2	RXF2 (receive FIFO interrupt)	N/A	N/A	PIBR7.PIR2
59			TXF2 (transmit FIFO interrupt)	N/A	N/A	PIBR7.PIR3
60			RXM2 (mailboxes 0 to 31 message reception completed)	N/A	N/A	PIBR7.PIR4
61			TXM2 (mailboxes 0 to 31 message transmission completed)	N/A	N/A	PIBR7.PIR5
62		USB0	USBI0 (status interrupt from 15 sources)	N/A	N/A	PIBR7.PIR6
63		Reserved	—	N/A	N/A	PIBR7.PIR7
64		S12AD	S12ADI (A/D conversion end)	✓	✓	PIBR8.PIR0
65			S12GBADI (group B A/D conversion end interrupt)	✓	✓	PIBR8.PIR1
66		Reserved	—	N/A	N/A	PIBR8.PIR2
67		Reserved	—	N/A	N/A	PIBR8.PIR3
68		S12AD1	S12ADI1 (A/D conversion end)	✓	✓	PIBR8.PIR4
69			S12GBADI1 (group B A/D conversion end interrupt)	✓	✓	PIBR8.PIR5
70		Reserved	—	N/A	N/A	PIBR8.PIR6
71		Reserved	—	N/A	N/A	PIBR8.PIR7
72		Reserved	—	N/A	N/A	PIBR9.PIR0
73		DES	DESEND (encryption/decryption completed)	✓	✓	PIBR9.PIR1
74		SHA	SHADEND (direct encryption/decryption completed)	✓	✓	PIBR9.PIR2
75			SHAEND (encryption/decryption completed)	✓	✓	PIBR9.PIR3
76		RNG	RNGEND (completed interrupt)	✓	✓	PIBR9.PIR4
77		Reserved	—	N/A	N/A	PIBR9.PIR5
78		Reserved	—	N/A	N/A	PIBR9.PIR6
79		ELC	ELSR18I (ELC interrupt)	✓	✓	PIBR9.PIR7
80			ELSR19I (ELC interrupt)	✓	✓	PIBRA.PIR0
81 to 254	—	Reserved	—	N/A	N/A	—
255	—	Reserved/ none *2	—/No interrupt selected	N/A	N/A	—

Note 1. No. 20 can be used as a trigger for EXDMAC0 or EXDMAC1.
Note 2. “Reserved” for the SLIBXRn register, and “none” for the SLIBRn register.

15.2.27 Software Configurable Interrupt A Source Select Register n (SLIARn) (n = 208 to 255)

Address(es): 0008 79D0h to 0008 79FFh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	00h: No interrupt source selected. 01h: Interrupt source number 1 : FEh: Interrupt source number 254 FFh: No interrupt source selected.	R/(W)*1

Note 1. Writing to these bits is ignored while the SLIPRCR.WPRC bit is set to 1.

The SLIARn register is used to assign interrupt vector numbers 208 and 255 to interrupt sources for software configurable interrupt A.

Table 15.4, Interrupt Sources for Software Configurable Interrupt A lists interrupt sources for software configurable interrupt A. Set the SLIARn register to an interrupt source number that is not reserved. When 00h or FFh are set, no interrupt source is assigned to interrupt vector number n.

Do not assign the same interrupt source to multiple SLIARn registers.

Some interrupt sources can be used to start the DTC or DMAC. Refer to Table 15.4, Interrupt Sources for Software Configurable Interrupt A for details on which sources can be used to start the DTC or DMAC.

Although interrupt sources of interrupt vector numbers 208 and 209 can be used to start EXDMAC0 and EXDMAC1, respectively, only interrupt source number 8, MTU1.TGIA1 (TGRA input capture/compare match) can be set as a source to start the EXDMAC. Set only one of either the SLIAR208 register or SLIAR209 register to 8, and set the corresponding bit (SELEXD0 or SELEXD1) in the SELEXDR register to 1.

Refer to section 15.7.7, Setting Software Configurable Interrupts for the procedure to set software configurable interrupts.

Table 15.4 Interrupt Sources for Software Configurable Interrupt A (1/3)

Interrupt Source No.	Category	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
0	—	None	No interrupt selected (initial value)	N/A	N/A	PIAR0.PIR0
1	Edge	MTU0	TGIA0 (TGRA input capture/compare match)	✓	✓	PIAR0.PIR1
2			TGIB0 (TGRB input capture/compare match)	✓	✓	PIAR0.PIR2
3			TGIC0 (TGRC input capture/compare match)	✓	✓	PIAR0.PIR3
4			TGID0 (TGRD input capture/compare match)	✓	✓	PIAR0.PIR4
5			TCIV0 (TCNT overflow)	N/A	N/A	PIAR0.PIR5
6			TGIE0 (TGRE input capture/compare match)	N/A	N/A	PIAR0.PIR6
7			TGIF0 (TGRF input capture/compare match)	N/A	N/A	PIAR0.PIR7
8 *1		MTU1	TGIA1 (TGRA input capture/compare match)	✓	✓	PIAR1.PIR0
9			TGIB1 (TGRB input capture/compare match)	✓	✓	PIAR1.PIR1
10			TCIV1 (TCNT overflow)	N/A	N/A	PIAR1.PIR2
11			TCIU1 (TCNT underflow)	N/A	N/A	PIAR1.PIR3
12		MTU2	TGIA2 (TGRA input capture/compare match)	✓	✓	PIAR1.PIR4
13			TGIB2 (TGRB input capture/compare match)	✓	✓	PIAR1.PIR5
14			TCIV2 (TCNT overflow)	N/A	N/A	PIAR1.PIR6
15			TCIU2 (TCNT underflow)	N/A	N/A	PIAR1.PIR7
16		MTU3	TGIA3 (TGRA input capture/compare match)	✓	✓	PIAR2.PIR0
17			TGIB3 (TGRB input capture/compare match)	✓	✓	PIAR2.PIR1
18			TGIC3 (TGRC input capture/compare match)	✓	✓	PIAR2.PIR2
19			TGID3 (TGRD input capture/compare match)	✓	✓	PIAR2.PIR3
20			TCIV3 (TCNT overflow)	N/A	N/A	PIAR2.PIR4
21		MTU4	TGIA4 (TGRA input capture/compare match)	✓	✓	PIAR2.PIR5
22			TGIB4 (TGRB input capture/compare match)	✓	✓	PIAR2.PIR6
23			TGIC4 (TGRC input capture/compare match)	✓	✓	PIAR2.PIR7
24			TGID4 (TGRD input capture/compare match)	✓	✓	PIAR3.PIR0
25			TCIV4 (TCNT overflow/underflow (only for complementary PWM mode))	✓	✓	PIAR3.PIR1
26		Reserved	—	N/A	N/A	PIAR3.PIR2
27		MTU5	TGIU5 (TGRU input capture/compare match)	✓	✓	PIAR3.PIR3
28			TGIV5 (TGRV input capture/compare match)	✓	✓	PIAR3.PIR4
29			TGIW5 (TGRW input capture/compare match)	✓	✓	PIAR3.PIR5
30		MTU6	TGIA6 (TGRA input capture/compare match)	✓	✓	PIAR3.PIR6
31			TGIB6 (TGRB input capture/compare match)	✓	✓	PIAR3.PIR7
32			TGIC6 (TGRC input capture/compare match)	✓	✓	PIAR4.PIR0
33			TGID6 (TGRD input capture/compare match)	✓	✓	PIAR4.PIR1
34			TCIV6 (TCNT overflow)	N/A	N/A	PIAR4.PIR2
35		MTU7	TGIA7 (TGRA input capture/compare match)	✓	✓	PIAR4.PIR3
36			TGIB7 (TGRB input capture/compare match)	✓	✓	PIAR4.PIR4
37			TGIC7 (TGRC input capture/compare match)	✓	✓	PIAR4.PIR5
38			TGID7 (TGRD input capture/compare match)	✓	✓	PIAR4.PIR6
39			TCIV7 (TCNT overflow/underflow (only for complementary PWM mode))	✓	✓	PIAR4.PIR7
40		Reserved	—	N/A	N/A	PIAR5.PIR0

Table 15.4 Interrupt Sources for Software Configurable Interrupt A (2/3)

Interrupt Source No.	Category	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
41	Edge	MTU8	TGIA8 (TGRA input capture/compare match)	✓	✓	PIAR5.PIR1
42			TGIB8 (TGRB input capture/compare match)	✓	✓	PIAR5.PIR2
43			TGIC8 (TGRC input capture/compare match)	✓	✓	PIAR5.PIR3
44			TGID8 (TGRD input capture/compare match)	✓	✓	PIAR5.PIR4
45			TCIV8 (TCNT overflow)	N/A	N/A	PIAR5.PIR5
46		Reserved	—	N/A	N/A	PIAR5.PIR6
47		GPT0	GTCIA0 (GTCCRA input capture/compare match)	✓	✓	PIAR5.PIR7
48			GTCIB0 (GTCCRB input capture/compare match)	✓	✓	PIAR6.PIR0
49			GTCIC0 (GTCCRC compare match)	✓	✓	PIAR6.PIR1
50			GTCID0 (GTCCRD compare match)	✓	✓	PIAR6.PIR2
51			GDTE0 (dead time error)	✓	✓	PIAR6.PIR3
52			GTCIE0 (GTCCRE compare match)	✓	✓	PIAR6.PIR4
53			GTCIF0 (GTCCRF compare match)	✓	✓	PIAR6.PIR5
54			GTCIV0 (GTCNT overflow (GTPR compare match))	✓	✓	PIAR6.PIR6
55			GTCIU0 (GTCNT underflow)	✓	✓	PIAR6.PIR7
56		Common to GPT	ETGIN (external trigger fall input)	✓	✓	PIAR7.PIR0
57			ETGIP (external trigger rise input)	✓	✓	PIAR7.PIR1
58		GPT1	GTCIA1 (GTCCRA input capture/compare match)	✓	✓	PIAR7.PIR2
59			GTCIB1 (GTCCRB input capture/compare match)	✓	✓	PIAR7.PIR3
60			GTCIC1 (GTCCRC compare match)	✓	✓	PIAR7.PIR4
61			GTCID1 (GTCCRD compare match)	✓	✓	PIAR7.PIR5
62			GDTE1 (dead time error)	✓	✓	PIAR7.PIR6
63			GTCIE1 (GTCCRE compare match)	✓	✓	PIAR7.PIR7
64			GTCIF1 (GTCCRF compare match)	✓	✓	PIAR8.PIR0
65			GTCIV1 (GTCNT overflow (GTPR compare match))	✓	✓	PIAR8.PIR1
66			GTCIU1 (GTCNT underflow)	✓	✓	PIAR8.PIR2
67		GPT2	GTCIA2 (GTCCRA input capture/compare match)	✓	✓	PIAR8.PIR3
68			GTCIB2 (GTCCRB input capture/compare match)	✓	✓	PIAR8.PIR4
69			GTCIC2 (GTCCRC compare match)	✓	✓	PIAR8.PIR5
70			GTCID2 (GTCCRD compare match)	✓	✓	PIAR8.PIR6
71			GDTE2 (dead time error)	✓	✓	PIAR8.PIR7
72			GTCIE2 (GTCCRE compare match)	✓	✓	PIAR9.PIR0
73			GTCIF2 (GTCCRF compare match)	✓	✓	PIAR9.PIR1
74			GTCIV2 (GTCNT overflow (GTPR compare match))	✓	✓	PIAR9.PIR2
75			GTCIU2 (GTCNT underflow)	✓	✓	PIAR9.PIR3
76		GPT3	GTCIA3 (GTCCRA input capture/compare match)	✓	✓	PIAR9.PIR4
77			GTCIB3 (GTCCRB input capture/compare match)	✓	✓	PIAR9.PIR5
78			GTCIC3 (GTCCRC compare match)	✓	✓	PIAR9.PIR6
79			GTCID3 (GTCCRD compare match)	✓	✓	PIAR9.PIR7
80			GDTE3 (dead time error)	✓	✓	PIARA.PIR0
81			GTCIE3 (GTCCRE compare match)	✓	✓	PIARA.PIR1
82			GTCIF3 (GTCCRF compare match)	✓	✓	PIARA.PIR2
83			GTCIV3 (GTCNT overflow (GTPR compare match))	✓	✓	PIARA.PIR3
84			GTCIU3 (GTCNT underflow)	✓	✓	PIARA.PIR4

Table 15.4 Interrupt Sources for Software Configurable Interrupt A (3/3)

Interrupt Source No.	Category	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
85	Edge	Reserved	—	N/A	N/A	—
86		EPTPC	IPLS (timer interrupt)	✓	✓	PIARA.PIR6
87		Reserved	—	N/A	N/A	—
88		AES	AESRDY (encryption/decryption ready)	✓	✓	PIARB.PIR0
89			AESEND (encryption/decryption completed)	✓	✓	PIARB.PIR1
90 to 126	—	Reserved	—	N/A	N/A	—
127	—	None	No interrupt selected	N/A	N/A	—

Note 1. No. 8 can be used as a trigger for EXDMAC0 or EXDMAC1.

15.2.28 EXDMAC Trigger Select Register (SELEXDR)

Address(es): 0008 7A01h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SELEXD1	SELEXD0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SELEXD0	EXDMAC0 Trigger Select	0: Interrupt source selected in the SLIBR144 register is used as the EXDMAC0 trigger. 1: Interrupt source selected in the SLIAR208 register is used as the EXDMAC0 trigger.	R/W*1
b1	SELEXD1	EXDMAC1 Trigger Select	0: Interrupt source selected in the SLIBR145 register is used as the EXDMAC1 trigger. 1: Interrupt source selected in the SLIAR209 register is used as the EXDMAC1 trigger.	R/W*1
b7 to b2	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note 1. Writing to this bit is ignored while the SLIPRCR.WPRC bit is 1.

The SELEXDR register selects an interrupt source for software configurable interrupts as the EXDMAC trigger. Refer to section 15.4.5.3, EXDMAC Start Trigger by Software Configurable Interrupts for details on starting the EXDMAC by an interrupt request.

SELEXD0 Bit (EXDMAC0 Trigger Select)

The SELEXD0 bit selects an interrupt source set in the SLIBR144 register or SLIAR208 register as the EXDMAC0 trigger.

SELEXD1 Bit (EXDMAC1 Trigger Select)

The SELEXD1 bit selects an interrupt source set in the SLIBR145 register or SLIAR209 register as the EXDMAC1 trigger.

15.2.29 Software Configurable Interrupt Source Select Register Write Protect Register (SLIPRCR)

Address(es): 0008 7A00h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	WPRC
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WPRC	Software Configurable Interrupt Source Select Register Write Protect	0: Write enabled. 1: Write disabled.	R/(W) *1
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R

Note 1. Once this bit is set to 1, it cannot be set to 0 by software.

The SLIPRCR register protects registers that control assignment of software configurable interrupts from being written to.

WPRC Bit (Software Configurable Interrupt Source Select Register Write Protect)

The WPRC bit disables writing to registers SLIBXRn, SLIBRn, SLIARn, and SELEXDR.

Once this bit is set to 1, it cannot be set to 0 by software.

After assigning software configurable interrupts, confirm that the WPRC bit is 1 before the corresponding interrupt request is generated. Refer to section 15.7.7, Setting Software Configurable Interrupts for the procedure to set software configurable interrupts.

15.3 Vector Table

There are two types of exceptions detected by the ICU: maskable interrupts (hereinafter referred to as “interrupts”) and non-maskable interrupts.
When the CPU accepts an interrupt or non-maskable interrupt, it acquires a 4-byte vector address from the vector table.

15.3.1 Interrupt Vector Table

The vector table that is used for maskable interrupts is called the interrupt vector table.
The interrupt vector table is allocated to a 1024-byte area (4 bytes × 256 sources) beginning with the address set in the INTB register in the CPU. Set the INTB register before enabling interrupts. Set a multiple of 4 in the INTB register.
An unconditional trap is generated when the INT or BRK instruction is executed. Interrupt vectors for unconditional traps use the same area as the interrupt vector table. The BRK instruction is assigned to interrupt vector number 0. The INT instruction is assigned to the interrupt vector number corresponding to the value set as the operand (0 to 255).
Table 15.5 lists details of the interrupt vectors. Details of the headings in Table 15.5 are listed below.

Heading	Description
Interrupt request generated by	Name of the source that generates the interrupt request (module symbol)
Name	Name of the interrupt source (symbol)
Vector no.	Interrupt vector number
Vector address offset	Offset from the address set in the INTB register
Interrupt detection method	“Edge” indicates that the interrupt is detected by edge detection. “Level” indicates that the interrupt is detected by level detection.
CPU interrupt	Interrupt source indicated by “✓” can be used as an interrupt source to the CPU.
Start the DTC	Interrupt source indicated by “✓” can be used as the DTC trigger.
Start the DMAC	Interrupt source indicated by “✓” can be used as the DMAC trigger.
Exit from SSBY	Interrupt source indicated by “✓” can be used as a source to exit software standby mode.
Exit from ACS	Interrupt source indicated by “✓” can be used as a source to exit all-module clock stop mode.
IER	Name of the bit in the IER register corresponding to the interrupt vector number
IPR	Name of the IPR register corresponding to the interrupt source
DTCER	Name of the DTCER register corresponding to the DTC trigger

Table 15.5 Interrupt Vector Table (1/6)

Interrupt Request is Generated By	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	Exit from ACS	IER	IPR	DTCER
—	For an unconditional trap	0	0000h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	1	0004h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	2	0008h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	3	000Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	4	0010h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	5	0014h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	6	0018h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	7	001Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	8	0020h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	9	0024h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	10	0028h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	11	002Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	12	0030h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	13	0034h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	14	0038h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	15	003Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
BSC	BUSERR	16	0040h	Level	✓	N/A	N/A	N/A	N/A	IER02.IEN0	IPR000	—
—	Reserved	17	0044h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
RAM	RAMERR*1	18	0048h	Level	✓	N/A	N/A	N/A	N/A	IER02.IEN2	IPR000	—
—	Reserved	19	004Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	20	0050h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
FCU	FIFERR	21	0054h	Level	✓	N/A	N/A	N/A	N/A	IER02.IEN5	IPR001	—
—	Reserved	22	0058h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
FCU	FRDYI	23	005Ch	Edge	✓	N/A	N/A	N/A	N/A	IER02.IEN7	IPR002	—
—	Reserved	24	0060h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	25	0064h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
ICU	SWINT2	26	0068h	Edge	✓	✓	N/A	N/A	N/A	IER03.IEN2	IPR003	DTCER026
	SWINT	27	006Ch	Edge	✓	✓	N/A	N/A	N/A	IER03.IEN3		DTCER027
CMT0	CMI0 (for OS)	28	0070h	Edge	✓	✓	✓	N/A	N/A	IER03.IEN4	IPR004	DTCER028
CMT1	CMI1	29	0074h	Edge	✓	✓	✓	N/A	N/A	IER03.IEN5	IPR005	DTCER029
CMTW0	CMWI0	30	0078h	Edge	✓	✓	✓	N/A	N/A	IER03.IEN6	IPR006	DTCER030
CMTW1	CMWI1	31	007Ch	Edge	✓	✓	✓	N/A	N/A	IER03.IEN7	IPR007	DTCER031
USBA	D0FIFO2	32	0080h	Edge	✓	✓	✓	N/A	N/A	IER04.IEN0	IPR032	DTCER032
	D1FIFO2	33	0084h	Edge	✓	✓	✓	N/A	N/A	IER04.IEN1	IPR033	DTCER033
USB0	D0FIFO0	34	0088h	Edge	✓	✓	✓	N/A	N/A	IER04.IEN2	IPR034	DTCER034
	D1FIFO0	35	008Ch	Edge	✓	✓	✓	N/A	N/A	IER04.IEN3	IPR035	DTCER035
—	Reserved	36	0090h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	37	0094h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
RSPI0	SPRI0	38	0098h	Edge	✓	✓	✓	N/A	N/A	IER04.IEN6	IPR038	DTCER038
	SPTI0	39	009Ch	Edge	✓	✓	✓	N/A	N/A	IER04.IEN7	IPR039	DTCER039
—	Reserved	40	00A0h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	41	00A4h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
QSPI	SPRI	42	00A8h	Edge	✓	✓	✓	N/A	N/A	IER05.IEN2	IPR042	DTCER042
	SPTI	43	00ACh	Edge	✓	✓	✓	N/A	N/A	IER05.IEN3	IPR043	DTCER043

Table 15.5 Interrupt Vector Table (2/6)

Interrupt Request is Generated By	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	Exit from ACS	IER	IPR	DTCER
SDHI	SBFAI	44	00B0h	Edge	✓	✓	✓	N/A	N/A	IER05.IEN4	IPR044	DTCER044
MMCIF	MBFAI	45	00B4h	Edge	✓	✓	✓	N/A	N/A	IER05.IEN5	IPR045	DTCER045
SSI0	SSITXI0	46	00B8h	Edge	✓	✓	✓	N/A	N/A	IER05.IEN6	IPR046	DTCER046
	SSIRXI0	47	00BCh	Edge	✓	✓	✓	N/A	N/A	IER05.IEN7	IPR047	DTCER047
SSI1	SSIRTI1	48	00C0h	Edge	✓	✓	✓	N/A	N/A	IER06.IEN0	IPR048	DTCER048
—	Reserved	49	00C4h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
SRC	IDEI	50	00C8h	Edge	✓	✓	✓	N/A	N/A	IER06.IEN2	IPR050	DTCER050
	ODFI	51	00CCh	Edge	✓	✓	✓	N/A	N/A	IER06.IEN3	IPR051	DTCER051
RIIC0	RXI0	52	00D0h	Edge	✓	✓	✓	N/A	N/A	IER06.IEN4	IPR052	DTCER052
	TXI0	53	00D4h	Edge	✓	✓	✓	N/A	N/A	IER06.IEN5	IPR053	DTCER053
RIIC2	RXI2	54	00D8h	Edge	✓	✓	✓	N/A	N/A	IER06.IEN6	IPR054	DTCER054
	TXI2	55	00DCh	Edge	✓	✓	✓	N/A	N/A	IER06.IEN7	IPR055	DTCER055
—	Reserved	56	00E0h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	57	00E4h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
SCI0	RXI0	58	00E8h	Edge	✓	✓	✓	N/A	N/A	IER07.IEN2	IPR058	DTCER058
	TXI0	59	00ECh	Edge	✓	✓	✓	N/A	N/A	IER07.IEN3	IPR059	DTCER059
SCI1	RXI1	60	00F0h	Edge	✓	✓	✓	N/A	N/A	IER07.IEN4	IPR060	DTCER060
	TXI1	61	00F4h	Edge	✓	✓	✓	N/A	N/A	IER07.IEN5	IPR061	DTCER061
SCI2	RXI2	62	00F8h	Edge	✓	✓	✓	N/A	N/A	IER07.IEN6	IPR062	DTCER062
	TXI2	63	00FCh	Edge	✓	✓	✓	N/A	N/A	IER07.IEN7	IPR063	DTCER063
ICU	IRQ0	64	0100h	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN0	IPR064	DTCER064
	IRQ1	65	0104h	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN1	IPR065	DTCER065
	IRQ2	66	0108h	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN2	IPR066	DTCER066
	IRQ3	67	010Ch	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN3	IPR067	DTCER067
	IRQ4	68	0110h	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN4	IPR068	DTCER068
	IRQ5	69	0114h	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN5	IPR069	DTCER069
	IRQ6	70	0118h	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN6	IPR070	DTCER070
	IRQ7	71	011Ch	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN7	IPR071	DTCER071
	IRQ8	72	0120h	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN0	IPR072	DTCER072
	IRQ9	73	0124h	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN1	IPR073	DTCER073
	IRQ10	74	0128h	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN2	IPR074	DTCER074
	IRQ11	75	012Ch	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN3	IPR075	DTCER075
	IRQ12	76	0130h	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN4	IPR076	DTCER076
	IRQ13	77	0134h	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN5	IPR077	DTCER077
	IRQ14	78	0138h	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN6	IPR078	DTCER078
	IRQ15	79	013Ch	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN7	IPR079	DTCER079

Table 15.5 Interrupt Vector Table (3/6)

Interrupt Request is Generated By	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	Exit from ACS	IER	IPR	DTCER
SCI3	RXI3	80	0140h	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN0	IPR080	DTCER080
	TXI3	81	0144h	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN1	IPR081	DTCER081
SCI4	RXI4	82	0148h	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN2	IPR082	DTCER082
	TXI4	83	014Ch	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN3	IPR083	DTCER083
SCI5	RXI5	84	0150h	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN4	IPR084	DTCER084
	TXI5	85	0154h	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN5	IPR085	DTCER085
SCI6	RXI6	86	0158h	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN6	IPR086	DTCER086
	TXI6	87	015Ch	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN7	IPR087	DTCER087
LVD1	LVD1	88	0160h	Edge	✓	N/A	N/A	✓	✓	IER0B.IEN0	IPR088	—
LVD2	LVD2	89	0164h	Edge	✓	N/A	N/A	✓	✓	IER0B.IEN1	IPR089	—
USB0	USBR0	90	0168h	Level	✓	N/A	N/A	✓	✓	IER0B.IEN2	IPR090	—
—	Reserved	91	016Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
RTC	ALM	92	0170h	Edge	✓	N/A	N/A	✓	✓	IER0B.IEN4	IPR092	—
	PRD	93	0174h	Edge	✓	N/A	N/A	✓	✓	IER0B.IEN5	IPR093	—
USBA	USBAR	94	0178h	Level	✓	N/A	N/A	✓	✓	IER0B.IEN6	IPR094	—
IWDT	IWUNI*1	95	017Ch	Edge	✓	N/A	N/A	✓	✓	IER0B.IEN7	IPR095	—
WDT	WUNI*1	96	0180h	Edge	✓	N/A	N/A	N/A	N/A	IER0C.IEN0	IPR096	—
PDC	PCDFI	97	0184h	Edge	✓	✓	✓	N/A	N/A	IER0C.IEN1	IPR097	DTCER097
SCI7	RXI7	98	0188h	Edge	✓	✓	✓	N/A	N/A	IER0C.IEN2	IPR098	DTCER098
	TXI7	99	018Ch	Edge	✓	✓	✓	N/A	N/A	IER0C.IEN3	IPR099	DTCER099
SCIFA8	RXIF8	100	0190h	Edge	✓	✓	✓	N/A	N/A	IER0C.IEN4	IPR100	DTCER100
	TXIF8	101	0194h	Edge	✓	✓	✓	N/A	N/A	IER0C.IEN5	IPR101	DTCER101
SCIFA9	RXIF9	102	0198h	Edge	✓	✓	✓	N/A	N/A	IER0C.IEN6	IPR102	DTCER102
	TXIF9	103	019Ch	Edge	✓	✓	✓	N/A	N/A	IER0C.IEN7	IPR103	DTCER103
SCIFA10	RXIF10	104	01A0h	Edge	✓	✓	✓	N/A	N/A	IER0D.IEN0	IPR104	DTCER104
	TXIF10	105	01A4h	Edge	✓	✓	✓	N/A	N/A	IER0D.IEN1	IPR105	DTCER105
ICU*2	GROUPBE0	106	01A8h	Level	✓	N/A	N/A	N/A	N/A	IER0D.IEN2	IPR106	—
	Reserved	107	01ACh	—	N/A	N/A	N/A	N/A	N/A	—	—	—
	Reserved	108	01B0h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
	Reserved	109	01B4h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
	GROUPBL0	110	01B8h	Level	✓	N/A	N/A	N/A	N/A	IER0D.IEN6	IPR110	—
	GROUPBL1	111	01BCh	Level	✓	N/A	N/A	N/A	N/A	IER0D.IEN7	IPR111	—
	GROUPAL0	112	01C0h	Level	✓	N/A	N/A	N/A	N/A	IER0E.IEN0	IPR112	—
	GROUPAL1	113	01C4h	Level	✓	N/A	N/A	N/A	N/A	IER0E.IEN1	IPR113	—
SCIFA11	RXIF11	114	01C8h	Edge	✓	✓	✓	N/A	N/A	IER0E.IEN2	IPR114	DTCER114
	TXIF11	115	01CCh	Edge	✓	✓	✓	N/A	N/A	IER0E.IEN3	IPR115	DTCER115
SCI12	RXI12	116	01D0h	Edge	✓	✓	✓	N/A	N/A	IER0E.IEN4	IPR116	DTCER116
	TXI12	117	01D4h	Edge	✓	✓	✓	N/A	N/A	IER0E.IEN5	IPR117	DTCER117
—	Reserved	118	01D8h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	119	01DCh	—	N/A	N/A	N/A	N/A	N/A	—	—	—
DMAC	DMAC0I	120	01E0h	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN0	IPR120	DTCER120
	DMAC1I	121	01E4h	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN1	IPR121	DTCER121
	DMAC2I	122	01E8h	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN2	IPR122	DTCER122
	DMAC3I	123	01ECh	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN3	IPR123	DTCER123
	DMAC74I	124	01F0h	Level	✓	N/A	N/A	N/A	N/A	IER0F.IEN4	IPR124	—

Table 15.5 Interrupt Vector Table (4/6)

Interrupt Request is Generated By	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	Exit from ACS	IER	IPR	DTCER
OST	OSTDI*1	125	01F4h	Edge	✓	N/A	N/A	N/A	N/A	IER0F.IEN5	IPR125	—
EXDMAC	EXDMAC0I	126	01F8h	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN6	IPR126	DTCER126
	EXDMAC1I	127	01FCh	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN7	IPR127	DTCER127
PERIB (software configurable interrupt B *3)	INTB128	128	0200h	Edge	✓	✓	✓	N/A	N/A	IER10.IEN0	IPR128	DTCER128
	INTB129	129	0204h	Edge	✓	✓	✓	N/A	N/A	IER10.IEN1	IPR129	DTCER129
	INTB130	130	0208h	Edge	✓	✓	✓	N/A	N/A	IER10.IEN2	IPR130	DTCER130
	INTB131	131	020Ch	Edge	✓	✓	✓	N/A	N/A	IER10.IEN3	IPR131	DTCER131
	INTB132	132	0210h	Edge	✓	✓	✓	N/A	N/A	IER10.IEN4	IPR132	DTCER132
	INTB133	133	0214h	Edge	✓	✓	✓	N/A	N/A	IER10.IEN5	IPR133	DTCER133
	INTB134	134	0218h	Edge	✓	✓	✓	N/A	N/A	IER10.IEN6	IPR134	DTCER134
	INTB135	135	021Ch	Edge	✓	✓	✓	N/A	N/A	IER10.IEN7	IPR135	DTCER135
	INTB136	136	0220h	Edge	✓	✓	✓	N/A	N/A	IER11.IEN0	IPR136	DTCER136
	INTB137	137	0224h	Edge	✓	✓	✓	N/A	N/A	IER11.IEN1	IPR137	DTCER137
	INTB138	138	0228h	Edge	✓	✓	✓	N/A	N/A	IER11.IEN2	IPR138	DTCER138
	INTB139	139	022Ch	Edge	✓	✓	✓	N/A	N/A	IER11.IEN3	IPR139	DTCER139
	INTB140	140	0230h	Edge	✓	✓	✓	N/A	N/A	IER11.IEN4	IPR140	DTCER140
	INTB141	141	0234h	Edge	✓	✓	✓	N/A	N/A	IER11.IEN5	IPR141	DTCER141
	INTB142	142	0238h	Edge	✓	✓	✓	N/A	N/A	IER11.IEN6	IPR142	DTCER142
	INTB143	143	023Ch	Edge	✓	✓	✓	N/A	N/A	IER11.IEN7	IPR143	DTCER143
	INTB144	144*5	0240h	Edge	✓	✓	✓	N/A	N/A	IER12.IEN0	IPR144	DTCER144
	INTB145	145*6	0244h	Edge	✓	✓	✓	N/A	N/A	IER12.IEN1	IPR145	DTCER145
	INTB146	146	0248h	Edge	✓	✓	✓	N/A	✓	IER12.IEN2	IPR146	DTCER146
	INTB147	147	024Ch	Edge	✓	✓	✓	N/A	✓	IER12.IEN3	IPR147	DTCER147
	INTB148	148	0250h	Edge	✓	✓	✓	N/A	✓	IER12.IEN4	IPR148	DTCER148
	INTB149	149	0254h	Edge	✓	✓	✓	N/A	✓	IER12.IEN5	IPR149	DTCER149
	INTB150	150	0258h	Edge	✓	✓	✓	N/A	✓	IER12.IEN6	IPR150	DTCER150
	INTB151	151	025Ch	Edge	✓	✓	✓	N/A	✓	IER12.IEN7	IPR151	DTCER151
	INTB152	152	0260h	Edge	✓	✓	✓	N/A	✓	IER13.IEN0	IPR152	DTCER152
	INTB153	153	0264h	Edge	✓	✓	✓	N/A	✓	IER13.IEN1	IPR153	DTCER153
	INTB154	154	0268h	Edge	✓	✓	✓	N/A	✓	IER13.IEN2	IPR154	DTCER154
	INTB155	155	026Ch	Edge	✓	✓	✓	N/A	✓	IER13.IEN3	IPR155	DTCER155
	INTB156	156	0270h	Edge	✓	✓	✓	N/A	✓	IER13.IEN4	IPR156	DTCER156
	INTB157	157	0274h	Edge	✓	✓	✓	N/A	✓	IER13.IEN5	IPR157	DTCER157
	INTB158	158	0278h	Edge	✓	✓	✓	N/A	N/A	IER13.IEN6	IPR158	DTCER158
	INTB159	159	027Ch	Edge	✓	✓	✓	N/A	N/A	IER13.IEN7	IPR159	DTCER159
	INTB160	160	0280h	Edge	✓	✓	✓	N/A	N/A	IER14.IEN0	IPR160	DTCER160
	INTB161	161	0284h	Edge	✓	✓	✓	N/A	N/A	IER14.IEN1	IPR161	DTCER161
	INTB162	162	0288h	Edge	✓	✓	✓	N/A	N/A	IER14.IEN2	IPR162	DTCER162
	INTB163	163	028Ch	Edge	✓	✓	✓	N/A	N/A	IER14.IEN3	IPR163	DTCER163
	INTB164	164	0290h	Edge	✓	✓	✓	N/A	N/A	IER14.IEN4	IPR164	DTCER164
	INTB165	165	0294h	Edge	✓	✓	✓	N/A	N/A	IER14.IEN5	IPR165	DTCER165
	INTB166	166	0298h	Edge	✓	✓	✓	N/A	N/A	IER14.IEN6	IPR166	DTCER166
	INTB167	167	029Ch	Edge	✓	✓	✓	N/A	N/A	IER14.IEN7	IPR167	DTCER167
	INTB168	168	02A0h	Edge	✓	✓	✓	N/A	N/A	IER15.IEN0	IPR168	DTCER168
	INTB169	169	02A4h	Edge	✓	✓	✓	N/A	N/A	IER15.IEN1	IPR169	DTCER169

Table 15.5 Interrupt Vector Table (5/6)

Interrupt Request is Generated By	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	Exit from ACS	IER	IPR	DTCER
PERIB (software configurable interrupt B *3)	INTB170	170	02A8h	Edge	✓	✓	✓	N/A	N/A	IER15.IEN2	IPR170	DTCER170
	INTB171	171	02ACh	Edge	✓	✓	✓	N/A	N/A	IER15.IEN3	IPR171	DTCER171
	INTB172	172	02B0h	Edge	✓	✓	✓	N/A	N/A	IER15.IEN4	IPR172	DTCER172
	INTB173	173	02B4h	Edge	✓	✓	✓	N/A	N/A	IER15.IEN5	IPR173	DTCER173
	INTB174	174	02B8h	Edge	✓	✓	✓	N/A	N/A	IER15.IEN6	IPR174	DTCER174
	INTB175	175	02BCh	Edge	✓	✓	✓	N/A	N/A	IER15.IEN7	IPR175	DTCER175
	INTB176	176	02C0h	Edge	✓	✓	✓	N/A	N/A	IER16.IEN0	IPR176	DTCER176
	INTB177	177	02C4h	Edge	✓	✓	✓	N/A	N/A	IER16.IEN1	IPR177	DTCER177
	INTB178	178	02C8h	Edge	✓	✓	✓	N/A	N/A	IER16.IEN2	IPR178	DTCER178
	INTB179	179	02CCh	Edge	✓	✓	✓	N/A	N/A	IER16.IEN3	IPR179	DTCER179
	INTB180	180	02D0h	Edge	✓	✓	✓	N/A	N/A	IER16.IEN4	IPR180	DTCER180
	INTB181	181	02D4h	Edge	✓	✓	✓	N/A	N/A	IER16.IEN5	IPR181	DTCER181
	INTB182	182	02D8h	Edge	✓	✓	✓	N/A	N/A	IER16.IEN6	IPR182	DTCER182
	INTB183	183	02DCh	Edge	✓	✓	✓	N/A	N/A	IER16.IEN7	IPR183	DTCER183
	INTB184	184	02E0h	Edge	✓	✓	✓	N/A	N/A	IER17.IEN0	IPR184	DTCER184
	INTB185	185	02E4h	Edge	✓	✓	✓	N/A	N/A	IER17.IEN1	IPR185	DTCER185
	INTB186	186	02E8h	Edge	✓	✓	✓	N/A	N/A	IER17.IEN2	IPR186	DTCER186
	INTB187	187	02ECh	Edge	✓	✓	✓	N/A	N/A	IER17.IEN3	IPR187	DTCER187
	INTB188	188	02F0h	Edge	✓	✓	✓	N/A	N/A	IER17.IEN4	IPR188	DTCER188
	INTB189	189	02F4h	Edge	✓	✓	✓	N/A	N/A	IER17.IEN5	IPR189	DTCER189
	INTB190	190	02F8h	Edge	✓	✓	✓	N/A	N/A	IER17.IEN6	IPR190	DTCER190
	INTB191	191	02FCh	Edge	✓	✓	✓	N/A	N/A	IER17.IEN7	IPR191	DTCER191
	INTB192	192	0300h	Edge	✓	✓	✓	N/A	N/A	IER18.IEN0	IPR192	DTCER192
	INTB193	193	0304h	Edge	✓	✓	✓	N/A	N/A	IER18.IEN1	IPR193	DTCER193
	INTB194	194	0308h	Edge	✓	✓	✓	N/A	N/A	IER18.IEN2	IPR194	DTCER194
	INTB195	195	030Ch	Edge	✓	✓	✓	N/A	N/A	IER18.IEN3	IPR195	DTCER195
	INTB196	196	0310h	Edge	✓	✓	✓	N/A	N/A	IER18.IEN4	IPR196	DTCER196
	INTB197	197	0314h	Edge	✓	✓	✓	N/A	N/A	IER18.IEN5	IPR197	DTCER197
	INTB198	198	0318h	Edge	✓	✓	✓	N/A	N/A	IER18.IEN6	IPR198	DTCER198
	INTB199	199	031Ch	Edge	✓	✓	✓	N/A	N/A	IER18.IEN7	IPR199	DTCER199
	INTB200	200	0320h	Edge	✓	✓	✓	N/A	N/A	IER19.IEN0	IPR200	DTCER200
	INTB201	201	0324h	Edge	✓	✓	✓	N/A	N/A	IER19.IEN1	IPR201	DTCER201
	INTB202	202	0328h	Edge	✓	✓	✓	N/A	N/A	IER19.IEN2	IPR202	DTCER202
	INTB203	203	032Ch	Edge	✓	✓	✓	N/A	N/A	IER19.IEN3	IPR203	DTCER203
	INTB204	204	0330h	Edge	✓	✓	✓	N/A	N/A	IER19.IEN4	IPR204	DTCER204
	INTB205	205	0334h	Edge	✓	✓	✓	N/A	N/A	IER19.IEN5	IPR205	DTCER205
	INTB206	206	0338h	Edge	✓	✓	✓	N/A	N/A	IER19.IEN6	IPR206	DTCER206
	INTB207	207	033Ch	Edge	✓	✓	✓	N/A	N/A	IER19.IEN7	IPR207	DTCER207
PERIA (software configurable interrupt A *4)	INTA208	208*5	0340h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN0	IPR208	DTCER208
	INTA209	209*6	0344h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN1	IPR209	DTCER209
	INTA210	210	0348h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN2	IPR210	DTCER210
	INTA211	211	034Ch	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN3	IPR211	DTCER211
	INTA212	212	0350h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN4	IPR212	DTCER212
	INTA213	213	0354h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN5	IPR213	DTCER213
	INTA214	214	0358h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN6	IPR214	DTCER214

Table 15.5 Interrupt Vector Table (6/6)

Interrupt Request is Generated By	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	Exit from ACS	IER	IPR	DTCER
PERIA (software configurable interrupt A *4)	INTA215	215	035Ch	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN7	IPR215	DTCER215
	INTA216	216	0360h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN0	IPR216	DTCER216
	INTA217	217	0364h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN1	IPR217	DTCER217
	INTA218	218	0368h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN2	IPR218	DTCER218
	INTA219	219	036Ch	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN3	IPR219	DTCER219
	INTA220	220	0370h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN4	IPR220	DTCER220
	INTA221	221	0374h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN5	IPR221	DTCER221
	INTA222	222	0378h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN6	IPR222	DTCER222
	INTA223	223	037Ch	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN7	IPR223	DTCER223
	INTA224	224	0380h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN0	IPR224	DTCER224
	INTA225	225	0384h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN1	IPR225	DTCER225
	INTA226	226	0388h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN2	IPR226	DTCER226
	INTA227	227	038Ch	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN3	IPR227	DTCER227
	INTA228	228	0390h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN4	IPR228	DTCER228
	INTA229	229	0394h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN5	IPR229	DTCER229
	INTA230	230	0398h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN6	IPR230	DTCER230
	INTA231	231	039Ch	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN7	IPR231	DTCER231
	INTA232	232	03A0h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN0	IPR232	DTCER232
	INTA233	233	03A4h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN1	IPR233	DTCER233
	INTA234	234	03A8h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN2	IPR234	DTCER234
	INTA235	235	03ACh	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN3	IPR235	DTCER235
	INTA236	236	03B0h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN4	IPR236	DTCER236
	INTA237	237	03B4h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN5	IPR237	DTCER237
	INTA238	238	03B8h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN6	IPR238	DTCER238
	INTA239	239	03BCh	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN7	IPR239	DTCER239
	INTA240	240	03C0h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN0	IPR240	DTCER240
	INTA241	241	03C4h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN1	IPR241	DTCER241
	INTA242	242	03C8h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN2	IPR242	DTCER242
	INTA243	243	03CCh	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN3	IPR243	DTCER243
	INTA244	244	03D0h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN4	IPR244	DTCER244
	INTA245	245	03D4h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN5	IPR245	DTCER245
	INTA246	246	03D8h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN6	IPR246	DTCER246
	INTA247	247	03DCh	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN7	IPR247	DTCER247
	INTA248	248	03E0h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN0	IPR248	DTCER248
	INTA249	249	03E4h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN1	IPR249	DTCER249
	INTA250	250	03E8h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN2	IPR250	DTCER250
	INTA251	251	03ECh	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN3	IPR251	DTCER251
	INTA252	252	03F0h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN4	IPR252	DTCER252
	INTA253	253	03F4h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN5	IPR253	DTCER253
	INTA254	254	03F8h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN6	IPR254	DTCER254
	INTA255	255	03FCh	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN7	IPR255	DTCER255

Note: This table lists the interrupt vectors for the maximum specification. The interrupt vectors for individual products correspond to the functions listed in Table 1.2. For details, refer to Table 1.2, Comparison of Functions for Different Packages.

Note 1. This is the case where the corresponding non-maskable interrupt enable bit is set to 0 (disabled).

Note 2. For the group interrupt sources, refer to Table 15.7, Group Interrupt Requests.

Note 3. For the software configurable interrupt B sources, refer to Table 15.3, Interrupt Sources for Software Configurable Interrupt B.

Note that some interrupt sources cannot start the DTC, DMAC, or EXDMAC.

Note 4. For the software configurable interrupt A sources, refer to Table 15.4, Interrupt Sources for Software Configurable Interrupt A. Note that some interrupt sources cannot start the DTC, DMAC, or EXDMAC.

Note 5. This can be used to start EXDMAC0. Set the SELEXDR.SELEXD0 bit to 0 when using the interrupt source assigned to interrupt vector number 144 as a trigger. Set the SELEXDR.SELEXD0 bit to 1 when using the interrupt source assigned to interrupt vector number 208 as a trigger. Refer to section 15.2.26, Software Configurable Interrupt B Source Select Register n (SLIBRn) (n = 144 to 207), section 15.2.27, Software Configurable Interrupt A Source Select Register n (SLIARn) (n = 208 to 255), and section 15.2.28, EXDMAC Trigger Select Register (SELEXDR).

Note 6. This can be used to start EXDMAC1. Set the SELEXDR.SELEXD1 bit to 0 when using the interrupt source assigned to interrupt vector number 145 as a trigger. Set the SELEXDR.SELEXD1 bit to 1 when using the interrupt source assigned to interrupt vector number 209 as a trigger.

15.3.2 Fast Interrupt Vector Area

The interrupt set as the fast interrupt uses the FINTV register in the CPU. Set the FINTV register before enabling the fast interrupt.

15.3.3 Non-maskable Interrupt Vector Area

Non-maskable interrupts use the vector area in the exception vector table.

The exception vector table is allocated to the 128-byte area (4 bytes × 32 sources) beginning with the address set in the EXTB register in the CPU. Set a multiple of 4 in the EXTB register.

15.4 Types of Interrupts

Interrupts are divided into maskable interrupts and non-maskable interrupts. Maskable interrupts can be masked by the PSW.I bit or IPL[3:0] bits of the processor status word in the CPU. Non-maskable interrupts can be accepted by the CPU regardless of those bits. While interrupt sources assigned to vector numbers 0 to 127 are fixed, an interrupt source assigned to each vector number from 128 to 255 (software configurable interrupt) can be selected from multiple sources. Note that maskable interrupts are referred to as interrupts in this chapter.
Figure 15.2 shows types of interrupts.

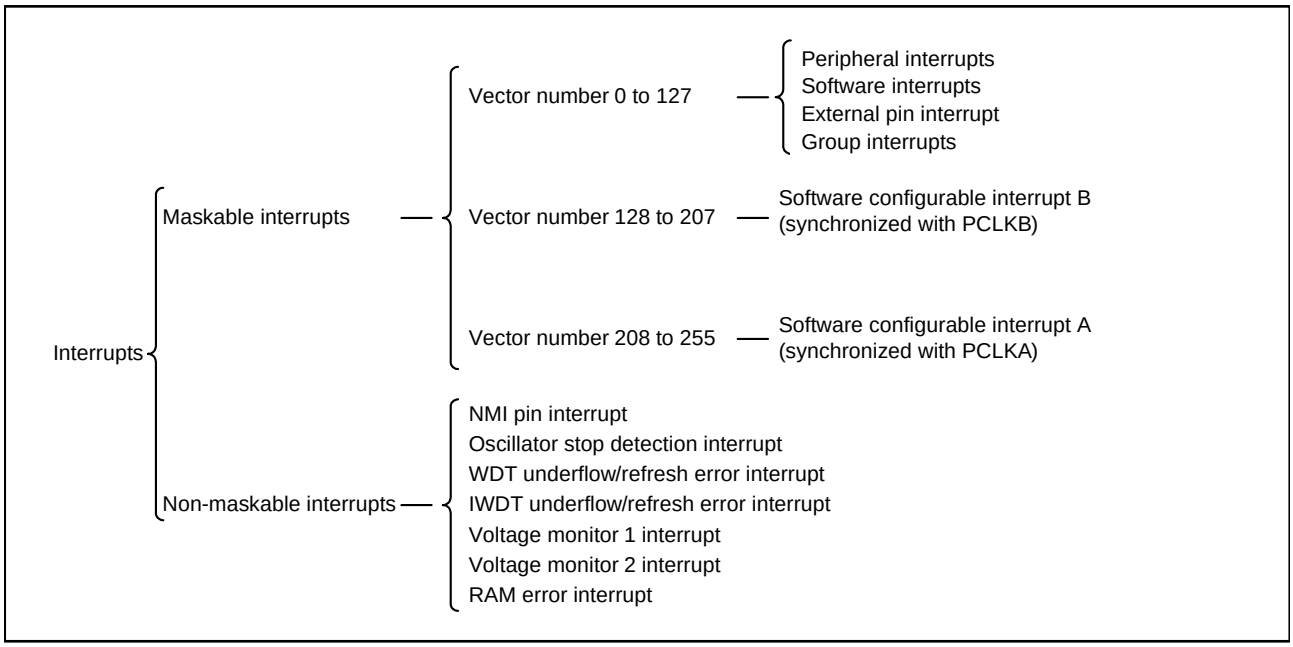


Figure 15.2 Types of Interrupts

15.4.1 Peripheral Interrupts

Peripheral interrupt is generated by peripherals. Peripheral interrupts sources assigned to vector numbers 0 to 127 cannot be assigned to the software configurable interrupts. Refer to section 15.4.5, Software Configurable Interrupts for details on software configurable interrupts.

15.4.2 Software Interrupts

When the SWINTR.SWINT bit and SWINT2R.SWINT2 bit are set to 1, the SWINT interrupt and SWINT2 interrupt occur, respectively.

15.4.3 External Pin Interrupt

An external pin interrupt is generated by signals input to the IRQi pin (i = 0 to 15). Refer to section 15.7.4, Setting the External Pin Interrupt for the procedure to set the external pin interrupt.

15.4.4 Group Interrupts

Multiple peripheral interrupt requests (up to 32 requests) are grouped together as one interrupt request. Interrupts are grouped depending on the peripheral operating clock (PCLKB or PCLKA) and method to detect interrupt requests (edge detection or level detection).

(1) Types of Group Interrupts

Table 15.6 lists types of group interrupts.

Table 15.6 Types of Group Interrupts

Interrupt Vector Number	Interrupt Name	Group Interrupt Source	
		Peripheral operating clock	Interrupt detection method
106	GROUPBE0	PCLKB	Edge detection
110	GROUPBL0		Level detection
111	GROUPBL1		
112	GROUPAL0	PCLKA	
113	GROUPAL1		

(2) Configuration of Group Interrupts

When an interrupt request is generated while the corresponding ENj bit in the group interrupt request enable register (GENBE0, GENBL0, GENBL1, GENAL0, GENAL1) is 1, the ISj flag in the group interrupt request register (GRPBE0, GRPBL0, GRPBL1, GRPAL0, GRPAL1) becomes 1 (j = 0 to 31).

Figure 15.3 shows the configuration of group interrupts.

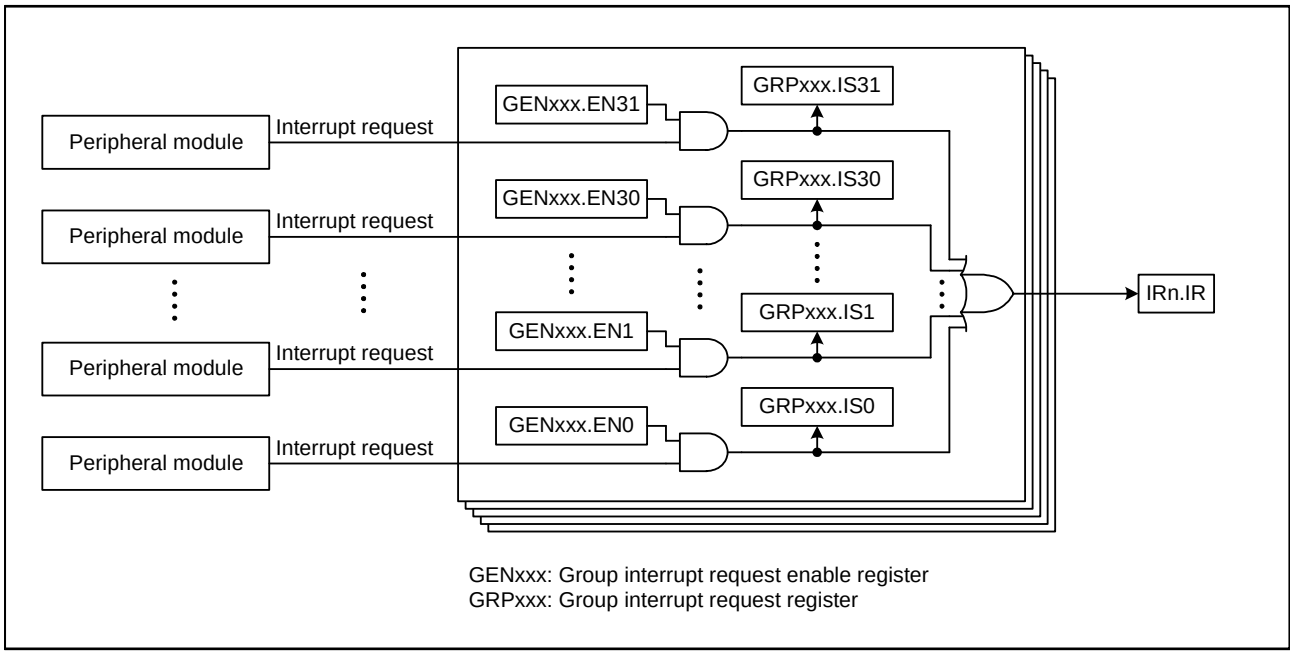


Figure 15.3 Group Interrupt Configuration (n = 106, 110 to 113)

(3) Group Interrupt Sources

Table 15.7 lists peripheral interrupt sources that are assigned to group interrupts.

Table 15.7 Group Interrupt Requests (1/3)

Group	No.	Interrupt Request Source	Name	GEN[B/A][E/L]m.ENj Bit	GRP[B/A][E/L]m.ISj Flag	GCRBE0.CLRj Bit	Vector No. (IRn.IR)
BE0	0	CAN0	ERS0 (error interrupt)	GENBE0.EN0	GRPBE0.IS0	GCRBE0.CLR0	106
	1	CAN1	ERS1 (error interrupt)	GENBE0.EN1	GRPBE0.IS1	GCRBE0.CLR1	
	2	CAN2	ERS2 (error interrupt)	GENBE0.EN2	GRPBE0.IS2	GCRBE0.CLR2	
	3 to 31	Reserved	—	—	—	—	
BL0	0	SCI0	TEI0 (transmission completed)	GENBL0.EN0	GRPBL0.IS0	—	110
	1		ERI0 (reception error)	GENBL0.EN1	GRPBL0.IS1	—	
	2	SCI1	TEI1 (transmission completed)	GENBL0.EN2	GRPBL0.IS2	—	
	3		ERI1 (reception error)	GENBL0.EN3	GRPBL0.IS3	—	
	4	SCI2	TEI2 (transmission completed)	GENBL0.EN4	GRPBL0.IS4	—	
	5		ERI2 (reception error)	GENBL0.EN5	GRPBL0.IS5	—	
	6	SCI3	TEI3 (transmission completed)	GENBL0.EN6	GRPBL0.IS6	—	
	7		ERI3 (reception error)	GENBL0.EN7	GRPBL0.IS7	—	
	8	SCI4	TEI4 (transmission completed)	GENBL0.EN8	GRPBL0.IS8	—	
	9		ERI4 (reception error)	GENBL0.EN9	GRPBL0.IS9	—	
	10	SCI5	TEI5 (transmission completed)	GENBL0.EN10	GRPBL0.IS10	—	
	11		ERI5 (reception error)	GENBL0.EN11	GRPBL0.IS11	—	
	12	SCI6	TEI6 (transmission completed)	GENBL0.EN12	GRPBL0.IS12	—	
	13		ERI6 (reception error)	GENBL0.EN13	GRPBL0.IS13	—	
	14	SCI7	TEI7 (transmission completed)	GENBL0.EN14	GRPBL0.IS14	—	
	15		ERI7 (reception error)	GENBL0.EN15	GRPBL0.IS15	—	
	16	SCI12	TEI12 (transmission completed)	GENBL0.EN16	GRPBL0.IS16	—	
	17		ERI12 (reception error)	GENBL0.EN17	GRPBL0.IS17	—	
	18		SCIX0 (Break Field Low width detection)	GENBL0.EN18	GRPBL0.IS18	—	
	19		SCIX1 (Control Field 0 match) (Control Field 1 match) (priority interrupt bit detection)	GENBL0.EN19	GRPBL0.IS19	—	
	20		SCIX2 (bus collision detection)	GENBL0.EN20	GRPBL0.IS20	—	
	21		SCIX3 (valid edge detection)	GENBL0.EN21	GRPBL0.IS21	—	
	22, 23	Reserved	—	—	—	—	
	24	QSPI	QSPSSLI (QSSL negation)	GENBL0.EN24	GRPBL0.IS24	—	
	25	Reserved	—	—	—	—	
	26	CAC	FERRI (frequency error)	GENBL0.EN26	GRPBL0.IS26	—	
	27		MENDI (measurement end)	GENBL0.EN27	GRPBL0.IS27	—	
	28		OVFI (overflow interrupt)	GENBL0.EN28	GRPBL0.IS28	—	
	29	DOC	DOPCI (data operation circuit interrupt)	GENBL0.EN29	GRPBL0.IS29	—	
	30	PDC	PCFEI (frame end interrupt)	GENBL0.EN30	GRPBL0.IS30	—	
	31		PCERI (error interrupt)	GENBL0.EN31	GRPBL0.IS31	—	

Table 15.7 Group Interrupt Requests (2/3)

Group	No.	Interrupt Request Source	Name	GEN[B/A][E/L]m.ENj Bit	GRP[B/A][E/L]m.ISj Flag	GCRBE0.CLRj Bit	Vector No. (IRn.IR)
BL1	0	SRC	OVFI (output data FIFO overflow)	GENBL1.EN0	GRPBL1.IS0	—	111
	1		UDFI (output data FIFO underflow)	GENBL1.EN1	GRPBL1.IS1	—	
	2		CEFI (conversion processing end)	GENBL1.EN2	GRPBL1.IS2	—	
	3	SDHI	CDETI (card detection interrupt)	GENBL1.EN3	GRPBL1.IS3	—	
	4		CACI (card access interrupt)	GENBL1.EN4	GRPBL1.IS4	—	
	5		SDACI (SDIO access interrupt)	GENBL1.EN5	GRPBL1.IS5	—	
	6	MMCIF	CDETIO (card detection interrupt)	GENBL1.EN6	GRPBL1.IS6	—	
	7		ERRIO (error interrupt)	GENBL1.EN7	GRPBL1.IS7	—	
	8		ACCIO (access interrupt)	GENBL1.EN8	GRPBL1.IS8	—	
	9	POE3	OEI1 (output enable interrupt 1)	GENBL1.EN9	GRPBL1.IS9	—	
	10		OEI2 (output enable interrupt 2)	GENBL1.EN10	GRPBL1.IS10	—	
	11		OEI3 (output enable interrupt 3)	GENBL1.EN11	GRPBL1.IS11	—	
	12		OEI4 (output enable interrupt 4)	GENBL1.EN12	GRPBL1.IS12	—	
	13	RIIC0	TEI0 (transmission completed)	GENBL1.EN13	GRPBL1.IS13	—	
	14		EEI0 (transmission error/event generation)	GENBL1.EN14	GRPBL1.IS14	—	
	15	RIIC2	TEI2 (transmission completed)	GENBL1.EN15	GRPBL1.IS15	—	
	16		EEI2 (transmission error/event generation)	GENBL1.EN16	GRPBL1.IS16	—	
	17	SSI0	SSIF0 (status interrupt)	GENBL1.EN17	GRPBL1.IS17	—	
	18	SSI1	SSIF1 (status interrupt)	GENBL1.EN18	GRPBL1.IS18	—	
	19	Reserved	—	—	—	—	
	20	S12AD	S12CMPI (compare interrupt)	GENBL1.EN20	GRPBL1.IS20	—	
	21	Reserved	—	—	—	—	
	22	S12AD1	S12CMPI1 (compare interrupt)	GENBL1.EN22	GRPBL1.IS22		
	23 to 31	Reserved	—	—	—	—	
AL0	0	SCIFA8	TEIF8 (transmission completed)	GENAL0.EN0	GRPAL0.IS0	—	112
	1		ERIF8 (FER/PER)	GENAL0.EN1	GRPAL0.IS1	—	
	2		BRIF8 (BRK/ORER)	GENAL0.EN2	GRPAL0.IS2	—	
	3		DRIF8 (receive data ready)	GENAL0.EN3	GRPAL0.IS3	—	
	4	SCIFA9	TEIF9 (transmission completed)	GENAL0.EN4	GRPAL0.IS4	—	
	5		ERIF9 (FER/PER)	GENAL0.EN5	GRPAL0.IS5	—	
	6		BRIF9 (BRK/ORER)	GENAL0.EN6	GRPAL0.IS6	—	
	7		DRIF9 (receive data ready)	GENAL0.EN7	GRPAL0.IS7	—	
	8	SCIFA10	TEIF10 (transmission completed)	GENAL0.EN8	GRPAL0.IS8	—	
	9		ERIF10 (FER/PER)	GENAL0.EN9	GRPAL0.IS9	—	
	10		BRIF10 (BRK/ORER)	GENAL0.EN10	GRPAL0.IS10	—	
	11		DRIF10 (receive data ready)	GENAL0.EN11	GRPAL0.IS11	—	
	12	SCIFA11	TEIF11 (transmission completed)	GENAL0.EN12	GRPAL0.IS12	—	
	13		ERIF11 (FER/PER)	GENAL0.EN13	GRPAL0.IS13	—	
	14		BRIF11 (BRK/ORER)	GENAL0.EN14	GRPAL0.IS14	—	
	15		DRIF11 (receive data ready)	GENAL0.EN15	GRPAL0.IS15	—	
	16	RSPI0	SPII0 (idle interrupt)	GENAL0.EN16	GRPAL0.IS16	—	
	17		SPEI0 (error interrupt)	GENAL0.EN17	GRPAL0.IS17	—	
	18 to 31	Reserved	—	—	—	—	

Table 15.7 Group Interrupt Requests (3/3)

Group	No.	Interrupt Request Source	Name	GEN[B/A][E/L]m.ENj Bit	GRP[B/A][E/L]m.ISj Flag	GCRBE0.CLRj Bit	Vector No. (IRn.IR)
AL1	0	EPTPC	MINT (status interrupt)	GENAL1.EN0	GRPAL1.IS0	—	113
	1	PTPEDMAC	PINT (status interrupt)	GENAL1.EN1	GRPAL1.IS1	—	
	2	Reserved	—	—	—	—	
	3	Reserved	—	—	—	—	
	4	EDMAC0	EINT0 (status interrupt)	GENAL1.EN4	GRPAL1.IS4	—	
	5	EDMAC1	EINT1 (status interrupt)	GENAL1.EN5	GRPAL1.IS5	—	
	6 to 31	Reserved	—	—	—	—	

15.4.5 Software Configurable Interrupts

An interrupt source assigned to each interrupt vector number from 128 to 255 can be selected from multiple sources. They are divided into software configurable interrupt B and software configurable interrupt A depending on the peripheral operating clock. Figure 15.4 shows the software configurable interrupt configuration.

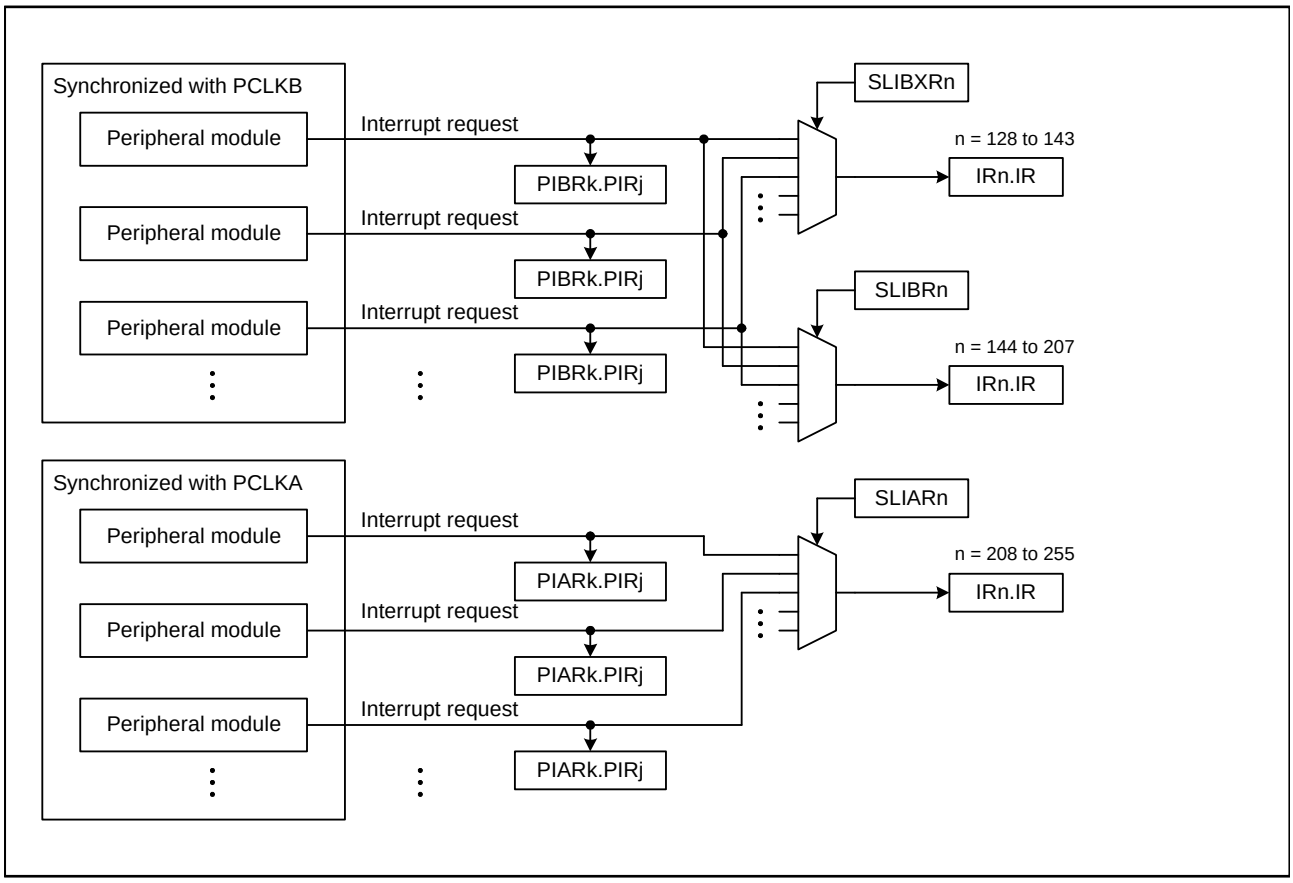


Figure 15.4 Software Configurable Interrupt Configuration

15.4.5.1 Software Configurable Interrupt B

For interrupt sources assigned to software configurable interrupts, interrupt sources of peripherals that operates in synchronization with PCLKB can be assigned to interrupt number from 128 to 207. The abbreviation for software configurable interrupt B is PERIB. Interrupt names are indicated by INTB128 to INTB207.
Refer to Table 15.3, Interrupt Sources for Software Configurable Interrupt B for interrupt sources that can be assigned to software configurable interrupt B.

15.4.5.2 Software Configurable Interrupt A

For interrupt sources assigned to software configurable interrupts, interrupt sources of peripherals that operates in synchronization with PCLKA can be assigned to interrupt number from 208 to 255. The abbreviation for software configurable interrupt A is PERIA. Interrupt names are indicated by INTA208 to INTA255.
Refer to Table 15.4, Interrupt Sources for Software Configurable Interrupt A for interrupt sources that can be assigned for software configurable interrupt A.

15.4.5.3 EXDMAC Start Trigger by Software Configurable Interrupts

Some interrupt sources assigned to software configurable interrupt B and software configurable interrupt A can be used as a trigger for EXDMAC0 or EXDMAC1.
To use an interrupt source as the EXDMAC0 trigger, assign the interrupt source to interrupt vector number 144 (INTB144) or 208 (INTA208). To use an interrupt source as the EXDMAC1 trigger, assign the interrupt source to interrupt vector number 145 (INTB145) or 208 (INTA209).
Only the TPU1.TGI1A interrupt of software configurable interrupt B and the MTU1.TGIA1 interrupt of software configurable interrupt A can be used as the EXDMAC trigger.
Triggers for EXDMAC0 and EXDMAC1 can be selected by setting bits SELEXDR.SELEXD0 and SELEXD1, respectively. Table 15.8 lists correspondences between the SELEXDR register settings and interrupt sources to start the EXDMAC.

Table 15.8 EXDMAC Start Triggers by Software Configurable Interrupts

SELEXDR Register	Bit Value	Interrupt Vector Number	Interrupt Request
SELEXD0 bit (EXDMAC0)	0	144 (INTB144)	TPU1.TGI1A (input capture/compare match)
	1	208 (INTA208)	MTU1.TGIA1 (input capture/compare match)
SELEXD1 bit (EXDMAC1)	0	145 (INTB145)	TPU1.TGI1A (input capture/compare match)
	1	209 (INTA209)	MTU1.TGIA1 (input capture/compare match)

15.4.6 Non-Maskable Interrupts

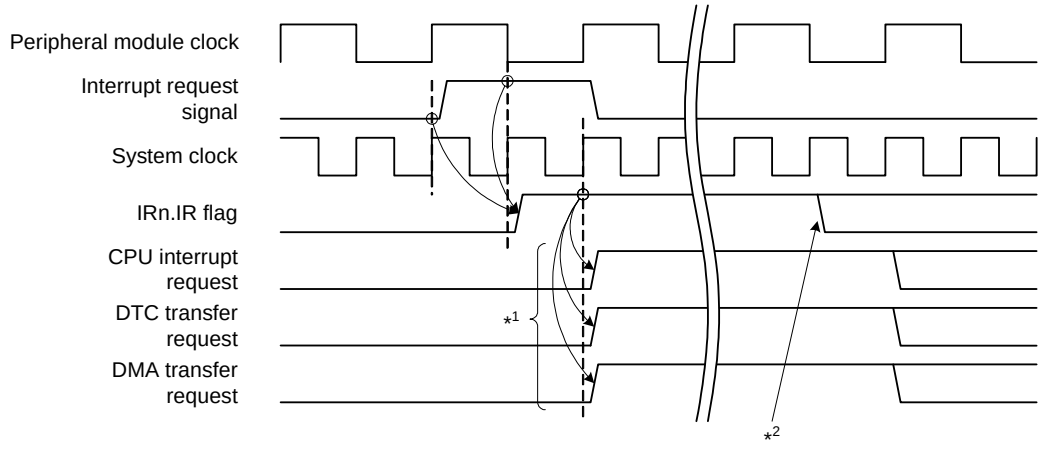
Non-maskable interrupts include the NMI pin interrupt, oscillation stop detection interrupt, WDT underflow/refresh error interrupt, IWDT underflow/refresh error interrupt, voltage monitoring 1 interrupt, voltage monitoring 2 interrupt, and RAM error interrupt.
Non-maskable interrupts have the highest priority in all interrupts, including the fast interrupt, so are accepted regardless of the settings of the PSW.I bit (interrupt enable) and IPL[3:0] bits (processor interrupt priority level) in the CPU.
The NMISR register can be used to confirm whether a non-maskable interrupt is generated.
Only the CPU can be selected as the interrupt request destination of non-maskable interrupt. The DTC and DMAC cannot be selected.

15.5 Interrupt Detection

Level detection or edge detection can be used for detecting an interrupt request.
For interrupt requests from the peripherals, edge detection or level detection is fixed for each interrupt source. For interrupt requests of the external pin interrupt, edge detection or level detection can be selected with the IRQCRi.IRQMD[1:0] bits.
Refer to Table 15.5, Interrupt Vector Table for the method to detect each interrupt request.
For group interrupts, interrupt sources are grouped depending on the method to detect interrupt requests.
Interrupt requests by interrupt sources assigned to group BE0 are detected by edge detection. Interrupt requests by interrupt sources assigned to groups BL0, BL1, AL0, and AL1 are detected by level detection. Note that group interrupts (GROUPBE0, GROUPBL0/GROUPBL1, GROUPAL0/GROUPAL1) are detected by level detection.
Refer to section 15.4.4, Group Interrupts for group interrupts. Refer to section 15.5.3, Group Interrupts Using Edge Detection and section 15.5.4, Group Interrupts Using Level Detection for interrupt requests of group interrupts.

15.5.1 Edge Detection

Figure 15.5 shows the operation of the IRn.IR flag at edge detection (n = 023 to 255).
The IRn.IR flag becomes 1 when the rising edge of the interrupt request signal is detected. Then, the IRn.IR flag does not become 0 by disabling the interrupt request of the peripheral module. When the CPU accepts the interrupt request or the DTC/DMAC accepts the transfer request, the IRn.IR flag automatically becomes 0. It is not required to set the IRn.IR flag to 0 by software. Refer to Table 15.9, Operations When Starting the DMAC/DTC for details on clearing the IRn.IR flag by DTC/DMAC.
For the external pin interrupts of interrupt vector number 64 to 79 and interrupt sources of interrupt vector number 88 to 95, the timing when the IRn.IR flag becomes 1 after an interrupt signal occurs is different from the other interrupts. For the external pin interrupts, the timing is delayed for internal delay plus two cycles of PCLKB after a signal is input to the IRQ pin (i = 0 to 15). For interrupts of interrupt vector number 88 to 95, the timing is delayed for 2 cycles of PCLKB.



Note 1. A CPU interrupt request, DTC transfer request, or DMA transfer request is generated. Refer to section 15.7.3.1, Interrupt Request Destination Setting Procedure.
Note 2. For a CPU interrupt request, the IRn.IR flag becomes 0 when the CPU accepts an interrupt request. For the timing for a DTC transfer request or DMA transfer request, refer to Table 15.9, Operation when starting the DMAC/DTC.

Figure 15.5 IRn.IR Flag Operation for Interrupts Detected by Edge Detection

(1) Detecting Consecutive Interrupt Request Signals

When interrupt request signals occur every cycle, the latter interrupt signal cannot be detected. To accept consecutive interrupt request signals, the interval of at least two cycles of the system clock or peripheral module clock, whichever is lower, is required between interrupt signals. Figure 15.6 shows the interval for accepting consecutive interrupt request signals.

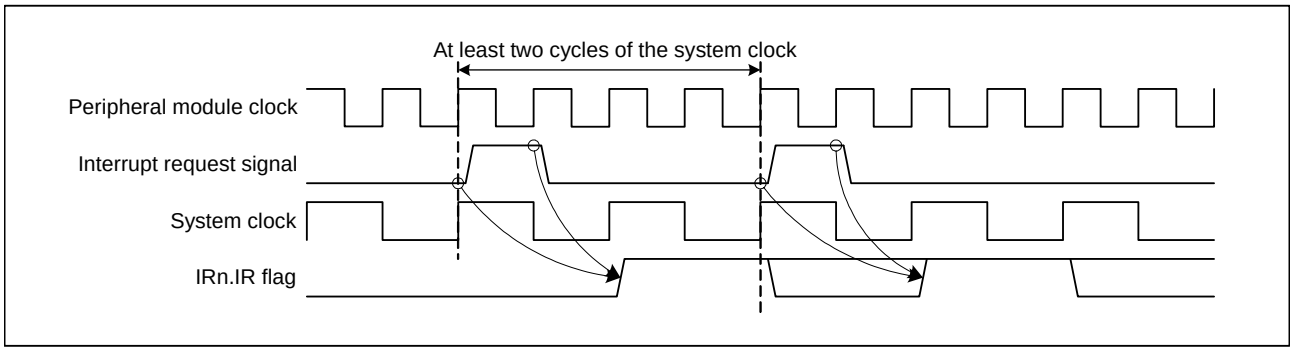


Figure 15.6 Accepting Consecutive Interrupt Signals (when the system clock frequency is lower than the peripheral clock frequency)

When an interrupt request is generated again while the IRn.IR flag is 1, the interrupt request is ignored (n = 023 to 255). However, for transmit interrupt requests, receive interrupt requests, and buffer access interrupt requests of the SCI, RIIC, RSPI, QSPI, SDHI, MMCIF, and PDC, when an interrupt request occurs while the IRn.IR flag is 1, the interrupt request is retained in the module. After the IRn.IR flag becomes 0, the IRn.IR flag is set to 1 again by the retained request. Refer to the descriptions for interrupts in each chapter of peripheral modules for details.
Figure 15.7 shows the timing when the IRn.IR flag is set again.

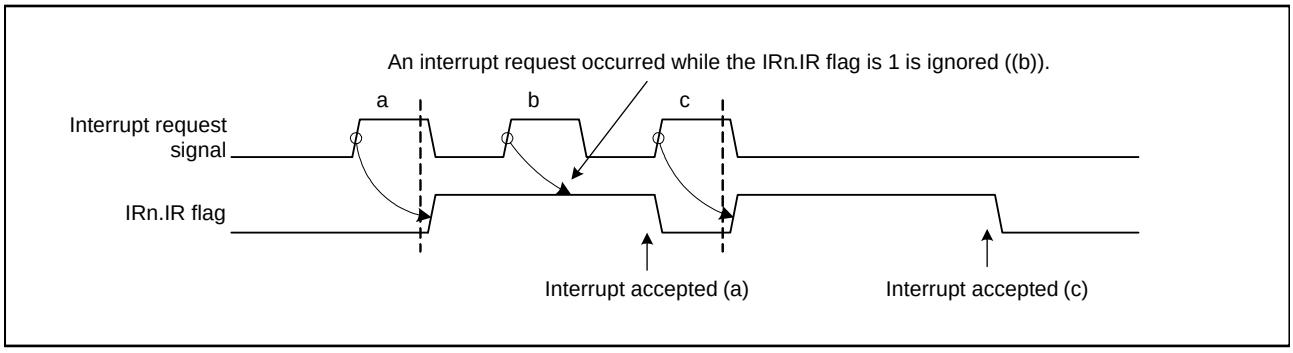


Figure 15.7 Timing to Set the IRn.IR Flag Again

(2) Relation between the IRn.IR flag and Interrupt Request Enable Bits

After the IRn.IR flag becomes 1, the IRn.IR flag does not become 0 even when an interrupt request enable bit in the corresponding peripheral module is set to 0.

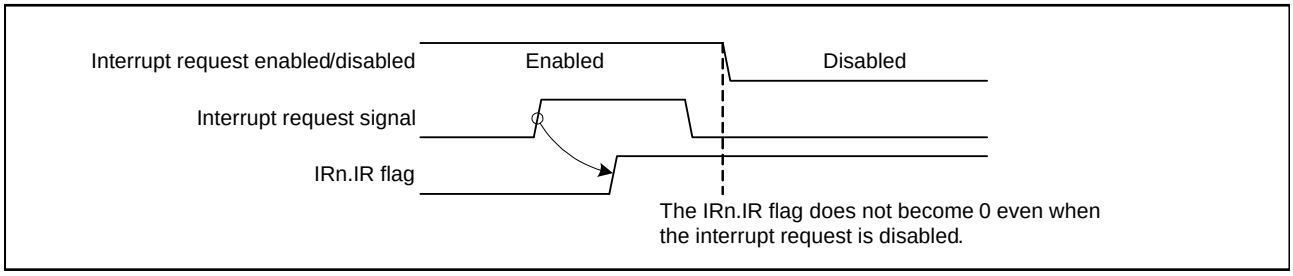


Figure 15.8 Relation Between Disabling the Interrupt Request and the IRn.IR Flag

15.5.2 Level Detection

Figure 15.9 shows operation of the interrupt request signal and the IRn.IR flag for level detection (n = 016 to 124). The IRn.IR flag is 1 while the interrupt request signal is 1. To set the IRn.IR flag to 0, set the corresponding interrupt request signal of the peripheral module to 0. Set the corresponding interrupt status flag of the peripheral module to 0 and wait for the time until the value is reflected in the IRn.IR flag before exiting the interrupt handler. Refer to (2) Notes on writing to I/O registers in section 5, I/O Registers for details on waiting for the reflection.

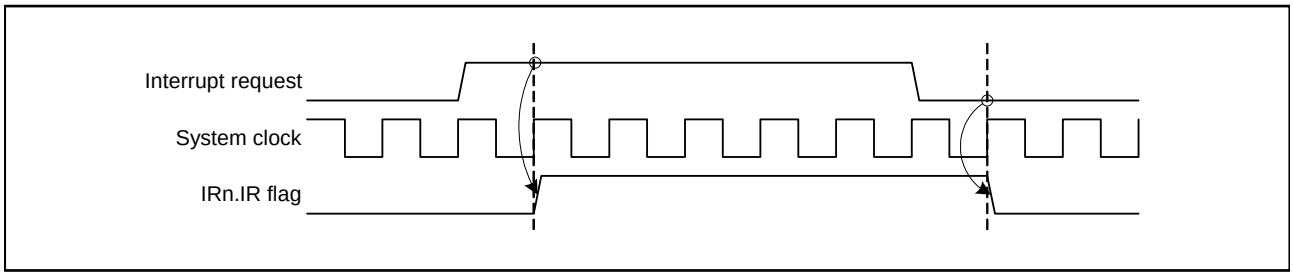


Figure 15.9 IRn.IR Flag Operation for Level Detection

Figure 15.10 shows an example of the procedure to handle interrupts for level detection.

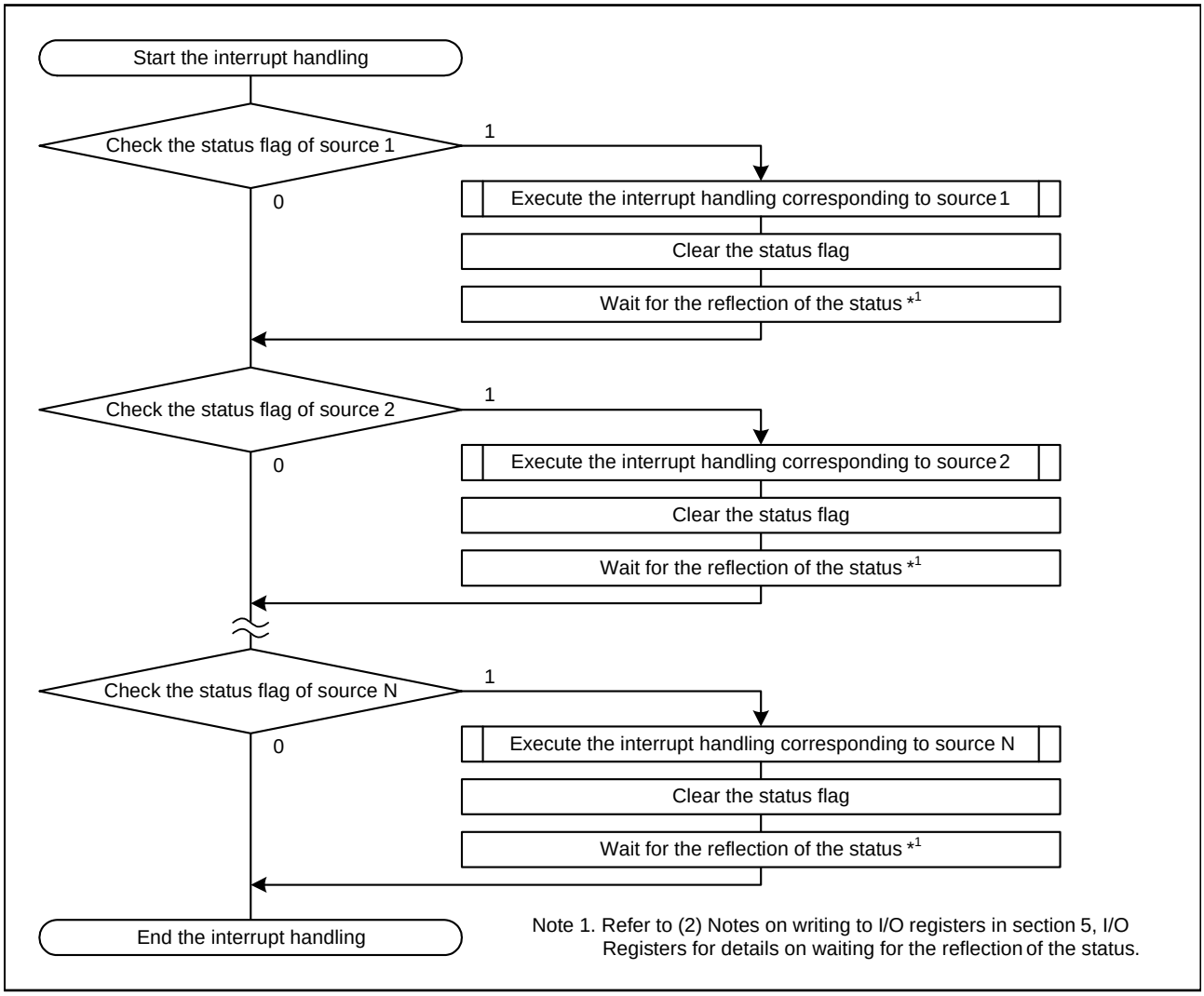


Figure 15.10 Example of Level Detection Interrupt Handling Procedure (N indicates the number of status flags)

15.5.3 Group Interrupts Using Edge Detection

Group BE0 of group interrupts includes interrupt sources that are detected by edge detection.
While the IR106.IR flag corresponding to the GROUPBE0 interrupt becomes 1 under the same conditions as edge detection, the IR106.IR flag becomes 0 under the same conditions as level detection.
When the rising edge of an interrupt request signal is detected while the corresponding GENBE0.ENj bit is 1, both the GRPBE0.ISj flag and IR106.IR flag become 1. Then, GRPBE0.ISj flag and IR106.IR flag do not become 0 by disabling the interrupt request of the peripheral module or setting the GENBE0.ENj bit to 0.
When the GCRBE0.CLRj bit is set to 1, the GRPBE0.ISj flag becomes 0, and consequently the IR106.IR flag becomes 0.
Figure 15.11 shows an operation example of group interrupts using edge detection. Figure 15.12 shows an example of operation when interrupt requests are generated by multiple interrupt sources in the same group.

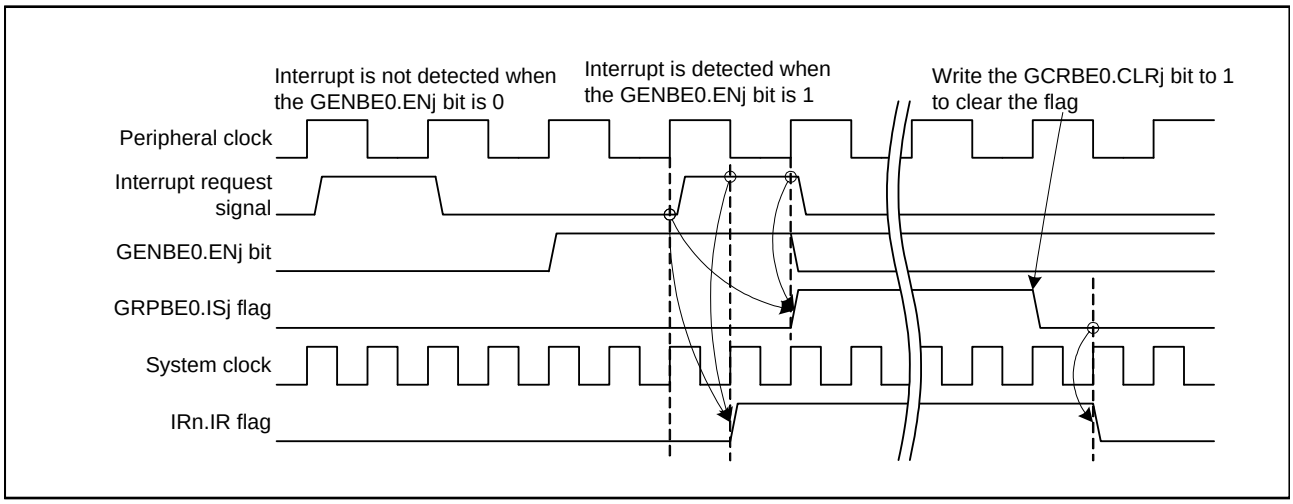


Figure 15.11 Example of Interrupt Request for Group Interrupt Using Edge Detection

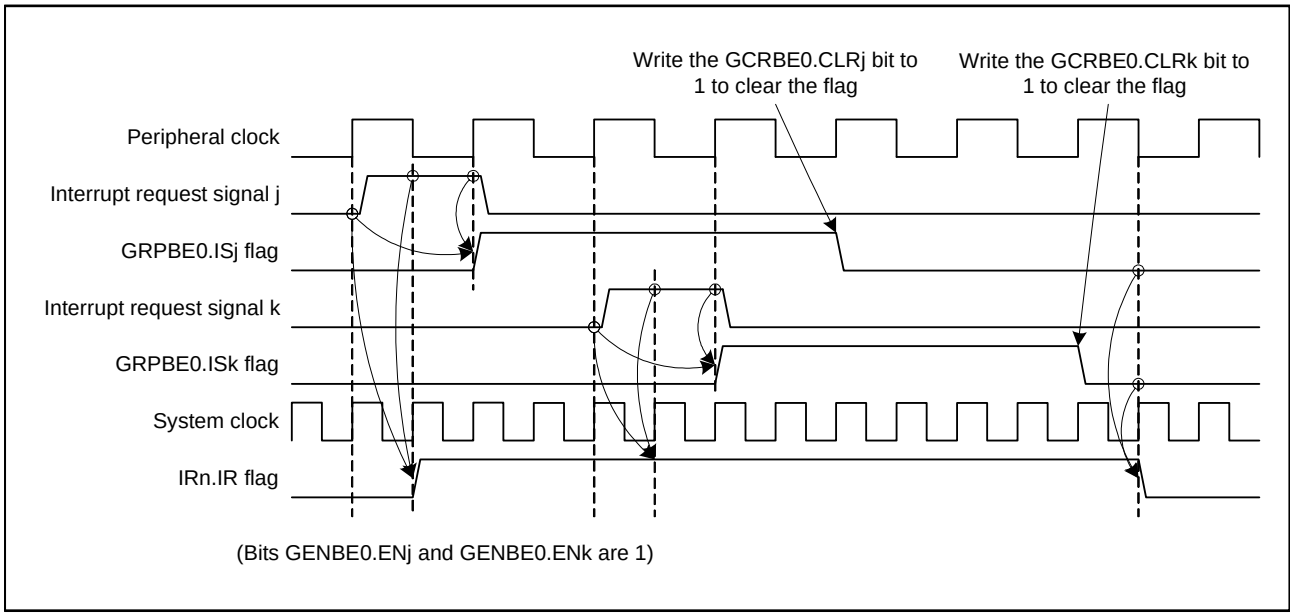


Figure 15.12 Example of Operation When Multiple Edge Detection Interrupt Requests Are Generated in the Same Group

Figure 15.13 shows an example of the procedure to handle group interrupts using edge detection.

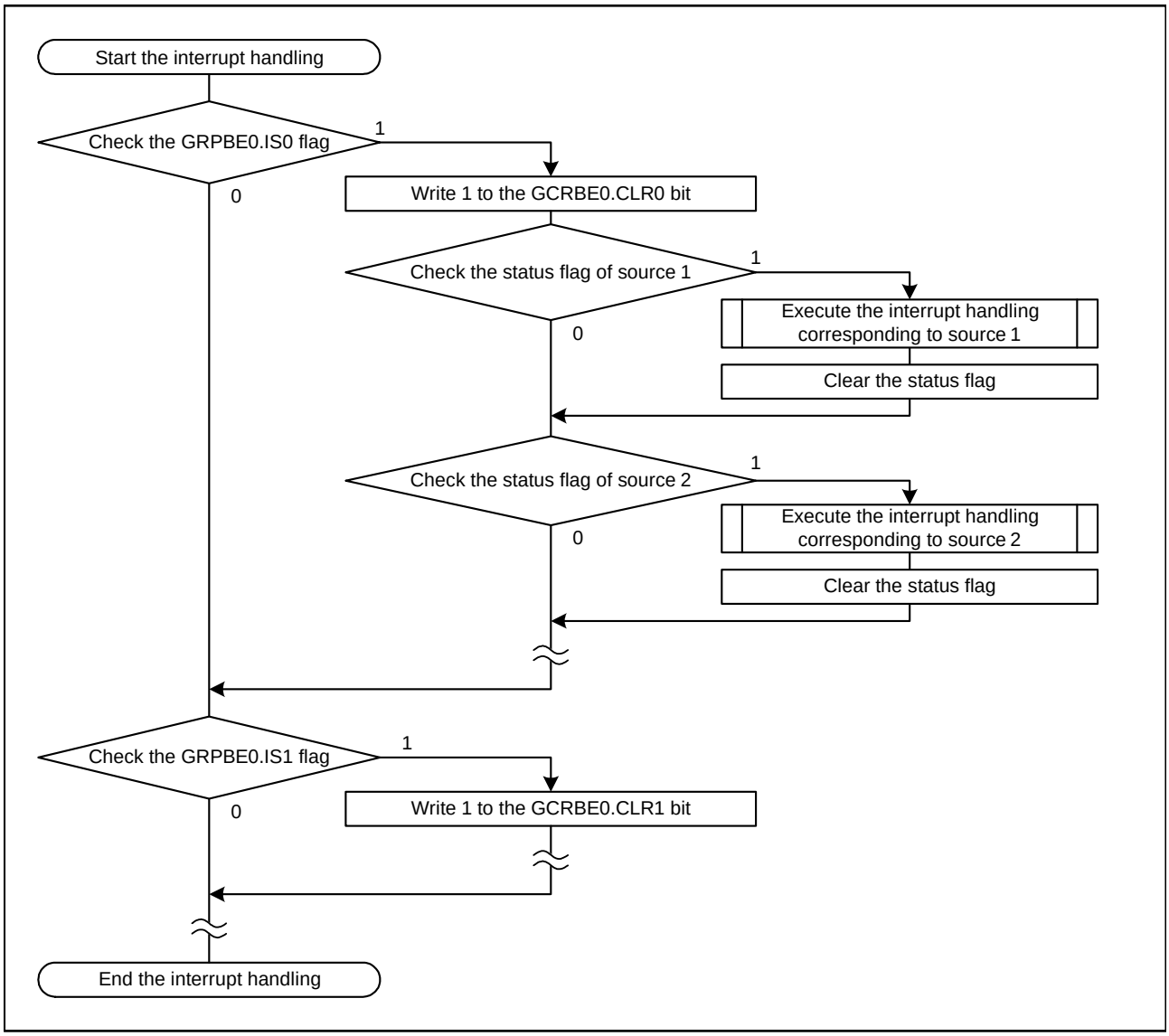


Figure 15.13 Example of Procedure to Handle Group Interrupts Using Edge Detection

15.5.4 Group Interrupts Using Level Detection

Groups BL0/BL1 and AL0/AL1 of group interrupts includes interrupt sources that are detected by level detection.
The IRn.IR flag corresponding to the GROUPBL0/GROUPBL1 and GROUPAL0/GROUPAL1 interrupts changes under the same conditions as level detection (n = 110 to 113).
When an interrupt signal becomes 1 while the corresponding GENBL0/GENBL1.ENj bit or GENAL0/GENAL1.ENj bit is 1, the GRPBL0/GRPBL1.ISj flag or GRPAL0/GRPAL1.ISj flag and the IRn.IR flag become 1. Then, the GRPBL0/GRPBL1.ISj flag or GRPAL0/GRPAL1.ISj flag and IRn.IR flag becomes 0 when the corresponding interrupt request signal becomes 0. Also, when the GENBL0/GENBL1.ENj bit or GENAL0/GENAL1.ENj bit is set to 0, the corresponding GRPBL0/GRPBL1.ISj flag or GRPAL0/GRPAL1.ISj flag and IRn.IR flag become 0.
Figure 15.14 shows an operation example of group interrupts using edge detection. Figure 15.15 shows an example of operation when interrupt requests are generated by multiple interrupt sources in the same group.

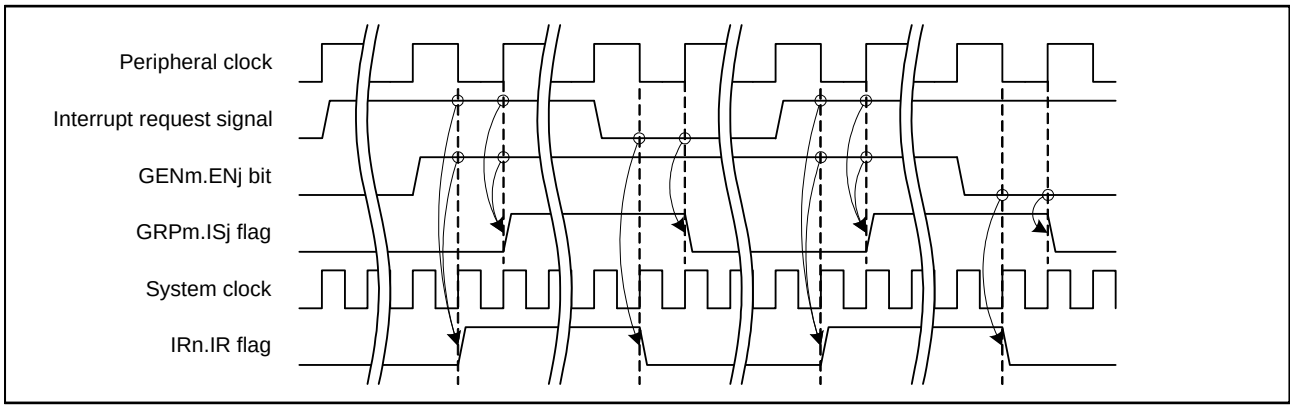


Figure 15.14 Operation Example of Group Interrupt Using Level Detection (m = BL0, BL1, AL0, AL1)

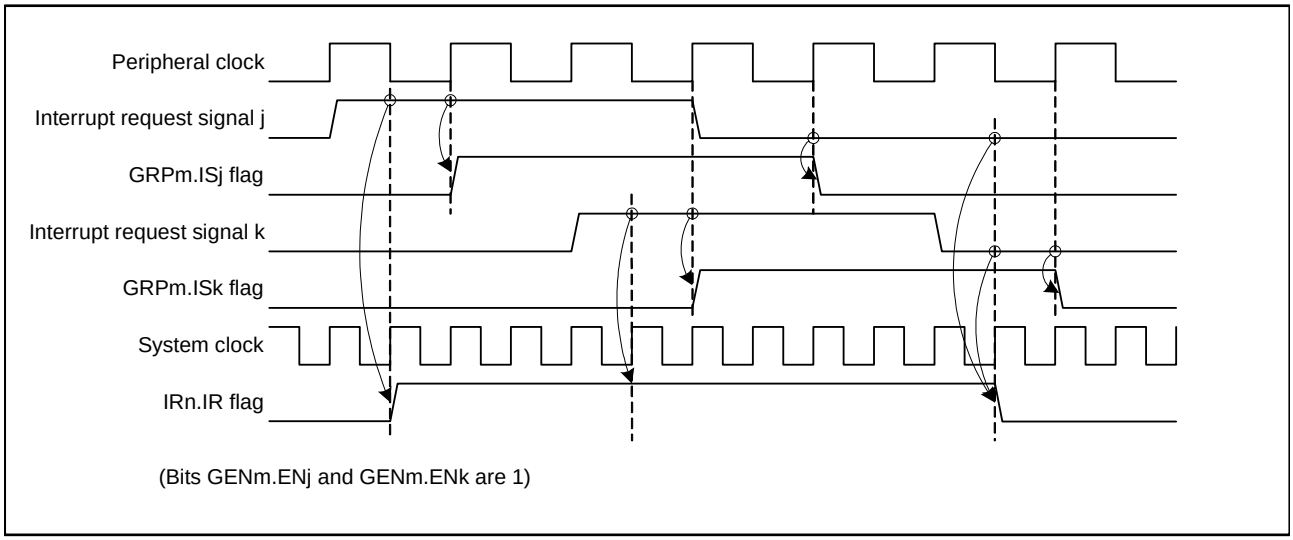


Figure 15.15 Operation Example When Multiple Interrupt Requests are Generated in the Same Group (m = BL0, BL1, AL0, AL1)

Figure 15.16 shows the procedure to handle group interrupts for level detection.

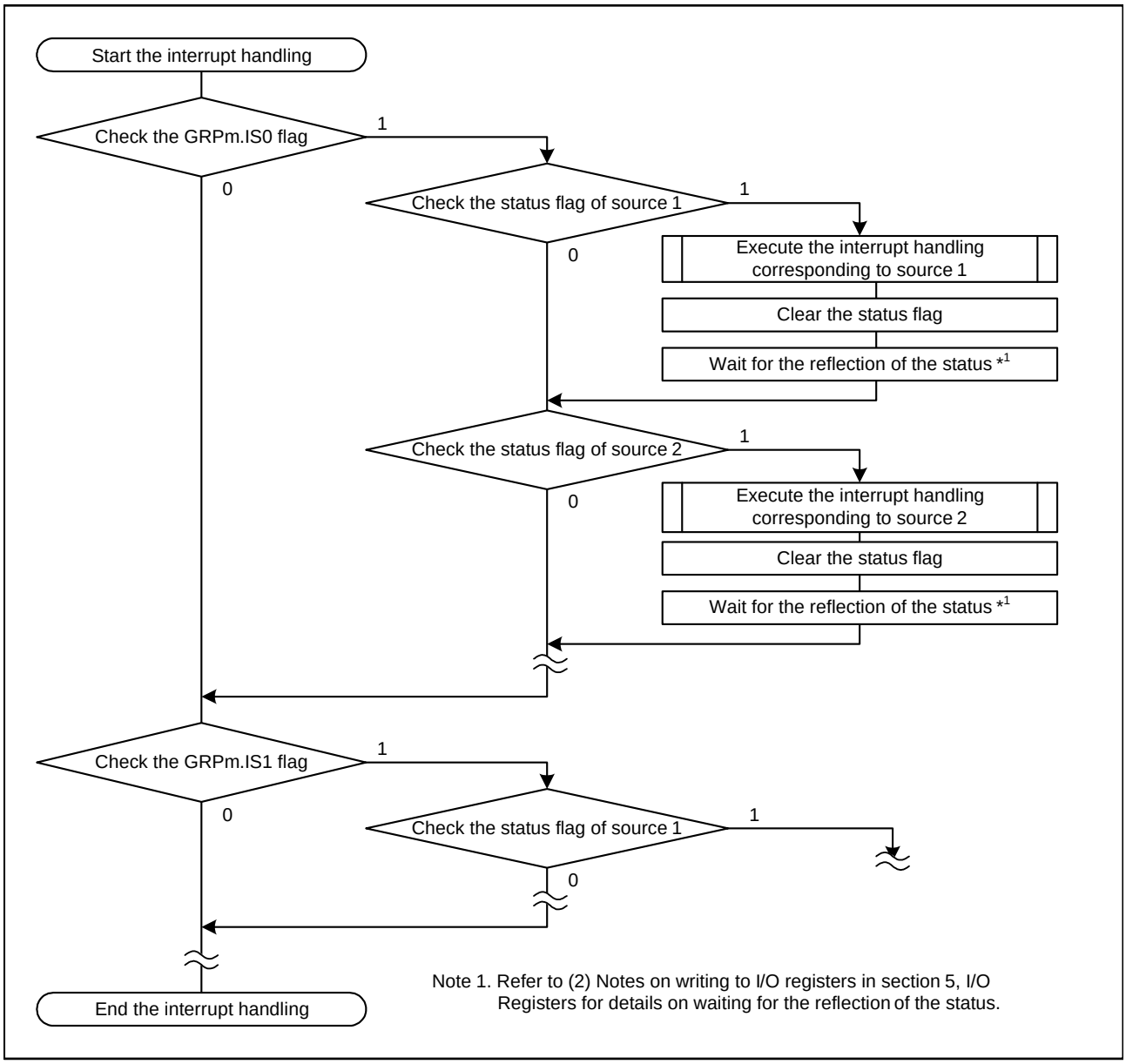


Figure 15.16 Example of Group Interrupt Handling Procedure for Level Detection (m = BL0, BL1, AL0, AL1)

15.5.5 Software Configurable Interrupts

Interrupt sources and interrupt requests for software configurable interrupts are detected by edge detection.
Figure 15.17 shows an operation example of the interrupt request and interrupt status flag for software configurable interrupts.

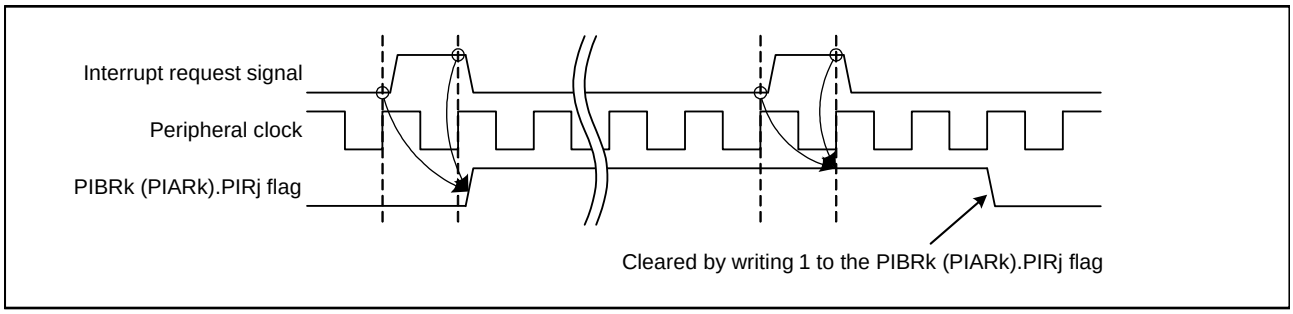


Figure 15.17 Operation Example of the Interrupt Request and Interrupt Status Flag for Software Configurable Interrupts

15.6 Determining Priority of Interrupt Requests

The ICU determines the priority for each interrupt request destination. The priority for each interrupt request destination is determined as follows.

(1) Determining Priority when the CPU is the Interrupt Request Destination

A source selected for the fast interrupt has the highest priority. Next to the fast interrupt, the priority is determined by the IPRr.IPR[3:0] bit value, and an interrupt source with a larger value has priority (r = 000 to 255). If multiple sources has the same IPRr.IPR[3:0] bit value, the priority is determined by the interrupt vector number, and the source with the smaller number has priority.

(2) Determining Priority when the DTC is the Interrupt Request Destination

The IPRr.IPR[3:0] bits have no effect (r = 000 to 255). The priority is determined by only the interrupt vector number, and an interrupt source with a smaller number has priority.

(3) Determining Priority when the DMAC is the Interrupt Request Destination

The IPRr.IPR[3:0] bits have no effect. The priority is determined by the DMAC channel number. Refer to section 18, DMA Controller (DMACAa) for details on the DMAC channel priority.

15.7 Interrupt Setting Procedure

15.7.1 Enabling Interrupt Requests

The following describes the procedure to enable interrupt requests.

1. Set the interrupt request enable bit of the peripheral modules to enable output of the interrupt request.
2. For group interrupts, set the corresponding ENj bit in the group interrupt request enable register to 1 to enable output of the interrupt request to the ISj flag in the group interrupt request register (j = 0 to 31).
3. Set the corresponding IERm.IENj bit to 1 to enable output of the interrupt request to the interrupt request destination (j = 0 to 7; m = 02h to 1Fh).

After the above procedure is completed, when a peripheral interrupt occurs, the IRn.IR flag corresponding to the interrupt source becomes 1 (n = 016 to 255).

For group interrupts, the ISj flag in the group interrupt request register becomes 1, the IRn.IR flag corresponding to the group becomes 1, and an interrupt request is output to the interrupt request destination.

When the IERm.IENj bit is 0, the interrupt request corresponding to the interrupt source is not output to the interrupt request destination.

15.7.2 Disabling Interrupt Requests

The following describes the procedure to disable interrupt requests.

1. Set the corresponding IERm.IENj bit to 0 (j = 0 to 7; m = 02h to 1Fh).
2. For group interrupts, set the corresponding ENj bit in the group interrupt request enable register to 0 to disable output of the interrupt request to the ISj flag in the group interrupt request register (j = 0 to 31).
3. Set the interrupt request enable bit of the peripheral modules to disable output of the interrupt request. Read the register that has been set to confirm that the value is reflected.
4. As needed, read the IRn.IR flag or set the IR flag to 0. *1
   For group interrupts, confirm that the ISj flag in the group interrupt request register is 0 or set the ISj flag to 0 (j = 0 to 31).

Note 1. When disabling the transmit interrupt request, receive interrupt request, or buffer access interrupt requests of the SCI, RIIC, RSPI, QSPI, SDHI, MMCIF, or PDC, set the IRn.IR flag to 0 according to the above procedure. Refer to the description of interrupts in the corresponding section of peripheral modules for details.

15.7.3 Selecting Interrupt Request Destination

15.7.3.1 Interrupt Request Destination Setting Procedure

The destination of an interrupt request can be selected from the CPU, DTC, or DMAC for each interrupt source. Destinations that can be selected differ depending on the interrupt source. Refer to Table 15.5, Interrupt Vector Table for details on the destinations. Do not select a destination that is not indicated as "✓" in Table 15.5.
When set the external pin interrupt as the DTC or DMAC trigger, set the IRQCRi.IRQMD[1:0] bits to select edge detection (i = 0 to 15).
The following describes the procedure to select a destination of an interrupt request.

(1) Setting Interrupt Sources as the DMAC trigger

Perform the following settings while the IERm.IENj bit is 0 of the interrupt source that is selected as the DMAC trigger (j = 0 to 7; m = 02h to 1Fh).

1. Set the interrupt vector number of the interrupt source used as the DMAC trigger in the DMRSRm register corresponding to the DMAC channel (m = DMAC channel number). *1
2. Set the DMTMD.DCTG[1:0] bits corresponding to the DMAC channel to 01b in order to select the peripheral interrupt or external pin interrupt as the DMAC trigger.
3. Set the DMCNT.DTE bit corresponding to the DMAC channel to 1.

After the above settings are completed, set the corresponding IERm.IENj bit to 1.
Also, set the DMAST.DMST bit to 1 before or after the above settings.
Refer to section 18.3.7, Activating the DMAC in section 18, DMA Controller (DMACAa) for the procedure to set the DMAC.

(2) Setting Interrupt Sources as the DTC trigger

Perform the following setting while the IERm.IENj bit of the interrupt source that is selected as the DTC trigger is 0 (j = 0 to 7; m = 02h to 1Fh).

1. Set the DTCERn.DTCE bit corresponding to the interrupt vector number n used for the DTC trigger to 1 (n = 026 to 255). *1

After the above setting is completed, set the IERm.IENj bit to 1.
Also set the DTCST.DTCST bit to 1 before or after the above settings.
Refer to section 20.5, DTC Setting Procedure in section 20, Data Transfer Controller (DTCa) for the procedure to set the DTC.

Note 1. Do not set the same interrupt source as DTC and DMAC triggers. Also, do not set the same interrupt source as triggers of multiple DMAC channels.

(3) Setting Interrupt Sources for the CPU

When an interrupt source is not selected as the DMAC or DTC trigger, the interrupt request is output to the CPU.
Set the IERm.IENj bit to 1 while the interrupt source is not selected as a DTC or DMAC trigger (j = 0 to 7; m = 02h to 1Fh).

15.7.3.2 Operations When the DTC/DMAC Selected

Table 15.9 lists operations when the DMAC or DTC is set as an interrupt request destination.

Table 15.9 Operations When Starting the DMAC/DTC

Interrupt Request Destination	DISEL *1	Number of Remaining Transfers	Operation per Request	IR Flag Clear Timing *2	Interrupt Request Destination after Transfer
DTC *3	1	≠ 0	DTC transfer → CPU interrupt	Cleared when the CPU accepts an interrupt request.	DTC
		= 0	DTC transfer → CPU interrupt	Cleared when the CPU accepts an interrupt request.	CPU (DTCERn.DTCE bit becomes 0)
	0	≠ 0	DTC transfer	Cleared when the DTC starts data transfer.	DTC
		= 0	DTC transfer → CPU interrupt *4	Cleared when the CPU accepts an interrupt request. *4	CPU (DTCERn.DTCE bit becomes 0)
DMAC	1	≠ 0	DMA transfer → CPU interrupt	Cleared when the CPU accepts an interrupt request.	DMAC
		= 0	DMA transfer → CPU interrupt	Cleared when the CPU accepts an interrupt request.	CPU (DMACm.DMCNT.DTE bit becomes 0)
	0	≠ 0	DMA transfer	Cleared when the DMAC starts data transfer.	DMAC
		= 0	DMA transfer *4	Cleared when the DMAC starts data transfer. *4	CPU (DMACm.DMCNT.DTE bit becomes 0)

Note 1. For the DTC, set the DTC.MRB.DISEL bit; For the DMAC, set the DMACm.DMCSL.DISEL bit.
Note 2. When the IRn.IR flag is 1, an interrupt request (DTC or DMA transfer request) that is generated again is ignored.
Note 3. For chain transfer, the DTC transfer continues until the chain transfer ends. After chain transfer ends, whether a CPU interrupt occurs, the IRn.IR flag clear timing, and the interrupt request destination differ depending on the DISEL bit value the number of remaining transfers. For the chain transfer, refer to Table 20.3, Chain Transfer Conditions in section 20, Data Transfer Controller (DTCa).
Note 4. When the DISEL bit is 0 and the number of remaining transfers is 0, operations in the DTC and DMAC are different.

15.7.3.3 Changing the Interrupt Request Destination

Set the IERm.IENj bit to 0 before changing the interrupt request destination (j = 0 to 7; m = 02h to 1Fh).

(1) When the current interrupt request destination is the DMAC

To change interrupt request destinations or change the DMAC trigger to another interrupt source while the DMA transfer is not completed (DMCNT.DTE bit is not cleared) after the procedure described in (1) Setting Interrupt Sources as the DMAC trigger of section 15.7.3.1, Interrupt Request Destination Setting Procedure, follow the procedure below.

1. Set the IERm.IENj bit of both the current and new triggers to 0 (j = 0 to 7; m = 02h to 1Fh).
2. Check the DMAC transfer status. If transfer is not completed, wait until the completion of transfer.
3. Perform the procedure described in 15.7.3.1 Interrupt Request Destination Setting Procedure.

(2) When the current interrupt request destination is the DTC

To change interrupt request destinations or change the DTC transfer information while the DTC transfer is not completed (the DTCERn.DTCE bit is not cleared) after the procedure described in (2) Setting Interrupt Sources as the DTC trigger of section 15.7.3.1, Interrupt Request Destination Setting Procedure, follow the procedure below (n = 026 to 255).

1. Set the IERm.IENj bit of both the current and new triggers to 0 (j = 0 to 7; m = 02h to 1Fh).
2. Check the DTC transfer status. If transfer is not completed, wait until the completion of transfer.
3. Perform the procedure described in 15.7.3.1 Interrupt Request Destination Setting Procedure.

15.7.4 Setting the External Pin Interrupt

The following describes the procedure to use the external pin interrupt.

(1) Set the IERm.IENj bit corresponding to the IRQi pin to 0 (interrupt request is disabled) (i = 0 to 15; j = 0 to 7; m = 02h to 1Fh).
(2) Set the IRQFLTE0.FLTENi or IRQFLTE1.FLTENi bit to 0 (digital filter is disabled).
(3) Set the IRQFLTC0.FCLKSELi[1:0] or IRQFLTC1.FCLKSELi[1:0] bits to select the sampling clock of the digital filter.
(4) Set the I/O port and confirm the setting.
(5) Set the IRQCRi.IRQMD[1:0] bits to select the detection method.
(6) When edge detection is selected, set the corresponding IRn.IR flag to 0 (n = 016 to 255).
(7) Set the IRQFLTE0.FLTENi or IRQFLTE1.FLTENi bit to 1 (digital filter is enabled).
(8) To select the DTC as the interrupt request destination, set the DTCERn.DTCE bit. To select the DMAC as the interrupt request destination, set the DMRSRm register. When neither the DTCERn.DTCE bit nor the DMRSRm register is set, the interrupt request is sent to the CPU (m = DMAC channel number; n = 026 to 255).
(9) Set the corresponding IERm.IENj bit to 1 (interrupt request is enabled).

15.7.5 Setting Non-Maskable Interrupts

After reset, non-maskable interrupts are disabled. To use non-maskable interrupts, follow the procedure below.

(1) Set the stack pointer (SP).
(2) When using the NMI pin, set the NMIFLTE.NFLTEN bit to 0 (digital filter is disabled).
(3) When using the NMI pin, set the NMIFLTC.NFCLKSEL[1:0] bits to select the sampling clock of the digital filter.
(4) When using the NMI pin, set the NMICR.NMIMD bit to select the edge for detection.
(5) When using the NMI pin, write 1 to the NMICLR.NMICLR bit to set the NMISR.NMIST flag to 0.
(6) When using the NMI pin, set the NMIFLTE.NFLTEN bit to 1 (digital filter is enabled).
(7) To enable generation of the non-maskable interrupt, set the bit in the NMIER register corresponding to the used interrupt source to 1.

Once a bit in the NMIER register is set to 1 (enabled), the bit cannot be rewritten so it cannot be set to 0 (disabled). To disable a non-maskable interrupt that has been enabled, reset the MCU.
Refer to section 14, Exception Handling for details on the flow of non-maskable interrupt handling.
Excluding the ECCRAMST flag, each flag in the NMISR register becomes 0 by writing 1 to the corresponding bit in the NMICLR register. To set the ECCRAMST flag to 0, set the RAMSTS.RAMERR flag, the ECCRAM2STS.ECC2ERR flag, or the ECCRAM1STS.ECC1ERR flag that has become 1 to 0.
Confirm that all flags in the NMISR register are 0 before exiting the interrupt handler of non-maskable interrupts.
Non-maskable interrupts, excluding the NMI pin interrupt, can be used as a maskable interrupt. When using as a maskable interrupt, do not change the NMIER register value from the value after reset. In addition, set the LVD1CR1.LVD1IRQSEL bit and LVD2CR1.LVD2IRQSEL bit to 1 when using voltage monitoring 1 interrupt and voltage monitoring 2 interrupt as a maskable interrupt.

15.7.6 Digital Filter

Noise included in signals input to pins IRQi and NMI can be reduced by enabling the digital filter (i = 0 to 15).
The digital filter samples signals input to pins using the sampling clock (PCLKB, PCLKB/8, PCLKB/32, PCLKB/64) for the digital filter, and passes the input signal only when three consecutive sampled signals are the same level.
When using the digital filter for the IRQi pin, refer to section 15.7.4, Setting the External Pin Interrupt to set the associated registers. When using the digital filter for the NMI pin, refer to section 15.7.5, Setting Non-Maskable Interrupts to set the associated registers.
When the external pin interrupt or the NMI pin interrupt is used as a source to exit software standby mode, the digital filter cannot be used. Set the IRQFLTE0.FLTENi bit, IRQFLTE1.FLTENi bit or NMIFLTE.NFLTEN bit to 0 before entering software standby mode. To enable the digital filter again, set the IRQFLTE0.FLTENi bit, IRQFLTE1.FLTENi bit, or NMIFLTE.NFLTEN bit to 1.
Figure 15.18 shows an example of digital filter operation.

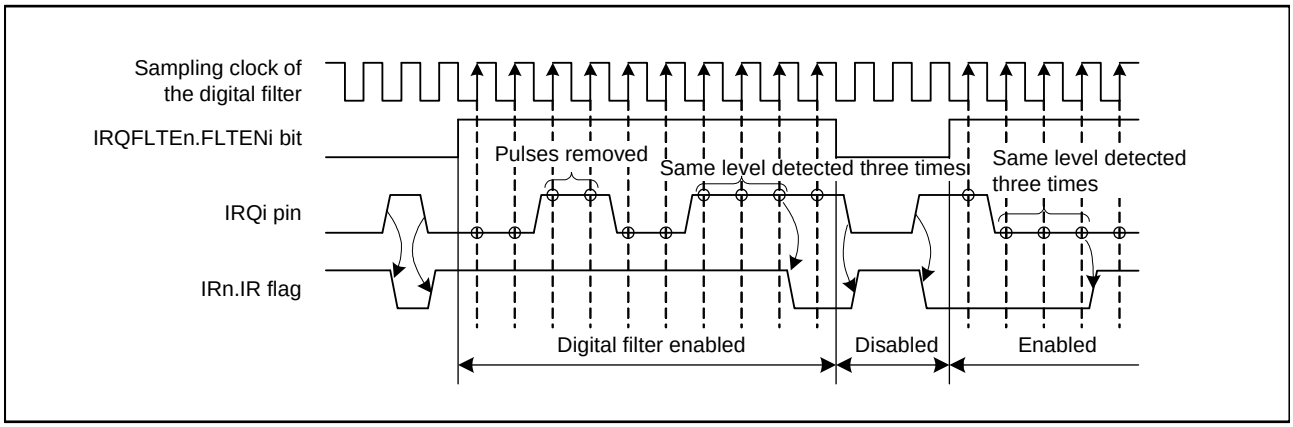


Figure 15.18 Digital Filter Operation Example (when the IRQCRi.IRQMD[1:0] bits are 00b (low level))

15.7.7 Setting Software Configurable Interrupts

The following describes the procedure to assign interrupt sources to software configurable interrupts.

(1) Set the IERm.IENj bit to 0 (j = 0 to 7; m = 02h to 1Fh). This setting is not required when the value does not change from the value after reset.
(2) For software configurable interrupt B, set the interrupt source number in registers SLIBXRn (n = 128 to 143) and SLIBRn (n = 144 to 207). Refer to Table 15.3, Interrupt Sources for Software Configurable Interrupt B for details on interrupt source numbers that are assigned to software configurable interrupt B.
(3) For software configurable interrupt A, set the interrupt source number in the SLIARn (n = 208 to 255) register. Refer to Table 15.4, Interrupt Sources for Software Configurable Interrupt A for details on interrupt source numbers that are assigned to software configurable interrupt A.
(4) When starting the EXDMAC by a software configurable interrupt, set each bit in the SELEXDR register.
(5) Set the SLIPRCR.WPRC bit to 1.
(6) Confirm that the SLIPRCR.WPRC bit is 1.
(7) Select the interrupt request destination from the CPU, DTC, or DMAC. Refer to section 15.7.3.1, Interrupt Request Destination Setting Procedure for details on the setting procedure.
(8) Write 0 to the IRn.IR flag only when edge detection is selected (n = 128 to 255).
(9) Set the IERm.IENj bit to 1.

15.7.7.1 Polling for Software Configurable Interrupts

When polling an interrupt request by reading the PIBRk.PIRj (k = 0h to Ah) or PIARk.PIRj (k = 0h to Bh), follow the procedure below (j = 0 to 7).

(1) Set the peripheral interrupt used.
(2) Clear the PIBRk.PIRj or PIARk.PIRj flag for polling by writing 1 to the flag.*1
(3) Enable output of the peripheral interrupt request.
(4) As needed, read the PIBRk.PIRj or PIARk.PIRj flag to check the value.
(5) When clearing the PIBRk.PIRj or PIARk.PIRj flag, write 1 to the targeted flag. *1
(6) As needed, repeat step (4) and (5).

Note 1. Do not use bit manipulation instructions. Multiple status flags may be cleared if a bit manipulation instruction is used. To clear a flag, write the PIBRk or PIARk register in bytes as follows: set the flag that is cleared to 1 and set the other flags to 0.

15.8 Multiple Interrupt

To enable another interrupt while processing an interrupt (multiple interrupt), set the PSW.I bit to 1 (interrupt enabled) in the interrupt handler of an accepted interrupt.
The PSW.IPL[3:0] bits in the interrupt handler are the same value as the priority level of the accepted interrupt request. In this case, when an interrupt request with the higher priority level than the PSW.IPL[3:0] bit value is generated, the interrupt request is accepted.
The PSW.I bit can be rewritten only in supervisor mode. Since the PSW.PM bit becomes 0 (supervisor mode is selected) when an interrupt is accepted, the PSW.I bit can be rewritten in the interrupt handler.

15.9 Fast interrupt

The fast interrupt is an interrupt that the CPU can respond to fast. Only one interrupt source can be assigned to the fast interrupt.
The priority level of the fast interrupt is 15 (highest) regardless of the IPRr.IPR[3:0] bit setting (r = 000 to 255). Also, the fast interrupt has higher priority than the other interrupt sources of which the priority level is 15. Note that the fast interrupt cannot be accepted when the PSW.IPL[3:0] bits are 1111b (priority level 15).
To assign an interrupt source to the fast interrupt, set the FIR.FVCT[7:0] bits to select the vector number of the interrupt source, and set the FIR.FIEN bit to 1 (fast interrupt is enabled).
The fast interrupt is enabled only when the CPU is selected as the interrupt request destination. When the DTC or DMAC is selected as the destination, the fast interrupt is disabled.
Refer to section 2, CPU and section 14, Exception Handling for details on the fast interrupt.

15.10 Exiting Low Power Consumption State

Interrupts can be used for exiting sleep mode, all-module clock stop mode, and software standby mode.
Refer to section 11, Low Power Consumption for details. This section describes the procedure to set an interrupt source for exiting each low power consumption mode.
Refer to section 11.6.4, Deep Software Standby Mode for details on exiting deep software standby mode.

15.10.1 Exiting Sleep Mode

Non-maskable interrupts and all interrupt sources can be used for exiting sleep mode. The following conditions must be satisfied.

(1) Non-maskable interrupt

- Generation of the interrupt that is used for exiting is enabled in the NMIER register.

(2) Interrupts

- The CPU is selected as the interrupt request destination.
- The interrupt request is enabled by the IERm.IENj bit (j = 0 to 7; m = 02h to 1Fh).
- The priority level is higher than the PSW.IPL[3:0] bit value in the CPU.
- For group interrupts, the interrupt request is enabled by the corresponding ENj bit in registers GRPBE0, GRPBL0, GRPBL1, GRPAL0, or GRPAL1 (j = 0 to 31).

15.10.2 Exiting All-Module Clock Stop Mode

Non-maskable interrupts and interrupt sources that have a "✓" in the Exit from ACS column in Table 15.5, Interrupt Vector Table can be used for exiting all-module clock stop mode. The conditions below must be satisfied.

(1) Non-maskable interrupt

- Generation of the interrupt that is used for exiting is enabled in the NMIER register.

(2) Interrupts

- The interrupt source can be used for exiting all-module clock stop mode.
- The CPU is selected as the interrupt request destination.
- The interrupt request is enabled by the IERm.IENj bit (j = 0 to 7; m = 02h to 1Fh).
- The priority level is higher than the PSW.IPL[3:0] bit value in the CPU.

15.10.3 Exiting Software Standby Mode

Non-maskable interrupts and interrupt sources that have a "✓" in the Exit from SSBY column in Table 15.5, Interrupt Vector Table can be used for exiting all-module clock stop mode. The conditions below must be satisfied.

(1) Non-maskable interrupt

- Generation of the interrupt that is used for exiting is enabled in the NMIER register.
- When using the NMI pin interrupt, the digital filter is disabled.

(2) Interrupts

- The interrupt source can be used for exiting software standby mode.
- The CPU is selected as the interrupt request destination.
- The interrupt request is enabled by the IERm.IENj bit (j = 0 to 7; m = 02h to 1Fh).
- The priority level is higher than the PSW.IPL[3:0] bit value in the CPU.
  When using the fast interrupt, set not only the FIR register but also the corresponding IPRr.IPR[3:0] bits (r = 000 to 255). Set the IPRr.IPR[3:0] bits to a higher level than the PSW.IPL[3:0] bit value in the CPU.
- When using the external pin interrupt, the digital filter of the IRQi pin used is disabled.

Refer to section 15.7.6, Digital Filter for details on the procedure to set the digital filter.

15.11 Usage Notes

15.11.1 Notes on the WAIT instruction When Using the Non-Maskable Interrupt

Confirm that all status flags in the NMISR register are 0 before executing the WAIT instruction.

15.11.2 Software Configurable Interrupts in All-Module Clock Stop Mode

When using an interrupt source assigned to a software configurable interrupt for exiting all-module clock stop mode, assign the interrupt source to software configurable interrupt B (INTB146 to INTB157) of interrupt vector numbers 146 to 157.

15.11.3 Interrupt Requests in Software Standby Mode

When an interrupt request occurs in software standby mode but the interrupt source is not set as a source for exiting software standby mode, the request is held in the ICU. The request is handled after exiting by another interrupt source. Note that the interrupt request for the external pin interrupt is not held.

16. Buses

16.1 Overview

Table 16.1 lists the bus specifications, Figure 16.1 shows the bus configuration, and Table 16.2 lists the addresses assigned for each bus.

Table 16.1 Bus Specifications

Bus Type		Description
CPU bus	Instruction bus	• Connected to the CPU (for instructions) • Connected to on-chip memory (RAM, code flash memory) • Operates in synchronization with the system clock (ICLK)
	Operand bus	• Connected to the CPU (for operands) • Connected to on-chip memory (RAM, code flash memory) • Operates in synchronization with the system clock (ICLK)
Memory bus	Memory bus 1	• Connected to RAM
	Memory bus 2	• Connected to code flash memory
	Memory bus 3	• Connected to ECCRAM
Internal main bus	Internal main bus 1	• Connected to the CPU • Operates in synchronization with the system clock (ICLK)
	Internal main bus 2	• Connected to the DMAC, DTC, and EDMAC • Connected to on-chip memory (RAM, code flash memory) • Operates in synchronization with the system clock (ICLK)
Internal peripheral bus	Internal peripheral bus 1	• Connected to peripheral modules (DTC, DMAC, EXDMAC, interrupt controller, and bus error monitoring section) • Operates in synchronization with the system clock (ICLK) (EDMAC operates in synchronization with the BCLK)
	Internal peripheral bus 2	• Connected to peripheral modules (modules other than those connected to internal peripheral buses 1, 3, 4, and 5) • Operates in synchronization with the peripheral-module clock (PCLKB)
	Internal peripheral bus 3	• Connected to peripheral modules (USBb, PDC, and standby RAM) • Operates in synchronization with the peripheral-module clock (PCLKB)
	Internal peripheral bus 4	• Connected to peripheral modules (EDMAC, ETHERC, EPTPC, MTU3, GPT, SCIF, RSPI, USBA, and AES) • Operates in synchronization with the peripheral-module clock (PCLKA)
	Internal peripheral bus 5	• Reserved area
	Internal peripheral bus 6	• Connected to code flash (in P/E) and data flash memory • Operates in synchronization with the FlashIF clock (FCLK)
External bus	CS area	• Connected to the external devices • Operates in synchronization with the external-bus clock (BCLK)
	SDRAM area	• Connected to the SDRAM • Operates in synchronization with the SDRAM clock (SDCLK)

P/E: Programming/Erasure
BCLK (external-bus clock): 120 MHz (max.) The CSC (CS area controller) and the EXDMAC operate in synchronization with the BCLK.
SDCLK (SDRAM clock): 60 MHz (max.) The SDRAMC (SDRAM area controller) operates in synchronization with the SDCLK.
BCLK pin output: The frequency is the same as the BCLK as default. 1/2 BCLK can be supplied by setting the BCLK pin output select bit (BCKCR.BCLKDIV) in the external bus clock control register. For details, see section 9, Clock Generation Circuit.
Note: The BCLK and the SDCLK should be operated with the same frequency when the SDRAM is in use.

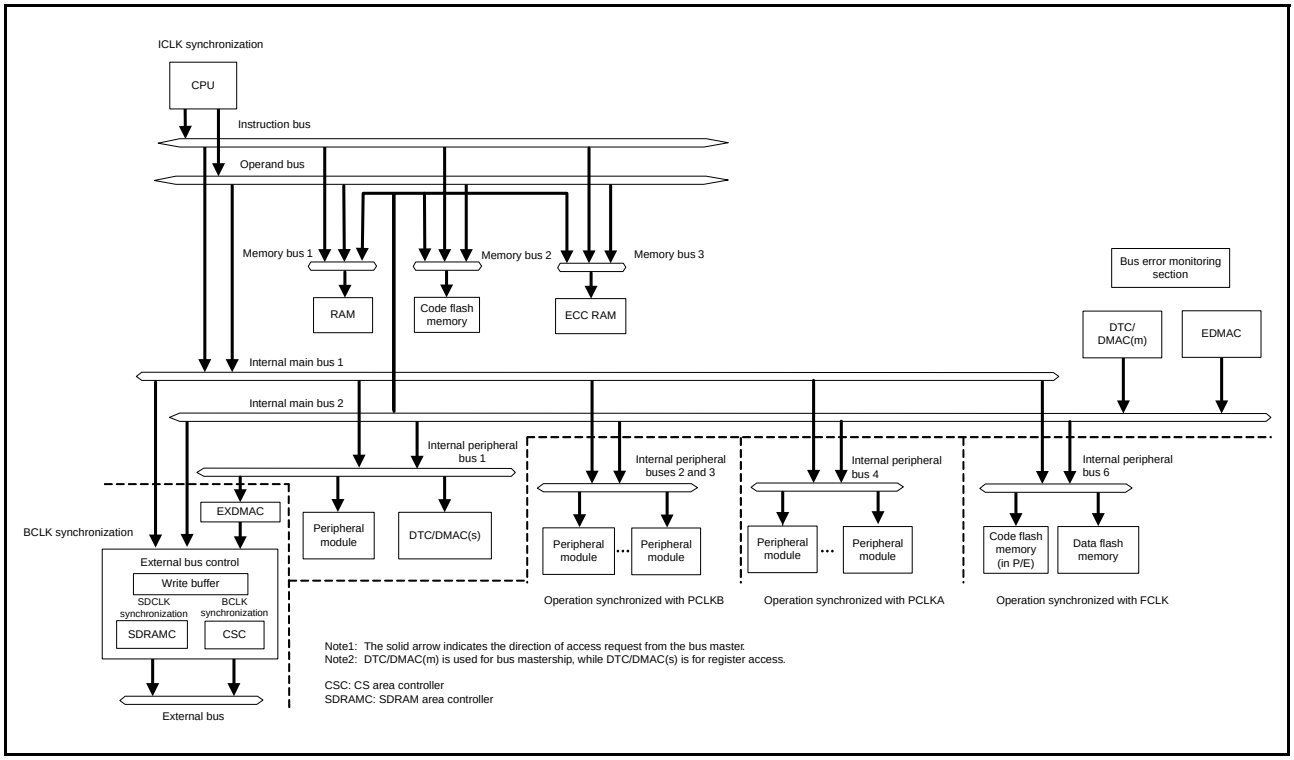


Figure 16.1 Bus Configuration

Table 16.2 Addresses Assigned for Each Bus

Address	Bus		Area	
	On-Chip ROM Enabled	On-Chip ROM Disabled	On-Chip ROM Enabled	On-Chip ROM Disabled
0000 0000h to 0007 FFFFh	Memory bus 1		RAM	
0008 0000h to 0008 7FFFh	Internal peripheral bus 1		Peripheral I/O registers	
0008 8000h to 0009 FFFFh	Internal peripheral bus 2			
000A 0000h to 000B FFFFh	Internal peripheral bus 3			
000C 0000h to 000D FFFFh	Internal peripheral bus 4			
000E 0000h to 000F FFFFh	Reserved area			
0010 0000h to 00FF FFFFh	Internal peripheral bus 6	Reserved area	Data flash memory, FCURAM, code flash memory (for programming only)	Reserved area
0080 0000h to 00FF 7FFFh	Memory bus 3		Reserved area	
00FF 8000h to 00FF FFFFh			ECCRAM	
0100 0000h to 07FF FFFFh	External bus		External address space (CS1 to CS7)	
0800 0000h to 0FFF FFFFh			SDRAM area	
1000 0000h to 7FFF FFFFh	Reserved area		Reserved area	
8000 0000h to FEFF FFFFh	Memory bus 2	Reserved area	Code flash memory (for reading only)	Reserved area
FF00 0000h to FFFF FFFFh		External bus		External address space (CS0)

16.2 Description of Buses

16.2.1 CPU Buses

The CPU buses consist of the instruction and operand buses, which are connected to internal main bus 1. As the names suggest, the instruction bus is used to fetch instructions for the CPU, while the operand bus is used for operand access. The instruction bus is 64 bits while the operand bus is 32 bits.
Connection of the instruction and operand buses to RAM and code flash memory provides the CPU with direct access to these areas, i.e. access is not via internal main bus 1. However, only reading is possible in direct access to code flash memory by the CPU; programming and erasure are handled via an internal peripheral bus.
Bus requests for instruction fetching and operand access are arbitrated through internal main bus 1. The order of priority is operand access then instruction fetching.
If instruction fetching and operand access are requested for different buses (memory bus 1, memory bus 2, memory bus 3, and internal main bus 1), the bus-access operations can proceed simultaneously. For example, parallel access to code flash memory and RAM or to code flash memory and external space is possible.

16.2.2 Memory Buses

The memory buses consist of memory bus 1, memory bus 2, and memory bus 3. The RAM is connected to memory bus 1, code flash memory is connected to memory bus 2, and ECCRAM is connected to memory bus 3. The memory buses are 64 bits. Requests for bus mastership from the CPU buses (instruction fetching and operand) and internal main bus 2 are arbitrated through memory buses 1, 2, and 3.
The priority order of the buses can be set using the memory bus 1 (RAM, ECCRAM) priority control bits (BPRA[1:0]) and memory bus 2 (code flash memory) priority control bits (BPRO[1:0]) in the bus priority control register (BUSPRI) for the corresponding memory buses. Memory bus 3 has the same configuration as memory bus 1. When the priority order is fixed, internal main bus 2 has priority over the CPU bus (operand over instruction fetching). When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

16.2.3 Internal Main Buses

The internal main buses consist of a bus for use by the CPU (internal main bus 1) and a bus for use by the other bus-master modules, i.e. the DTC, DMAC, and EDMAC (internal main bus 2).
Bus requests for instruction fetching and operand access are arbitrated through internal main bus 1. The order of priority is operand access then instruction fetching.
Requests for bus mastership from the DTC, DMAC, and EDMAC are arbitrated by internal main bus 2. The order of priority is EDMAC, DMAC, and then DTC as listed in Table 16.3.
Between the DTC and DMAC, only the one that accepted the transfer request issues the bus mastership request. The priority order of transfer requests between the DTC and DMAC is DMAC0, DMAC1, DMAC2, DMAC3, and then DTC, regardless of the BUSPRI setting.
If the CPU and another bus master are requesting access to different buses (on-chip memory, internal peripheral buses 1 to 6, and external bus), the respective bus-access operations can proceed simultaneously.
However, when the CPU executes the XCHG instruction, requests for bus access from masters other than the CPU are not accepted until data transfer for the XCHG instruction is completed regardless of the bus priority control register (BUSPRI) setting. Furthermore, requests for bus access from masters other than the DTC are not accepted during reading and writing-back of transfer control information for the DTC.

Table 16.3 Order of Priority for Bus Masters

Priority	Internal main bus	Bus Master
High	—	EXDMAC
▲	2	EDMAC
		DMAC
		DTC
Low	1	CPU

Note: The above applies when the priority order of the buses is fixed.
The priority order of the internal main bus 1 and other buses (internal main bus 2 and EXDMAC) can be toggled by using the bus priority control register (round-robin method).
However, the priority order of EXDMAC to be connected only to the external bus has priority over internal main bus 2 regardless of the setting of the bus priority control register (BUSPRI) (EXDMAC > internal main bus 2).

16.2.4 Internal Peripheral Buses

Connection of peripheral modules to the internal peripheral buses is as described in Table 16.4.

Table 16.4 Connection of Peripheral Modules to the Internal Peripheral Buses

Type of Bus	Peripheral Modules
Internal peripheral bus 1	DTC, DMAC, EXDMAC, interrupt controller, and bus error monitoring section
Internal peripheral bus 2	Peripheral modules other than those connected to internal peripheral buses 1, 3, 4, and 5
Internal peripheral bus 3	USBb, PDC, and standby RAM
Internal peripheral bus 4	EDMAC, ETHERC, EPTPC, MTU3, GPT, SCIF, RSPI, USBA, and AES
Internal peripheral bus 5	Reserved area
Internal peripheral bus 6	Code flash memory (in P/E) or data flash memory, and FCU RAM

Requests for bus mastership from the CPU (internal main bus 1) and other bus masters (internal main bus 2) are arbitrated through internal peripheral buses 1 to 6.
The priority order of two internal main buses can be set using the bus priority control register (BUSPRI). The priority order can be set with the internal peripheral bus 1 priority control bits (BUSPRI.BPIB[1:0]), internal peripheral bus 2 and 3 priority control bits (BUSPRI.BPGB[1:0]), internal peripheral bus 4 and 5 priority control bits (BUSPRI.BPHB[1:0]), and internal peripheral bus 6 priority control bits (BUSPRI.BPFB[1:0]) for the corresponding internal peripheral buses.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1. When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted (round-robin method).
The order of accepting requests may change depending on the BUSPRI setting (Refer to Figure 16.2).

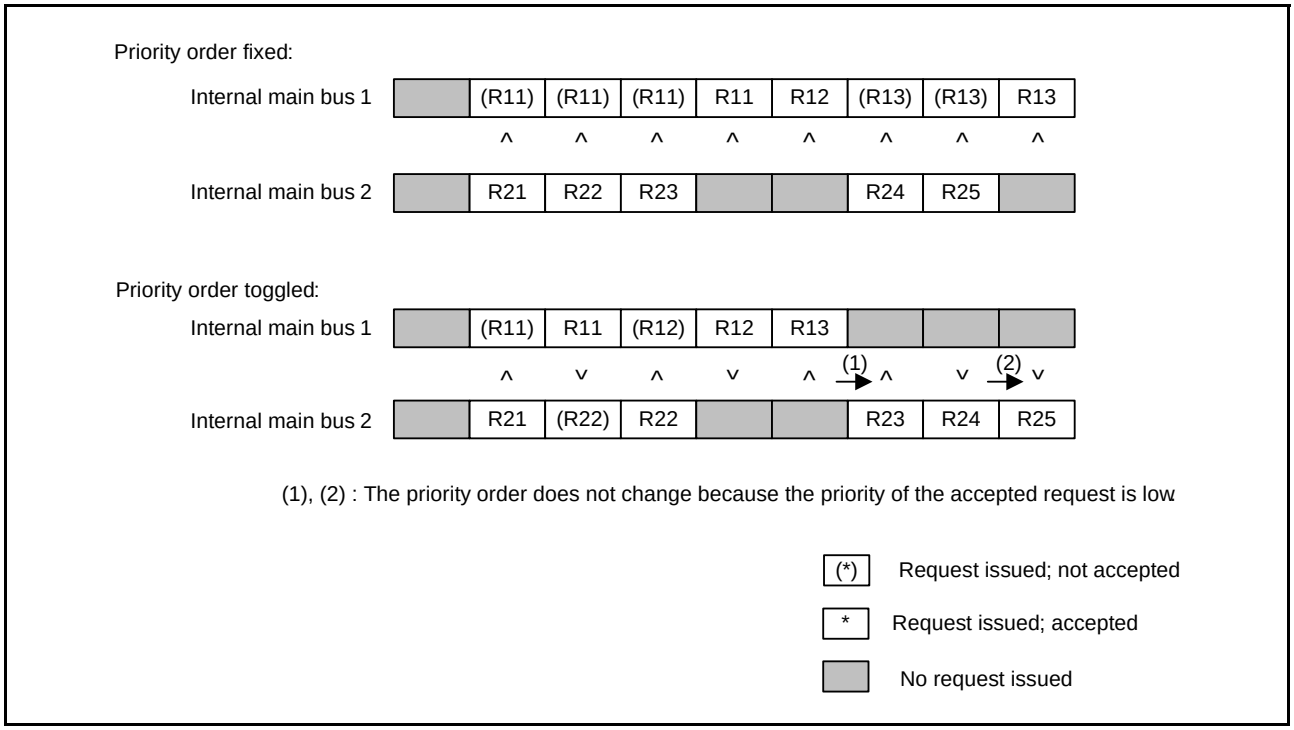


Figure 16.2 Priority Order between Internal Peripheral Bus Accesses

16.2.5 Write Buffer Function (Internal Peripheral Bus)

The internal peripheral bus has the write buffer function, which allows the next round of bus access to start, before the current write access is completed, in write access. However, if the following round of bus access is from the same bus master but to the different internal peripheral bus, it is suspended until the bus operations already in progress are completed. When access to the internal memory is scheduled after the write access to the internal peripheral bus from the CPU, the following round of bus access can be started before the current bus operation is completed and thus the order of accesses may be changed (Refer to Figure 16.3).

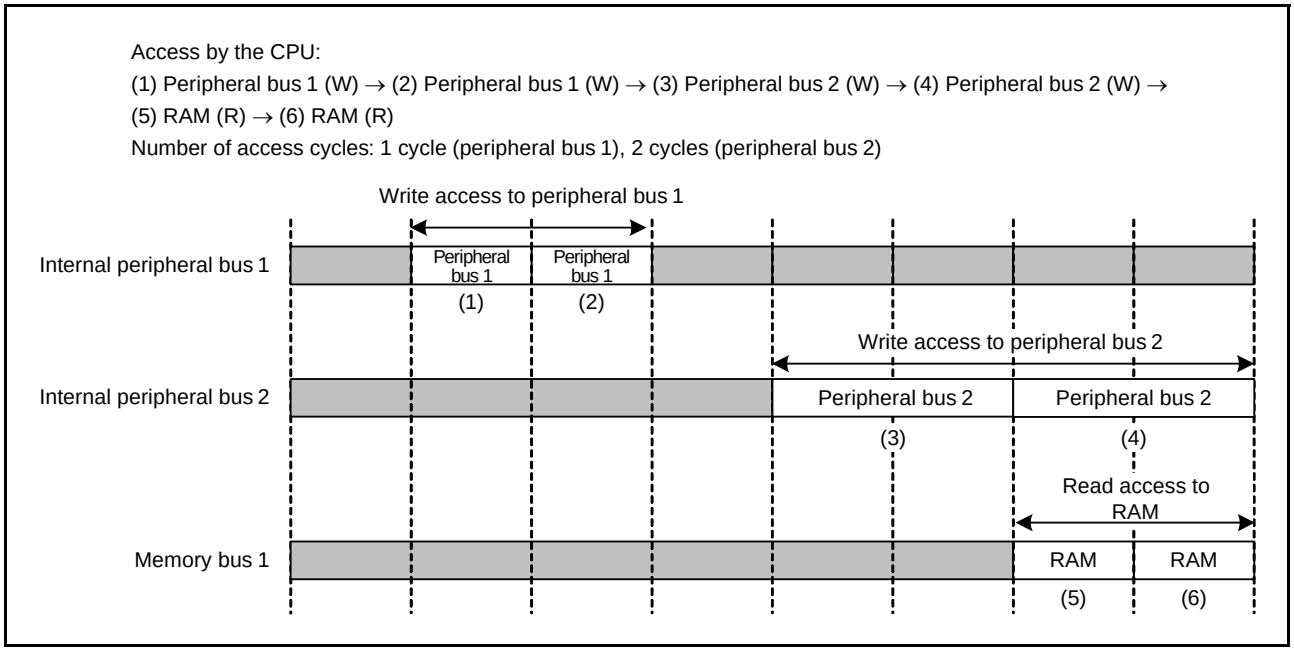


Figure 16.3 Write Buffer Function

16.2.6 External Bus

Table 16.5 lists the specifications of the external bus.

The external bus controller arbitrates requests for bus mastership on the external address space and external bus controller registers (CSC and SDRAMC) from internal main bus 1, internal main bus 2, and EXDMAC. However, the external address space is only accessible from the EXDMAC.

The priority order of these three buses can be set using the external bus priority control bits (BPEB[1:0]) in the bus priority control register (BUSPRI). When the priority order is EXDMAC, fixed, the order is internal main bus 2, and then internal main bus 1. When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted, between internal main bus 1 and the other buses (internal main bus 2 and EXDMAC). However, the order of priority is EXDMAC and then internal main bus 2, regardless of the external bus priority control bits settings.

The order of accepting requests may change depending on the BUSPRI setting (Refer to Figure 16.4).

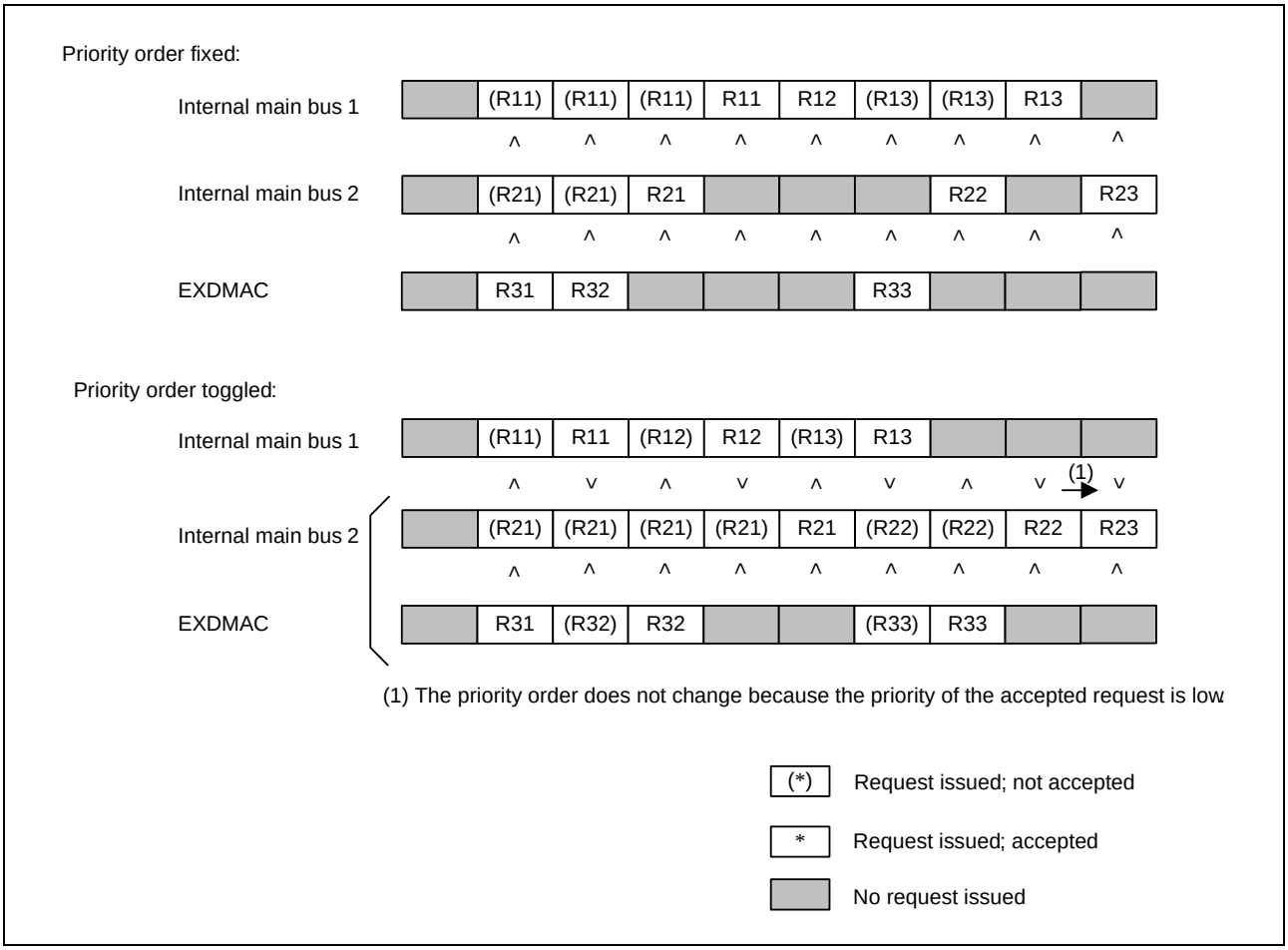


Figure 16.4 Priority Order of Internal Peripheral Bus Accesses

Table 16.5 Specifications of the External Bus

Item	Description
External address space	• An external address space is divided into eight CS areas (CS0 to CS7) and the SDRAM area (SDCS) for management. • Chip select signals can be output for each area. • Bus width can be set for each area. Separate bus: An 8, 16, or 32-bit bus space is selectable. Address/data multiplexed bus: An 8 or 16-bit bus space is selectable. • An endian mode can be specified for each area.
CS area controller	• Recovery cycles can be inserted. Read recovery: Up to 15 cycles Write recovery: Up to 15 cycles • Cycle wait function: Wait for up to 31 cycles (page access: up to 7 cycles) • Wait control can be used to set up the following. Timing of assertion and negation for chip-select signals (CS0# to CS7#) The timing of assertion of the read signal (RD#) and write signals (WR0#/WR#, and WR1# to WR3#) The timing with which data output starts and ends • Write access mode: Single write strobe mode/byte strobe mode • Separate bus or address/data multiplexed bus can be set for each area.
SDRAM area controller	• Multiplexing output of row address/column address (8, 9, 10, or 11 bits) • Self-refresh and auto-Refresh selectable • CAS latency can be specified from one to three cycles
Write buffer function	When write data from the bus master has been written to the write buffer, write access by the bus master is completed.
Frequency	• The CS area controller (CSC) operates in synchronization with the external-bus clock (BCLK).*1 • The SDRAM area controller (SDRAMC) operates in synchronization with the SDRAM clock (SDCLK).

Note 1. The BCLK and the SDCLK should be operated with the same frequency when the SDRAM is in use.

Table 16.6 lists the input/output pins of the external bus.

Table 16.6 Pin Configuration of the External Bus (1/2)

Pin Name	I/O	Description
A23 to A0*1	Output	Address output pins
D31 to D0	I/O	Data input/output pins D31 to D0 pins are enabled when the 32-bit bus space is specified. D15 to D0 pins are enabled when the 16-bit bus space is specified. D7 to D0 pins are enabled when the 8-bit bus space is specified.
BC0#*1	Output	A strobe signal; (the BC0# signal being at the low level) during access to an external address space in single write strobe mode indicates that D7 to D0 are valid. When an 8-bit bus space is specified, this output pin is always held low regardless of write access mode.
BC1#	Output	A strobe signal; (the BC1# signal being at the low level) during access to an external address space in single write strobe mode indicates that D15 to D8 are valid. This pin is not used when the 8-bit bus space is specified.
BC2#	Output	A strobe signal; (the BC2# signal being at the low level) during access to an external address space in single write strobe mode indicates that D23 to D16 are valid. This pin is not used when the 8- or 16-bit bus space is specified.
BC3#	Output	A strobe signal; (the BC3# signal being at the low level) during access to an external address space in single write strobe mode indicates that D31 to D24 are valid. This pin is not used when the 8- or 16-bit bus space is specified.
CS0#	Output	A chip select signal for area 0 (CS0)
CS1#	Output	A chip select signal for area 1 (CS1)
CS2#	Output	A chip select signal for area 2 (CS2)
CS3#	Output	A chip select signal for area 3 (CS3)
CS4#	Output	A chip select signal for area 4 (CS4)
CS5#	Output	A chip select signal for area 5 (CS5)
CS6#	Output	A chip select signal for area 6 (CS6)

Table 16.6 Pin Configuration of the External Bus (2/2)

Pin Name	I/O	Description
CS7#	Output	A chip select signal for area 7 (CS7)
RD#	Output	A strobe signal indicating that reading from an external address space (CS0 to CS7) is in progress
WR0#/WR#*2	Output	WR0# signal is a strobe signal indicates that (the WR0# signal being at the low level) writing to an external address space is in progress in byte strobe mode, and D7 to D0 are valid. WR# signal is a strobe signal that indicates writing to an external address space is in progress in single write strobe mode. When an 8-bit bus space is specified, this output pin is held low during a write access regardless of write access mode.
WR1#	Output	A strobe signal; (the WR1# signal being at the low level) during writing to an external address space in byte strobe mode indicates that D15 to D8 are valid. This signal is invalid in single write strobe mode. This pin is not used when the 8-bit bus space is specified.
WR2#	Output	A strobe signal; (the WR2# signal being at the low level) during writing to an external address space in byte strobe mode indicates that D23 to D16 are valid. This signal is invalid in single write strobe mode. This pin is not used when the 8- or 16-bit bus space is specified.
WR3#	Output	A strobe signal; (the WR3# signal being at the low level) during writing to an external address space in byte strobe mode indicates that D31 to D24 are valid. This signal is invalid in single write strobe mode. This pin is not used when the 8- or 16-bit bus space is specified.
ALE	Output	Address latch signal when address/data multiplexed bus is selected.
WAIT#	Input	A wait request signal when accessing the external address space (CS0 to CS7) (Low: Wait request)
SDCLK	Output	SDRAM clock
CKE	Output	SDRAM clock enable signal
SDCS#	Output	SDRAM chip select signal
RAS#	Output	SDRAM low address strobe signal
CAS#	Output	SDRAM column address strobe signal
WE#	Output	SDRAM write enable signal
DQM0	Output	SDRAM I/O data mask enable signal for D7 to D0
DQM1	Output	SDRAM I/O data mask enable signal for D15 to D8
DQM2	Output	SDRAM I/O data mask enable signal for D23 to D16
DQM3	Output	SDRAM I/O data mask enable signal for D31 to D24

Note 1. The A0 and BC0# pin functions share the same pin, and either becomes effective according to the area, with the function being A0 in byte strobe mode and BC0# in single write strobe mode. Note that setting the 8-bit external bus width is prohibited in single write strobe mode. For information on other multiplexed pin functions, see section 22, I/O Ports.

Note 2. The WR0# signal and WR# signal are identical. The WR0# signal is particularly referred to as WR# in single write strobe mode.

16.2.7 Parallel Operation

Parallel operation is possible when different bus-master modules are requesting access to different slave modules. For example, if the CPU is fetching an instruction from code flash memory and an operand from RAM, the DMAC is able to handle transfer between a peripheral bus and the external bus at the same time.
An example of parallel operations is shown in Figure 16.5. In this example, the CPU is able to employ the instruction and operand buses for simultaneous access to code flash memory and RAM, respectively. Furthermore, the DMAC simultaneously employs internal main bus 2 for access to a peripheral bus or the external bus during access to RAM and code flash memory by the CPU.

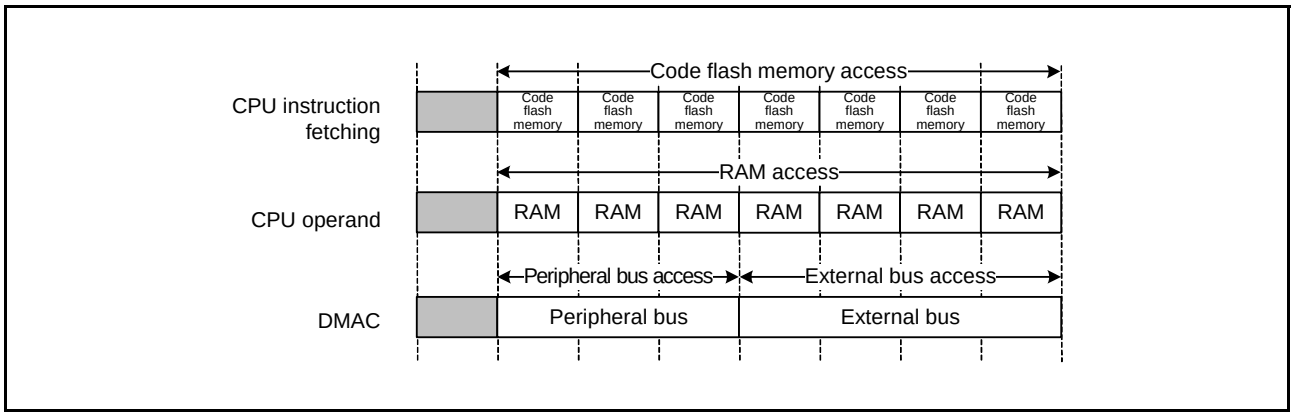


Figure 16.5 Example of Parallel Operations

16.2.8 Bus Settings

(1) Set the mode of the external bus in the CSn mode register (CSnMOD), CSn wait-control register 1 (CSnWCR1), CSn wait-control register 2 (CSnWCR2), CSn control register (CSnCR), CSn recovery-cycle setting register (CSnREC), CS recovery cycle insertion enable register (CSRECEN), bus error monitoring enable register (BEREN), and bus priority control register (BUSPRI).
(2) Make settings for pins in the CS output enable register (PFCSE), CS output pin select register 0 (PFCSS0), CS output pin select register 1 (PFCSS1), address output enable register 0 (PFAOE0), address output enable register 1 (PFAOE1), external-bus control register 0 (PFBCR0), and external-bus control register 1 (PFBCR1).
(3) Set up pins to be used as input port pins.
(4) Set the external-bus enable bit (EXBE) in the system control register 0 (SYSCR0) to 1 (enabling the external bus).

16.2.9 Restrictions

(1) Prohibition of Access that Spans Areas of Address Space

Single access that spans two areas of the address space is prohibited, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single word or longword access.
Single access that spans two areas of the address space is also prohibited in the EXDMAC block transfer in single address mode or cluster transfer, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single access in the EXDMAC block transfer in single address mode or cluster transfer.

(2) Restrictions in Relation to RMPA and String-Manipulation Instructions

(a) Although the external space has a per-area ending-switching facility (only for data), the allocation of data to be handled by RMPA or string-manipulation instructions to an area where the endian differs from that of the chip is prohibited, and operation is not guaranteed if this restriction is not observed. If data to be handled by RMPA or string-manipulation instructions are allocated to the external space, they must be allocated to areas where the endian setting is the same as that for the chip.

(b) The allocation of data to be handled by RMPA or string-manipulation instructions to I/O registers is prohibited, and operation is not guaranteed if this restriction is not observed.

(3) Restriction on Endian

When the endian setting for each area is different from that for the chip, no instruction code can be allocated in the area. The instruction code should be allocated to the external space whose endian setting is the same as that for the chip.

16.3 Register Descriptions

16.3.1 CSn Control Register (CSnCR) (n = 0 to 7)

Address(es): CS0CR 0008 3802h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MPXEN	—	—	—	EMODE	—	—	BSIZE[1:0]		—	—	—	EXENB
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1

Address(es): CS1CR 0008 3812h, CS2CR 0008 3822h, CS3CR 0008 3832h, CS4CR 0008 3842h, CS5CR 0008 3852h, CS6CR 0008 3862h, CS7CR 0008 3872h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MPXEN	—	—	—	EMODE	—	—	BSIZE[1:0]		—	—	—	EXENB
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EXENB	Operation Enable	0: Operation is disabled 1: Operation is enabled	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	BSIZE[1:0]	External Bus Width Select	b5 b4 0 0: A 16-bit bus space is selected 0 1: A 32-bit bus space is selected 1 0: An 8-bit bus space is selected 1 1: Setting prohibited	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	EMODE	Endian Mode	0: Endian of area n is the same as the endian of operating mode. 1: Endian of area n is not the endian of operating mode. (n = 0 to 7)	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	MPXEN	Address/Data Multiplexed I/O Interface Select	0: Separate bus interface is selected for area n. 1: Address/data multiplexed I/O interface is selected for area n. (n = 0 to 7)	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Do not attempt to write the CSnCR register while the external bus is being accessed.

EXENB Bit (Operation Enable)

This bit enables or disables operation of the respective CS areas.

After this LSI is reset, operation is enabled (EXENB = 1) only for area 0; operation in other areas is disabled (EXENB = 0).

An attempt at access to an area for which operation has been disabled does not lead to access via the external bus. However, if the illegal address access detection enable bit in the bus error monitoring enable register has been set to enable detection (BEREN.IGAEN = 1), such an attempt will lead to an illegal-access error.

BSIZE[1:0] Bits (External Bus Width Select)

These bits specify the data bus width of each area.

The data bus width of area 0 (CS0) after a reset depends on the setting of the bus width in operating mode.

When the address/data multiplexed I/O interface is selected with the MPXEN bit, the BSIZE[1:0] bits should not be set to the 32-bit bus space. If set, the operation cannot be guaranteed.

EMODE Bit (Endian Mode)

This bit specifies the endian of each area.

When the endian setting for each area is different from that for the chip, no instruction code can be allocated in the area. The instruction code should be allocated to the external space whose endian setting is the same as that for the chip.

16.3.2 CSn Recovery Cycle Register (CSnREC) (n = 0 to 7)

Address(es): CS0REC 0008 380Ah, CS1REC 0008 381Ah, CS2REC 0008 382Ah, CS3REC 0008 383Ah, CS4REC 0008 384Ah, CS5REC 0008 385Ah, CS6REC 0008 386Ah, CS7REC 0008 387Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	WRCV[3:0]				—	—	—	—	RRCV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	RRCV[3:0]	Read Recovery	b3 b0 0 0 0 0: No recovery cycle is inserted. 0 0 0 1: 1 recovery cycle is inserted. 0 0 1 0: 2 recovery cycles are inserted. 0 0 1 1: 3 recovery cycles are inserted. 0 1 0 0: 4 recovery cycles are inserted. 0 1 0 1: 5 recovery cycles are inserted. 0 1 1 0: 6 recovery cycles are inserted. 0 1 1 1: 7 recovery cycles are inserted. 1 0 0 0: 8 recovery cycles are inserted. 1 0 0 1: 9 recovery cycles are inserted. 1 0 1 0: 10 recovery cycles are inserted. 1 0 1 1: 11 recovery cycles are inserted. 1 1 0 0: 12 recovery cycles are inserted. 1 1 0 1: 13 recovery cycles are inserted. 1 1 1 0: 14 recovery cycles are inserted. 1 1 1 1: 15 recovery cycles are inserted.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	WRCV[3:0]	Write Recovery	b11 b8 0 0 0 0: No recovery cycle is inserted. 0 0 0 1: 1 recovery cycle is inserted. 0 0 1 0: 2 recovery cycles are inserted. 0 0 1 1: 3 recovery cycles are inserted. 0 1 0 0: 4 recovery cycles are inserted. 0 1 0 1: 5 recovery cycles are inserted. 0 1 1 0: 6 recovery cycles are inserted. 0 1 1 1: 7 recovery cycles are inserted. 1 0 0 0: 8 recovery cycles are inserted. 1 0 0 1: 9 recovery cycles are inserted. 1 0 1 0: 10 recovery cycles are inserted. 1 0 1 1: 11 recovery cycles are inserted. 1 1 0 0: 12 recovery cycles are inserted. 1 1 0 1: 13 recovery cycles are inserted. 1 1 1 0: 14 recovery cycles are inserted. 1 1 1 1: 15 recovery cycles are inserted.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Do not attempt to write the CSnREC register while the external bus is being accessed.

When the preceding bus access is a separate bus access, CSnREC is valid when the recovery cycle insertion is enabled with the separate bus recovery cycle insertion enable bit (RCVENj (j = 0 to 7)) in CSRECEN. When the preceding bus access is an address/data multiplexed bus access, CSnREC is valid when the recovery cycle insertion is enabled with the multiplexed bus recovery cycle insertion enable bit (RCVENMj) in CSRECEN.

RRCV[3:0] Bits (Read Recovery)

These bits specify the number of recovery cycles to be inserted after a read access to the external bus.
When the recovery cycle insertion is enabled and a value except 0000b is written to these bits, one to 15 recovery cycles are inserted in the following cases.

- After a read access to the external bus, a read access is made to the external bus in the same area.
- After a read access to the external bus, a read access is made to the external bus in a different area.
- After a read access to the external bus, a write access is made to the external bus in the same area.
- After a read access to the external bus, a write access is made to the external bus in a different area.

WRCV[3:0] Bits (Write Recovery)

These bits specify the number of recovery cycles to be inserted after a write access to the external bus.
When the recovery cycle insertion is enabled and a value except 0000b is written to these bits, one to 15 recovery cycles are inserted in the following cases.

- After a write access to the external bus, a read access is made to the external bus in the same area.
- After a write access to the external bus, a read access is made to the external bus in a different area.
- After a write access to the external bus, a write access is made to the external bus in the same area.
- After a write access to the external bus, a write access is made to the external bus in a different area.

16.3.3 CS Recovery Cycle Insertion Enable Register (CSRECEN)

Address(es): 0008 3880h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCVENM7	RCVENM6	RCVENM5	RCVENM4	RCVENM3	RCVENM2	RCVENM1	RCVENM0	RCVEN7	RCVEN6	RCVEN5	RCVEN4	RCVEN3	RCVEN2	RCVEN1	RCVEN0
Value after reset:	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	RCVEN0	Separate Bus Recovery Cycle Insertion Enable 0	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b1	RCVEN1	Separate Bus Recovery Cycle Insertion Enable 1	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b2	RCVEN2	Separate Bus Recovery Cycle Insertion Enable 2	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b3	RCVEN3	Separate Bus Recovery Cycle Insertion Enable 3	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b4	RCVEN4	Separate Bus Recovery Cycle Insertion Enable 4	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b5	RCVEN5	Separate Bus Recovery Cycle Insertion Enable 5	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b6	RCVEN6	Separate Bus Recovery Cycle Insertion Enable 6	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b7	RCVEN7	Separate Bus Recovery Cycle Insertion Enable 7	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b8	RCVENM0	Multiplexed Bus Recovery Cycle Insertion Enable 0	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b9	RCVENM1	Multiplexed Bus Recovery Cycle Insertion Enable 1	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b10	RCVENM2	Multiplexed Bus Recovery Cycle Insertion Enable 2	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b11	RCVENM3	Multiplexed Bus Recovery Cycle Insertion Enable 3	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b12	RCVENM4	Multiplexed Bus Recovery Cycle Insertion Enable 4	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b13	RCVENM5	Multiplexed Bus Recovery Cycle Insertion Enable 5	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b14	RCVENM6	Multiplexed Bus Recovery Cycle Insertion Enable 6	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b15	RCVENM7	Multiplexed Bus Recovery Cycle Insertion Enable 7	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W

Do not attempt to write the CSRECEN register while the external bus is being accessed.

RCVENn Bit (Separate Bus Recovery Cycle Insertion Enable n) (n = 0 to 7)

This bit enables or disables the insertion of read recovery cycles when, after a read access to the external bus, a read access is made to the external bus in the same area.

RCVENMn Bit (Multiplexed Bus Recovery Cycle Insertion Enable n) (n = 0 to 7)

This bit enables or disables the insertion of read recovery cycles when, after a read access to the external bus, a read access is made to the external bus in the same area.

Table 16.7 Insertion of Recovery Cycles

Access Type	External Address Space	Insertion of Recovery Cycles	Corresponding Bits (Separate/Multiplexed)
Read access after read access	Same area	Recovery cycles specified by the RRCV[3:0] bits are inserted.	RCVEN0/RCVENM0
	Different area	Recovery cycles specified by the RRCV[3:0] bits are inserted.	RCVEN1/RCVENM1
Read access after write access	Same area	Recovery cycles specified by the RRCV[3:0] bits are inserted.	RCVEN2/RCVENM2
	Different area	Recovery cycles specified by the RRCV[3:0] bits are inserted.	RCVEN3/RCVENM3
Write access after read access	Same area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN4/RCVENM4
	Different area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN5/RCVENM5
Write access after write access	Same area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN6/RCVENM6
	Different area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN7/RCVENM7

16.3.4 CSn Mode Register (CSnMOD) (n = 0 to 7)

Address(es): CS0MOD 0008 3002h, CS1MOD 0008 3012h, CS2MOD 0008 3022h, CS3MOD 0008 3032h, CS4MOD 0008 3042h, CS5MOD 0008 3052h, CS6MOD 0008 3062h, CS7MOD 0008 3072h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRMOD	—	—	—	—	—	PWENB	PRENB	—	—	—	—	EWENB	—	—	WRMOD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WRMOD	Write Access Mode Select	0: Byte strobe mode 1: Single write strobe mode	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	EWENB	External Wait Enable	0: External wait is disabled 1: External wait is enabled	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PRENB	Page Read Access Enable	0: Page read access is disabled 1: Page read access is enabled	R/W
b9	PWENB	Page Write Access Enable	0: Page write access is disabled 1: Page write access is enabled	R/W
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	PRMOD	Page Read Access Mode Select	0: Normal access compatible mode 1: External data read continuous assertion mode	R/W

Do not write to the CSnMOD register while access to the CSn area is in progress.

WRMOD Bit (Write Access Mode Select)

This bit selects a write access operating mode.

Writing 0 to this bit selects byte strobe mode where data write operation is controlled by the WRn# (n = 0 to 3) signal corresponding to the respective byte positions.

Writing 1 to this bit selects single write strobe mode where data write operation is controlled by the BCn# (n = 0 to 3) signal and the WR# signal corresponding to respective byte positions. Note that setting the external bus width of 8 bits is prohibited in single write strobe mode.

Table 16.8 Control Signals for Write Access Mode

Mode	Pin Name							
Write Access Mode	WR3#	WR2#	WR1#	WR0#/ WR#	BC3#	BC2#	BC1#	BC0#
Byte strobe mode	✓	✓	✓	✓ (WR0#)	×	×	×	×
Single write strobe mode	×	×	×	✓ (WR#)	✓	✓	✓	✓

✓: Enabled, ×: Disabled

EWENB Bit (External Wait Enable)

This bit enables or disables external wait.

Writing 1 to this bit selects external wait and allows control of the number of waits in each cycle with the WAIT# signal. In this state, wait cycles are inserted while the WAIT# signal is at the low level.

Writing 0 to this bit disables the WAIT# signal.

PRENB Bit (Page Read Access Enable)

This bit enables or disables page read accesses.

Note: When the address/data multiplexed I/O interface is selected with the MPXEN bit in CSnCR, this bit should not be set to enable page read accesses. Page read accesses are not supported in the address/data multiplexed I/O interface.

PWENB Bit (Page Write Access Enable)

This bit enables or disables page write accesses.

Note: When the address/data multiplexed I/O interface is selected with the MPXEN bit in CSnCR, this bit should not be set to enable page write accesses. Page write accesses are not supported in the address/data multiplexed I/O interface.

PRMOD Bit (Page Read Access Mode Select)

This bit selects a page read access operating mode.

Writing 0 to this bit selects normal access compatible mode where the RD# signal is negated and RD assert wait is inserted each time a piece of data is read. However, when there is no RD assert wait, the RD# signal is negated only in the final transfer of the external bus access.

Writing 1 to this bit selects external data read continuous assertion mode where RD assert wait is inserted and the RD# signal is continuously asserted during this time period.

16.3.5 CSn Wait Control Register 1 (CSnWCR1) (n = 0 to 7)

Address(es): CS0WCR1 0008 3004h, CS1WCR1 0008 3014h, CS2WCR1 0008 3024h, CS3WCR1 0008 3034h, CS4WCR1 0008 3044h, CS5WCR1 0008 3054h, CS6WCR1 0008 3064h, CS7WCR1 0008 3074h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	CSRWAIT[4:0]					—	—	—	CSWWAIT[4:0]				
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CSPRWAIT[2:0]			—	—	—	—	—	CSPWWAIT[2:0]		
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CSPWWAIT[2:0]	Page Write Cycle Wait Select*1	b2 b0 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	CSPRWAIT[2:0]	Page Read Cycle Wait Select*2	b10 b8 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b20 to b16	CSWWAIT[4:0]	Normal Write Cycle Wait Select	b20 b16 0 0 0 0 0: No wait is inserted. 0 0 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 0 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 0 0 1 1: Wait with a length of 3 clock cycles are inserted. 0 0 1 0 0: Wait with a length of 4 clock cycles are inserted. 0 0 1 0 1: Wait with a length of 5 clock cycles are inserted. 0 0 1 1 0: Wait with a length of 6 clock cycles are inserted. 0 0 1 1 1: Wait with a length of 7 clock cycles are inserted. 0 1 0 0 0: Wait with a length of 8 clock cycles are inserted. 0 1 0 0 1: Wait with a length of 9 clock cycles are inserted. 0 1 0 1 0: Wait with a length of 10 clock cycles are inserted. 0 1 0 1 1: Wait with a length of 11 clock cycles are inserted. 0 1 1 0 0: Wait with a length of 12 clock cycles are inserted. 0 1 1 0 1: Wait with a length of 13 clock cycles are inserted. 0 1 1 1 0: Wait with a length of 14 clock cycles are inserted. 0 1 1 1 1: Wait with a length of 15 clock cycles are inserted. 1 0 0 0 0: Wait with a length of 16 clock cycles are inserted. 1 0 0 0 1: Wait with a length of 17 clock cycles are inserted. 1 0 0 1 0: Wait with a length of 18 clock cycles are inserted. 1 0 0 1 1: Wait with a length of 19 clock cycles are inserted. 1 0 1 0 0: Wait with a length of 20 clock cycles are inserted. 1 0 1 0 1: Wait with a length of 21 clock cycles are inserted. 1 0 1 1 0: Wait with a length of 22 clock cycles are inserted. 1 0 1 1 1: Wait with a length of 23 clock cycles are inserted. 1 1 0 0 0: Wait with a length of 24 clock cycles are inserted. 1 1 0 0 1: Wait with a length of 25 clock cycles are inserted. 1 1 0 1 0: Wait with a length of 26 clock cycles are inserted. 1 1 0 1 1: Wait with a length of 27 clock cycles are inserted. 1 1 1 0 0: Wait with a length of 28 clock cycles are inserted. 1 1 1 0 1: Wait with a length of 29 clock cycles are inserted. 1 1 1 1 0: Wait with a length of 30 clock cycles are inserted. 1 1 1 1 1: Wait with a length of 31 clock cycles are inserted.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28 to b24	CSRWAIT[4:0]	Normal Read Cycle Wait Select	b28 b24 0 0 0 0 0: No wait is inserted. 0 0 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 0 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 0 0 1 1: Wait with a length of 3 clock cycles are inserted. 0 0 1 0 0: Wait with a length of 4 clock cycles are inserted. 0 0 1 0 1: Wait with a length of 5 clock cycles are inserted. 0 0 1 1 0: Wait with a length of 6 clock cycles are inserted. 0 0 1 1 1: Wait with a length of 7 clock cycles are inserted. 0 1 0 0 0: Wait with a length of 8 clock cycles are inserted. 0 1 0 0 1: Wait with a length of 9 clock cycles are inserted. 0 1 0 1 0: Wait with a length of 10 clock cycles are inserted. 0 1 0 1 1: Wait with a length of 11 clock cycles are inserted. 0 1 1 0 0: Wait with a length of 12 clock cycles are inserted. 0 1 1 0 1: Wait with a length of 13 clock cycles are inserted. 0 1 1 1 0: Wait with a length of 14 clock cycles are inserted. 0 1 1 1 1: Wait with a length of 15 clock cycles are inserted. 1 0 0 0 0: Wait with a length of 16 clock cycles are inserted. 1 0 0 0 1: Wait with a length of 17 clock cycles are inserted. 1 0 0 1 0: Wait with a length of 18 clock cycles are inserted. 1 0 0 1 1: Wait with a length of 19 clock cycles are inserted. 1 0 1 0 0: Wait with a length of 20 clock cycles are inserted. 1 0 1 0 1: Wait with a length of 21 clock cycles are inserted. 1 0 1 1 0: Wait with a length of 22 clock cycles are inserted. 1 0 1 1 1: Wait with a length of 23 clock cycles are inserted. 1 1 0 0 0: Wait with a length of 24 clock cycles are inserted. 1 1 0 0 1: Wait with a length of 25 clock cycles are inserted. 1 1 0 1 0: Wait with a length of 26 clock cycles are inserted. 1 1 0 1 1: Wait with a length of 27 clock cycles are inserted. 1 1 1 0 0: Wait with a length of 28 clock cycles are inserted. 1 1 1 0 1: Wait with a length of 29 clock cycles are inserted. 1 1 1 1 0: Wait with a length of 30 clock cycles are inserted. 1 1 1 1 1: Wait with a length of 31 clock cycles are inserted.	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The CSPWWAIT[2:0] value is valid only when the PWENB bit in CSnMOD is set to 1.
Note 2. The CSPRWAIT[2:0] value is valid only when the PRENB bit in CSnMOD is set to 1.

Do not attempt to write the CSnWCR1 register while the external bus is being accessed.
Set each of these bits within a range of the restrictions described in section 16.5.7 (1) Limitations on Using Separate Bus Interface or section 16.5.7 (2) Limitations on Using Address/Data Multiplexed Bus Interface, according to the bus interface used. In addition, during the EXDMAC transfer in single address mode, set each of these bits within a range of the restrictions described in section 16.5.7 (5) Limitations on EXDMAC Single Address Transfer Mode.

CSPWWAIT[2:0] Bits (Page Write Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page write cycle.
This setting is enabled when the PWENB bit in CSnMOD is set to 1.

Note: Be sure to satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

CSPRWAIT[2:0] Bits (Page Read Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page read cycle.
This setting is enabled when the PRENB bit in CSnMOD is set to 1.

Note: Be sure to satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.

CSWWAIT[4:0] Bits (Normal Write Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the first access during a normal write cycle or page write cycle.

Note: Be sure to satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

CSRWAIT[4:0] Bits (Normal Read Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the first access during a normal read cycle or page read cycle.

Note: Be sure to satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

16.3.6 CSn Wait Control Register 2 (CSnWCR2) (n = 0 to 7)

Address(es): CS0WCR2 0008 3008h, CS1WCR2 0008 3018h, CS2WCR2 0008 3028h, CS3WCR2 0008 3038h, CS4WCR2 0008 3048h, CS5WCR2 0008 3058h, CS6WCR2 0008 3068h, CS7WCR2 0008 3078h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CSON[2:0]			—	WDON[2:0]			—	WRON[2:0]			—	RDON[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AWAIT[1:0]		—	WDOFF[2:0]			—	CSWOFF[2:0]			—	CSROFF[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CSROFF[2:0]	Read-Access CS Extension Cycle Select	b2 b1 b0 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	CSWOFF[2:0]	Write-Access CS Extension Cycle Select	b6 b5 b4 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	WDOFF[2:0]	Write Data Output Extension Cycle Select	b10 b9 b8 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13, b12	AWAIT[1:0]	Address Cycle Wait Select	b13 b12 0 0: No wait is inserted. 0 1: Wait with a length of 1 clock cycle is inserted. 1 0: Wait with a length of 2 clock cycles are inserted. 1 1: Wait with a length of 3 clock cycles are inserted.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	RDON[2:0]	RD Assert Wait Select	b18 b17 b16 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b22 to b20	WRON[2:0]	WR Assert Wait Select	b22 b20 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b26 to b24	WDON[2:0]	Write Data Output Wait Select	b26 b24 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30 to b28	CSON[2:0]	CS Assert Wait Select	b30 b28 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Do not attempt to write the CSnWCR2 register while the external bus is being accessed.
Set each of these bits within a range of the restrictions described in section 16.5.7 (1) Limitations on Using Separate Bus Interface or section 16.5.7 (2) Limitations on Using Address/Data Multiplexed Bus Interface, according to the bus interface used. In addition, during the EXDMAC transfer in single address mode, set each of these bits within a range of the restrictions described in section 16.5.7 (5) Limitations on EXDMAC Single Address Transfer Mode.

CSROFF[2:0] Bits (Read-Access CS Extension Cycle Select)

These bits specify the number of wait cycles to be inserted in a time period from the end of a wait cycle (RD# signal negated) until the CSn# signal (n = 0 to 7) is negated in read access mode.

CSWOFF[2:0] Bits (Write-Access CS Extension Cycle Select)

These bits specify the number of wait cycles to be inserted in a time period from the end of a wait cycle (WRn# signal (n = 0 to 3) negated) until the CSn# signal (n = 0 to 7) is negated in write access mode.

Note: Be sure to satisfy CSnWCR2.WDOFF[2:0] value ≤ CSnWCR2.CSWOFF[2:0] value.

WDOFF[2:0] Bits (Write Data Output Extension Cycle Select)

These bits specify the number of wait cycles to be inserted in a time period from the end of a wait cycle (WRn# signal (n = 0 to 3) negated) until the write data output is completed in write access mode.
When the EXDMAC is in single-address transfer mode, although the output of write-data from the chip does not proceed, the value for cycles of delay until output of the write data in divided-up page access over the bus becomes effective.

Note: Be sure to satisfy CSnWCR2.WDOFF[2:0] value ≤ CSnWCR2.CSWOFF[2:0] value.

AWAIT[1:0] Bits (Address Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into an address output cycle with the address/data multiplexed I/O interface.

Note: CSnWCR2.CSON[2:0] value ≤ CSnWCR2.AWAIT[1:0] value
For read access, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.
For write access, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value and CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

RDON[2:0] Bits (RD Assert Wait Select)

These bits specify the number of wait cycles to be inserted before the RD# signal is asserted.

Note: For normal read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.
For page read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.
Note: For read access by the EXDMAC in single-address transfer mode, as well as satisfying the above conditions, set the CSnWCR2.RDON[2:0] bits to one or a greater value.
Note: When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

WRON[2:0] Bits (WR Assert Wait Select)

These bits specify the number of wait cycles to be inserted before the WRn# signal (n = 0 to 3) is asserted.

Note: For normal write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.
For page write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.
Note: For write access by the EXDMAC in single-address transfer mode, as well as satisfying the above conditions, set the CSnWCR2.WRON[2:0] bits to one or a greater value.
Note: When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

WDON[2:0] Bits (Write Data Output Wait Select)

These bits specify the number of wait cycles to be inserted before the write data is output.

Note: For normal write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.
For page write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.
Note: When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

CSON[2:0] Bits (CS Assert Wait Select)

These bits specify the number of wait cycles to be inserted before the CSn# signal (n = 0 to 7) is asserted.

Note: For normal read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.
For page read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.

For normal write access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.
For page write access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

Note: When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.AWAIT[1:0] value.

16.3.7 SDC Control Register (SDCCR)

Address(es): 0008 3C00h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	BSIZE[1:0]		—	—	—	EXENB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EXENB	Operation Enable	0: Operation is disabled 1: Operation is enabled	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	BSIZE[1:0]	SDRAM Bus Width Select	b5 b4 0 0: A 16-bit bus space is selected 0 1: A 32-bit bus space is selected 1 0: An 8-bit bus space is selected 1 1: Setting prohibited	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

EXENB Bit (Operation Enable)

This bit enables or disables the operation of the SDRAM address space. After reset, this bit is set to 0 (operation disabled). An attempt at access to an area for which operation has been disabled does not lead to SDRAM access. If the illegal address access detection enable bit (IGAEN) in the bus error monitoring enable register (BEREN) has been set to 1 (detection enabled), such an attempt will lead to a bus error.

16.3.8 SDC Mode Register (SDCMOD)

Address(es): 0008 3C01h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	EMODE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EMODE	Endian Mode	0: Endian of SDRAM address space is the same as the endian of operating mode. 1: Endian of SDRAM address space is not the endian of operating mode.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Writing to this register is possible only once after release from the reset state. Operation is not guaranteed after write access more than once is attempted.

EMODE Bit (Endian Mode)

This bit specifies the endian of the SDRAM address space.

When the endian setting for each area is different from that for the chip, no instruction code can be allocated in the area. The instruction code should be allocated to the external space whose endian setting is the same as that for the chip.

16.3.9 SDRAM Access Mode Register (SDAMOD)

Address(es): 0008 3C02h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	BE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BE	Continuous Access Enable	0: Continuous access is disabled 1: Continuous access is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Set SDAMOD while the conditions listed in Table 16.15, Conditions for Register Modification, are satisfied. The operation is not guaranteed if this register is set while these conditions are not satisfied.

BE Bit (Continuous Access Enable)

This bit enables or disables continuous access to the SDRAM access space.

Note: When the SDRAM area is accessed from bus masters other than EXDMAC, continuous access is always disabled regardless of the setting.

16.3.10 SDRAM Self-Refresh Control Register (SDSELF)

Address(es): 0008 3C10h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SFEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SFEN	SDRAM Self-Refresh Enable	0: Self-refresh is disabled 1: Self-refresh is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Set SDSELF while the conditions listed in Table 16.15, Conditions for Register Modification, are satisfied. The operation is not guaranteed if this register is set while these conditions are not satisfied.

SFEN Bit (SDRAM Self-Refresh Enable)

This bit controls self-refresh operation.
Setting this bit to 1 performs auto-refresh cycle operation, after which self-refresh operation begins.
Clearing this bit to 0 ends self-refresh operation, and auto-refresh operation resumes afterwards.
If this bit was set to 1, the value written to this bit is reflected when self-refresh operation starts. If this bit was set to 0, the value written to this bit has already been reflected when auto-refresh operation starts following the end of self-refresh operation.

16.3.11 SDRAM Refresh Control Register (SDRFCR)

Address(es): 0008 3C14h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REFW[3:0]				RFC[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	RFC[11:0]	Auto-Refresh Request Interval Setting	b11 b0 0 0 0 0 0 0 0 0 0 0 0 0: Setting prohibited 0 0 0 0 0 0 0 0 0 0 0 1: 2 cycles 0 0 0 0 0 0 0 0 0 0 1 0: 3 cycles : 1 1 1 1 1 1 1 1 1 1 1 1: 4096 cycles	R/W
b15 to b12	REFW[3:0]	Auto-Refresh Cycle/ Self-Refresh Clearing Cycle Count Setting	b15 b12 0 0 0 0: 1 cycle 0 0 0 1: 2 cycles 0 0 1 0: 3 cycles 0 0 1 1: 4 cycles 0 1 0 0: 5 cycles 0 1 0 1: 6 cycles 0 1 1 0: 7 cycles 0 1 1 1: 8 cycles 1 0 0 0: 9 cycles 1 0 0 1: 10 cycles 1 0 1 0: 11 cycles 1 0 1 1: 12 cycles 1 1 0 0: 13 cycles 1 1 0 1: 14 cycles 1 1 1 0: 15 cycles 1 1 1 1: 16 cycles	R/W

RFC[11:0] Bits (Auto-Refresh Request Interval Setting)

These bits specify the auto-refresh request interval.

These bits can be written to at any time, regardless of the state of the auto-refresh operation enable (RFEN) bit in SDRFEN.

If auto-refresh is enabled, the value written to these bits is reflected after the end of auto-refresh cycles. The refresh counter operates in SDCLK.

REFW[3:0] Bits (Auto-Refresh Cycle/ Self-Refresh Clearing Cycle Count Setting)

These bits specify the number of auto-refresh cycles and the number of self-refresh clearing cycles.

These bits can be written to at any time, regardless of the state of the auto-refresh operation enable (RFEN) bit in SDRFEN.

If an auto-refresh cycle is in progress, the value written to these bits while auto-refresh is enabled takes effect after the cycle completes.

Note: Auto-refresh requests are not accepted while SDRAM is being accessed; they must wait until the access completes, so the auto-refresh interval may become enlarged in some cases. Set the RFC[11:0] bits to an auto-refresh request interval value that satisfies the auto-refresh interval specification of the SDRAM being used. Furthermore, make sure to set the auto-refresh request interval to a duration longer than the auto-refresh cycle. Note that the auto-refresh interval cannot be automatically adjusted when the frequency is changed during operation; in this case, perform self-refresh operation and set the auto-refresh interval appropriate for the frequency again.

- Auto-Refresh Request Interval and RFC Set Value

SDRAMC (SDRAM area controller) includes a 12-bit refresh counter that generates auto-refresh requests at fixed intervals. The following equation is used to calculate the set value for the RFC[11:0] bits from the auto-refresh request interval.

RFC = (Auto-refresh request interval / SDCLK cycle) – 1

Note: Auto-refresh requests are not accepted while SDRAM is being accessed; they must wait until the access completes. However, the counter value is updated regardless of whether or not the request was accepted. Note that if two or more auto-refresh requests are generated while SDRAM is being accessed, the second and subsequent requests are ignored.

16.3.12 SDRAM Auto-Refresh Control Register (SDRFEN)

Address(es): 0008 3C16h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RFEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFEN	Auto-Refresh Operation Enable	0: Auto-refresh operation is disabled 1: Auto-refresh operation is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RFEN Bit (Auto-Refresh Operation Enable)

Setting this bit to 0 while auto-refreshing is enabled causes RFEN to be set to 0 and auto-refresh operation to halt after the end of the auto-refresh cycle. However, if RFEN is again set to 1 before the end of the auto-refresh cycle, auto-refreshing continues and the RFEN bit is not set to 0. Setting the RFEN bit to 1 while auto-refresh is disabled starts auto-refresh operation, and refresh requests are then generated at fixed intervals determined by a counter. The interval at which refresh requests are generated is determined by the value of the auto-refresh request interval setting (RFC[11:0]) bits in the SDRAM refresh control register (SDRFCR).

Refresh requests are not accepted while SDRAM is being accessed; they must wait until the access completes.

If an SDRAM access and a refresh request are generated at the same time, the refresh request takes precedence.

16.3.13 SDRAM Initialization Sequence Control Register (SDICR)

Address(es): 0008 3C20h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	INIRQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	INIRQ	Initialization Sequence Start	0: Invalid 1: Initialization sequence starts	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Writing to this register is possible only once after release from the reset state. Operation is not guaranteed after write access more than once is attempted.

INIRQ Bit (Initialization Sequence Start)

Setting this bit to 1 causes the SDRAM initialization sequence to start and automatically sets the initialization status bit (INIST) in the SDRAM status register (SDSR) to 1. The INIST bit is cleared automatically after the initialization sequence ends.

The value written to the INIRQ bit is not retained.

If access to an external address space or an external bus controller register occurs after the initialization sequence is started, the access is suspended until the initialization sequence ends.

Note: Set the INIRQ bit to start the SDRAM initialization sequence while the conditions listed in Table 16.15, Conditions for Register Modification, are satisfied. The operation is not guaranteed if this bit is set while these conditions are not satisfied.

16.3.14 SDRAM Initialization Register (SDIR)

Address(es): 0008 3C24h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PRC[2:0]			ARFC[3:0]				ARFI[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ARFI[3:0]	Initialization Auto-Refresh Interval	b3 b0 0 0 0 0: 3 cycles 0 0 0 1: 4 cycles 0 0 1 0: 5 cycles 0 0 1 1: 6 cycles 0 1 0 0: 7 cycles 0 1 0 1: 8 cycles 0 1 1 0: 9 cycles 0 1 1 1: 10 cycles 1 0 0 0: 11 cycles 1 0 0 1: 12 cycles 1 0 1 0: 13 cycles 1 0 1 1: 14 cycles 1 1 0 0: 15 cycles 1 1 0 1: 16 cycles 1 1 1 0: 17 cycles 1 1 1 1: 18 cycles	R/W
b7 to b4	ARFC[3:0]	Initialization Auto-Refresh Count	b7 b4 0 0 0 0: Setting prohibited 0 0 0 1: 1 time 0 0 1 0: 2 times 0 0 1 1: 3 times 0 1 0 0: 4 times 0 1 0 1: 5 times 0 1 1 0: 6 times 0 1 1 1: 7 times 1 0 0 0: 8 times 1 0 0 1: 9 times 1 0 1 0: 10 times 1 0 1 1: 11 times 1 1 0 0: 12 times 1 1 0 1: 13 times 1 1 1 0: 14 times 1 1 1 1: 15 times	R/W
b10 to b8	PRC[2:0]	Initialization Precharge Cycle Count	b10 b8 0 0 0: 3 cycles 0 0 1: 4 cycles 0 1 0: 5 cycles 0 1 1: 6 cycles 1 0 0: 7 cycles 1 0 1: 8 cycles 1 1 0: 9 cycles 1 1 1: 10 cycles	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Writing to this register is possible only once after release from the reset state. Operation is not guaranteed after write access more than once is attempted.

ARFI[3:0] Bits (Initialization Auto-Refresh Interval)

These bits specify the interval at which auto-refresh commands are issued in the SDRAM initialization sequence.

ARFC[3:0] Bits (Initialization Auto-Refresh Count)

These bits specify the number of times auto-refresh is to be performed in the SDRAM initialization sequence.

PRC[2:0] Bits (Initialization Precharge Cycle Count)

These bits specify the number of precharge cycles in the SDRAM initialization sequence.

Note: Make settings that satisfy the specifications of the connected SDRAM before starting the initialization sequence.

16.3.15 SDRAM Address Register (SDADR)

Address(es): 0008 3C40h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MXC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MXC[1:0]	Address Multiplex Select	b1 b0 0 0: 8-bit shift 0 1: 9-bit shift 1 0: 10-bit shift 1 1: 11-bit shift	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Set SDADR while the conditions listed in Table 16.15, Conditions for Register Modification, are satisfied. The operation is not guaranteed if this bit is set while these conditions are not satisfied.

MXC[1:0] Bits (Address Multiplex Select)

These bits select the size of the shift toward the lower half of the row address in row address/column address multiplexing. These bits also select the row address bits to be used for comparison in the SDRAMC continuous access operation.
For details, refer to Table 16.20, Address Multiplexing.

16.3.16 SDRAM Timing Register (SDTR)

Address(es): 0008 3C44h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	RAS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RCD[1:0]		RP[2:0]			WR	—	—	—	—	—	CL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CL[2:0]	SDRAMC Column Latency	b2 b0 0 0 0: Setting prohibited 0 0 1: 1 cycle 0 1 0: 2 cycles 0 1 1: 3 cycles 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: Setting prohibited	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	WR	Write Recovery Interval	0: 1 cycle 1: 2 cycles	R/W
b11 to b9	RP[2:0]	Row Precharge Interval	b11 b9 0 0 0: 1 cycle 0 0 1: 2 cycles 0 1 0: 3 cycles 0 1 1: 4 cycles 1 0 0: 5 cycles 1 0 1: 6 cycles 1 1 0: 7 cycles 1 1 1: 8 cycles	R/W
b13, b12	RCD[1:0]	Row Column Latency	b13 b12 0 0: 1 cycle 0 1: 2 cycles 1 0: 3 cycles 1 1: 4 cycles	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	RAS[2:0]	Row Active Interval	b18 b16 0 0 0: 1 cycle 0 0 1: 2 cycles 0 1 0: 3 cycles 0 1 1: 4 cycles 1 0 0: 5 cycles 1 0 1: 6 cycles 1 1 0: 7 cycles 1 1 1: Setting prohibited	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SDTR specifies the timing for read and write accesses to SDRAM. For details, see section 16.6.12.3, Timing Register Settings and Access Timing.

Set SDTR while the conditions listed in Table 16.15, Conditions for Register Modification, are satisfied. The operation is not guaranteed if this bit is set while these conditions are not satisfied.

This register can be written to only once after a reset. When it is written to multiple times, the operation is not guaranteed.

Writing to this register is possible only once after release from the reset state. Operation is not guaranteed after write access more than once is attempted.

CL[2:0] Bits (SDRAMC Column Latency)

These bits specify the column latency of the SDRAM controller. This setting only affects the latency setting on the SDRAM controller side. To specify the column latency for externally connected SDRAM, use the SDRAM mode register (SDMOD), which is described below.

WR Bit (Write Recovery Interval)

This bit specifies the interval that must elapse between the SDRAM write command (WRIT) and deactivation (PALL).

RP[2:0] Bits (Row Precharge Interval)

These bits specify the minimum number of cycles that must elapse between the SDRAM deactivation command (PALL) and the next valid command.

RAS[2:0] Bits (Row Active Interval)

These bits specify the minimum interval that must elapse between the SDRAM row activation command (ACTV) and deactivation (PALL). The value specified by these bits should be less than or equal to the sum of the row-column latency (RCD[1:0]) and column latency (CL[2:0]) settings.

16.3.17 SDRAM Mode Register (SDMOD)

Address(es): 0008 3C48h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	MR[14:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b14 to b0	MR[14:0]	Mode Register Setting	Writing to these bits: Mode register set command is issued	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

SDMOD specifies the value to be written to the SDRAM mode register.
Writing to SDMOD causes a mode register set command to be issued automatically to SDRAM.
Set SDMOD while the conditions listed in Table 16.15, Conditions for Register Modification, are satisfied. The operation is not guaranteed if this bit is set while these conditions are not satisfied.
This register can be written to only once after a reset. When it is written to multiple times, the operation is not guaranteed.
Writing to this register is possible only once after release from the reset state. Operation is not guaranteed after write access more than once is attempted.

MR[14:0] Bits (Mode Register Setting)

Writing to these bits causes a mode register set command to be issued to SDRAM. The setting of the MR[14:0] bits is output to the lower bits of the address. For details, refer to section 16.6.11, Setting Mode Register.
If access to an external address space or an external bus controller register occurs after writing to mode register, the access is suspended until the mode register set command is issued.

Note: The following points should be kept in mind regarding SDMOD settings.
- Make sure to set a burst length of 1 for SDRAM. Operation cannot be guaranteed with settings other than burst length 1.
- The SDRAM column latency must match the setting of the SDRAMC column latency setting bits (CL[2:0]) in the SDRAM timing register (SDTR). Operation cannot be guaranteed if the latency settings do not agree.
- Make sure the status bits (SRFST, INIST, and MRSST) in the SDRAM status register (SDSR) are all 0.

16.3.18 SDRAM Status Register (SDSR)

Address(es): 0008 3C50h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SRFST	INIST	—	—	MRSST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MRSST	Mode Register Setting Status	0: Mode register setting not in progress 1: Mode register setting in progress	R
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	INIST	Initialization Status	0: Initialization sequence not in progress 1: Initialization sequence in progress	R
b4	SRFST	Self-Refresh Transition/Recovery Status	0: Transition/recovery not in progress 1: Transition/recovery in progress	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

MRSST Bit (Mode Register Setting Status)

When set to 1, this bit indicates that SDRAM mode register setting is in progress. If SDSR is accessed during the mode register setting operation, the CPU processing can be suspended until the setting operation ends.

INIST Bit (Initialization Status)

When set to 1, this bit indicates that the SDRAM initialization sequence is in progress. If SDSR is accessed during initialization sequence, the CPU processing can be suspended until the initialization sequence ends.

SRFST Bit (Self-Refresh Transition/Recovery Status)

When set to 1, this bit indicates that a transition to or recovery from self-refresh operation is in progress for SDRAM. “Transition to or recovery from self-refresh operation in progress” refers to the interval from the point at which the bits listed in Table 16.9 are written until the corresponding commands are issued.

Note: Execution of a self-refresh, an initialization sequence, or mode register setting may only be performed when all the status bits are 0. Do not rewrite the registers (bits) listed in Table 16.9 when any of the status bits (SRFST, INIST, MRSST) is set to 1.

Table 16.9 List of Registers and Bits Requiring Checking Status Bits

Function	Register	Bits
Self-refresh	SDSELF	SFEN
Initialization sequence	SDICR	INIRQ
Mode register setting	SDMOD	MR[14:0]

16.3.19 Bus Error Status Clear Register (BERCLR)

Address(es): 0008 1300h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	STSCLR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STSCLR	Status Clear	0: Invalid 1: Bus error status register cleared	(W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only writing 1 is effective; i.e. writing 0 has no effect.

STSCLR Bit (Status Clear)

Writing 1 to this bit clears the bus error status registers 1 and 2 (BERSR1 and BERSR2).
Writing 0 has no effect. It is read as 0.

16.3.20 Bus Error Monitoring Enable Register (BEREN)

Address(es): 0008 1304h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TOEN	IGAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IGAEN	Illegal Address Access Detection Enable	0: Illegal address access detection is disabled. 1: Illegal address access detection is enabled.	R/W
b1	TOEN	Timeout Detection Enable*1, *2	0: Bus timeout detection is disabled. 1: Bus timeout detection is enabled.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When detection is disabled (the TOEN bit is set to 0), bus access can cause the bus to freeze.
Note 2. Do not set the TOEN bit to 0 (bus timeout detection disabled) while timeout errors are being detected.

16.3.21 Bus Error Status Register 1 (BERSR1)

Address(es): 0008 1308h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MST[2:0]			—	—	TO	IA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IA	Illegal Address Access	0: Illegal address access not made 1: Illegal address access made	R
b1	TO	Timeout	0: Timeout not generated 1: Timeout generated	R
b3, b2	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b6 to b4	MST[2:0]	Bus Master Code	b6 b4 0 0 0: CPU 0 0 1: Reserved 0 1 0: Reserved 0 1 1: DTC/DMAC 1 0 0: Reserved 1 0 1: Reserved 1 1 0: EDMAC 1 1 1: EXDMAC	R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

MST[2:0] Bits (Bus Master Code)

These bits indicate the bus master that accessed a bus when a bus error occurred.

16.3.22 Bus Error Status Register 2 (BERSR2)

Address(es): 0008 130Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADDR[12:0]													—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b15 to b3	ADDR[12:0]	Bus Error Occurrence Address	The upper 13 bits of an address that was accessed when a bus error occurred (in units of 512 Kbytes).	R

16.3.23 Bus Priority Control Register (BUSPRI)

Address(es): 0008 1310h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	BPEB[1:0]		BPFB[1:0]		BPHB[1:0]		BPGB[1:0]		BPIB[1:0]		BPRO[1:0]		BPRA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	BPRA[1:0]	Memory Bus 1 and 3 (RAM/ ECCRAM) Priority Control	b1 b0 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b3, b2	BPRO[1:0]	Memory Bus 2 (Code Flash Memory) Priority Control	b3 b2 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b5, b4	BPIB[1:0]	Internal Peripheral Bus 1 Priority Control	b5 b4 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b7, b6	BPGB[1:0]	Internal Peripheral Bus 2 and 3 Priority Control	b7 b6 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b9, b8	BPHB[1:0]	Internal Peripheral Bus 4 and 5 Priority Control	b9 b8 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b11, b10	BPFB[1:0]	Internal Peripheral Bus 6 Priority Control	b11 b10 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b13, b12	BPEB[1:0]	External Bus Priority Control	b13 b12 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be written to only once while the DTC, DMAC, EXDMAC, and EDMAC are written to more than one time, the operation is not guaranteed.

BPRA[1:0] Bits (Memory Bus 1 and 3 (RAM/ECCRAM) Priority Control)

These bits specify the priority order for memory bus 1 (RAM) and memory bus 3 (ECCRAM).

When the priority order is fixed, internal main bus 2 has priority over CPU buses (instruction and operand buses).

When the priority order is toggled, the bus from which a request has currently been accepted has the lower priority.

BPRO[1:0] Bits (Memory Bus 2 (Code Flash Memory) Priority Control)

These bits specify the priority order for memory bus 2 (code flash memory).

When the priority order is fixed, internal main bus 2 has priority over CPU buses (instruction and operand buses).

When the priority order is toggled, the bus from which a request has currently been accepted has the lower priority.

BPIB[1:0] Bits (Internal Peripheral Bus 1 Priority Control)

These bits specify the priority order for internal peripheral bus 1.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPGB[1:0] Bits (Internal Peripheral Bus 2 and 3 Priority Control)

These bits specify the priority order for internal peripheral buses 2 and 3.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPHB[1:0] Bits (Internal Peripheral Bus 4 and 5 Priority Control)

These bits specify the priority order for internal peripheral buses 4 and 5.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPFB[1:0] Bits (Internal Peripheral Bus 6 Priority Control)

These bits specify the priority order for internal peripheral bus 6.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPEB[1:0] Bits (External Bus Priority Control)

These bits specify the priority order for the external bus.
When the priority order is fixed, the order is EXDMAC, internal main bus 2, and then internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted, between internal main bus 1 and other buses (internal main bus 2 and EXDMAC). However, the order of priority is EXDMAC and then internal main bus 2, regardless of the BPEB[1:0] bits settings.

16.4 Endian and Data Alignment

The external bus has a data-alignment function to control which byte of the data bus (D31 to D24, D23 to D16, D15 to D8, or D7 to D0) is used according to the bus specifications of the area to be accessed (8-bit, 16-bit, or 32-bit bus space), data size, and endian format when accessing the external address space (the CS and SDRAM areas).

16.4.1 Data Alignment Control for CS Area

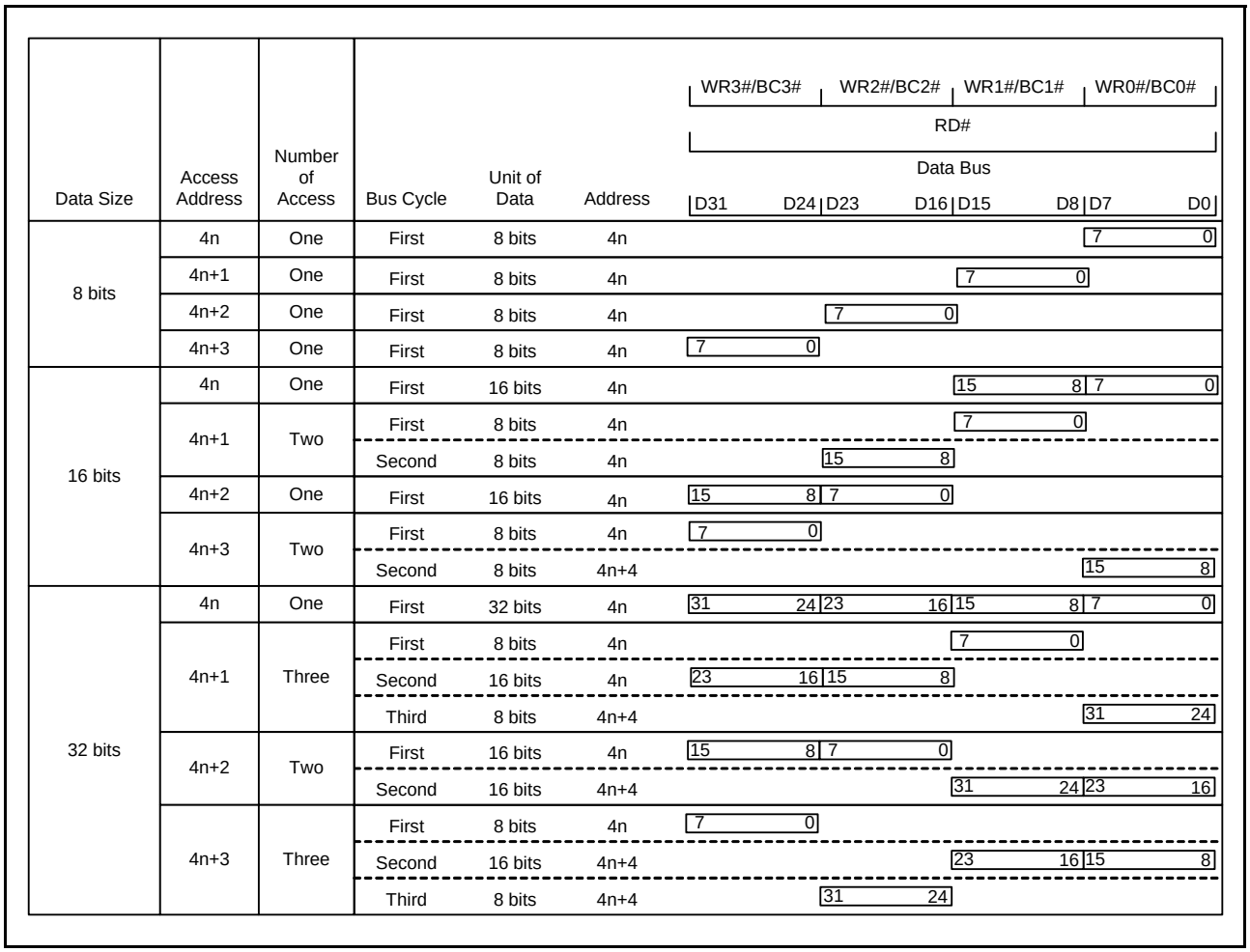
(1) 32-Bit Bus Space

When a 32-bit width is selected for a bus space by the BSIZE[1:0] bits in CSnCR, the address buses A23 to A2 are enabled to output address signals in units of 32 bits, and the address buses A1 and A0 are disabled (always output the low level).

When byte strobe mode is selected (the WRMOD bit = 0 in CSnMOD), the WR0# to WR3# pins are enabled. The BC0# to BC3# pins are not used.

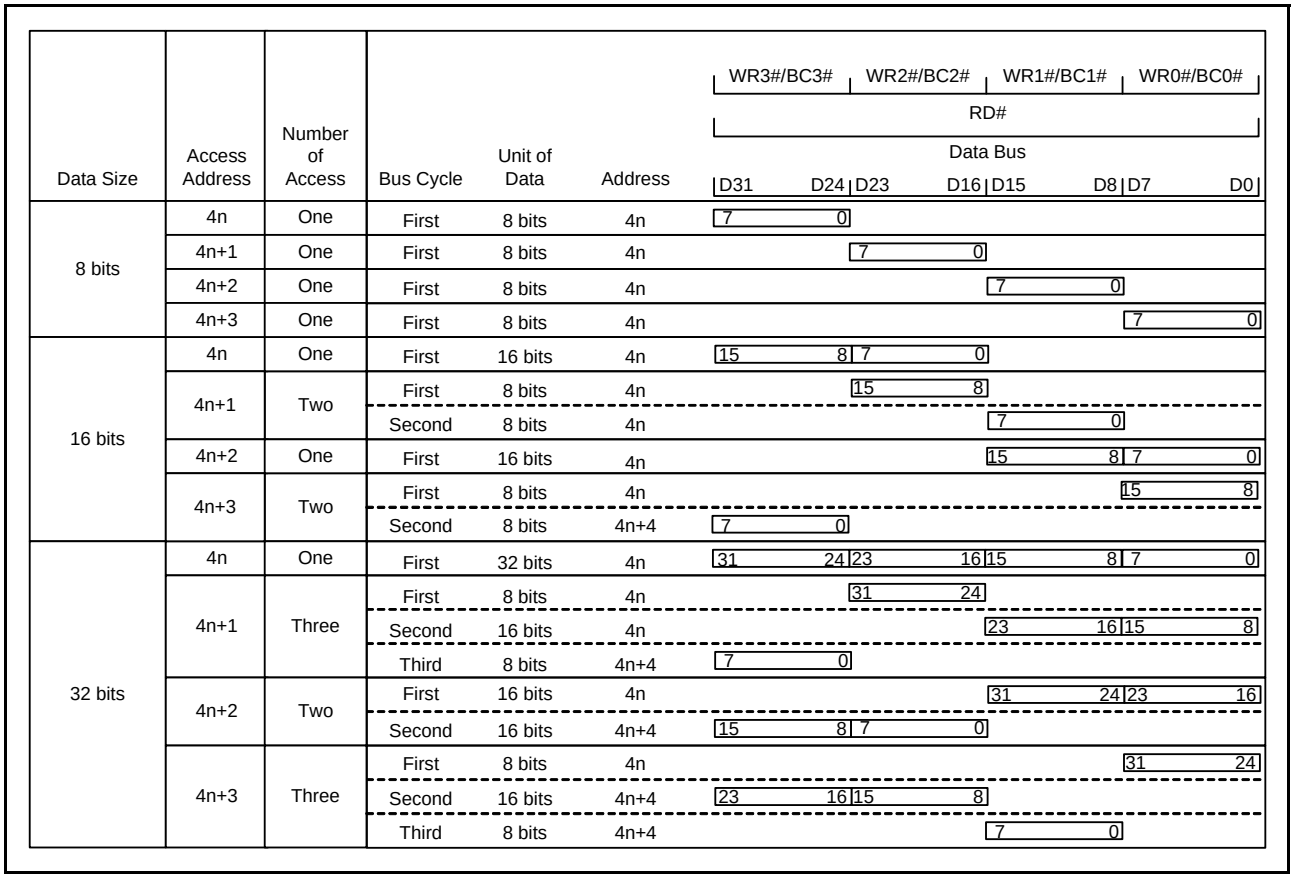
When single write strobe mode is selected (the WRMOD bit = 1 in CSnMOD), only the WR0# pin is enabled and always outputs the low level during write access, regardless of the data size. Here, the WR1# to WR3# pins are invalid (always output the high level). The valid byte position is indicated by the BC0# to BC3# pins.

In 32-bit bus space, the valid positions of data external to the chip and of control signals differ according to whether the endian is big or little.



Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address
8 bits	4n	One	First	8 bits	4n
	4n+1	One	First	8 bits	4n
	4n+2	One	First	8 bits	4n
	4n+3	One	First	8 bits	4n
16 bits	4n	One	First	16 bits	4n
	4n+1	Two	First	8 bits	4n
			Second	8 bits	4n
	4n+2	One	First	16 bits	4n
	4n+3	Two	First	8 bits	4n
			Second	8 bits	4n+4
32 bits	4n	One	First	32 bits	4n
	4n+1	Three	First	8 bits	4n
			Second	16 bits	4n
			Third	8 bits	4n+4
	4n+2	Two	First	16 bits	4n
			Second	16 bits	4n+4
	4n+3	Three	First	8 bits	4n
			Second	16 bits	4n+4
			Third	8 bits	4n+4

Figure 16.6 Data Alignment (Little Endian) in 32-Bit Bus Space



Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address
8 bits	4n	One	First	8 bits	4n
	4n+1	One	First	8 bits	4n
	4n+2	One	First	8 bits	4n
	4n+3	One	First	8 bits	4n
16 bits	4n	One	First	16 bits	4n
	4n+1	Two	First	8 bits	4n
			Second	8 bits	4n
	4n+2	One	First	16 bits	4n
	4n+3	Two	First	8 bits	4n
			Second	8 bits	4n+4
32 bits	4n	One	First	32 bits	4n
	4n+1	Three	First	8 bits	4n
			Second	16 bits	4n
			Third	8 bits	4n+4
	4n+2	Two	First	16 bits	4n
			Second	16 bits	4n+4
	4n+3	Three	First	8 bits	4n
			Second	16 bits	4n+4
			Third	8 bits	4n+4

Figure 16.7 Data Alignment (Big Endian) in 32-Bit Bus Space

(2) 16-Bit Bus Space

When a 16-bit width is selected for a bus space by the BSIZE[1:0] bits in CSnCR, the address buses A23 to A1 are enabled to output address signals in units of 16 bits, and the address bus A0 is disabled (always output the low level). When byte strobe mode is selected (the WRMOD bit = 0 in CSnMOD), the WR0# and WR1# pins are enabled, and the WR2# and WR3# pins are disabled (fixed high). The BC0# to BC3# pins are not used.

When single write strobe mode is selected (the WRMOD bit = 1 in CSnMOD), only the WR0# pin is enabled and always outputs the low level during write access, regardless of the data size. Here, the WR1# to WR3# pins are invalid (always output the high level). The valid byte position is indicated by the BC0# and BC1# pins. The BC2# and BC3# pins are not used.

In 16-bit bus space, page access can occur in access to data in 32-bit units. Specifically, page access can occur when an access does not spread over a 32-bit boundary and causes no change in BC0# and BC1# signals. The situations in which page access occurs are indicated by the letter (p) in Figure 16.8 and Figure 16.9.

In 16-bit bus space, the valid positions of data external to the chip and of control signals differ according to whether the endian is big or little.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		WR1#/BC1# RD# Data Bus D15 D8	WR0#/BC0# Data Bus D7 D0
8 bits	4n	One	First	8 bits	4n			7 0
	4n+1	One	First	8 bits	4n		7 0	
	4n+2	One	First	8 bits	4n+2			7 0
	4n+3	One	First	8 bits	4n+2		7 0	
16 bits	4n	One	First	16 bits	4n		15 8	7 0
	4n+1	Two	First	8 bits	4n		7 0	
			Second	8 bits	4n+2			15 8
	4n+2	One	First	16 bits	4n+2		15 8	7 0
	4n+3	Two	First	8 bits	4n+2		7 0	
			Second	8 bits	4n+4			15 8
32 bits	4n	Two	First	16 bits	4n		15 8	7 0
			Second	16 bits	4n+2	(p)	31 24	23 16
	4n+1	Three	First	8 bits	4n		7 0	
			Second	16 bits	4n+2		23 16	15 8
			Third	8 bits	4n+4			31 24
	4n+2	Two	First	16 bits	4n+2		15 8	7 0
			Second	16 bits	4n+4		31 24	23 16
	4n+3	Three	First	8 bits	4n+2		7 0	
			Second	16 bits	4n+4		23 16	15 8
			Third	8 bits	4n+6			31 24

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 16.8 Data Alignment (Little Endian) in 16-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	WR1#/BC1# RD# Data Bus D15 D8	WR0#/BC0# RD# Data Bus D7 D0
8 bits	4n	One	First	8 bits	4n	7 0	
	4n+1	One	First	8 bits	4n		7 0
	4n+2	One	First	8 bits	4n+2	7 0	
	4n+3	One	First	8 bits	4n+2		7 0
16 bits	4n	One	First	16 bits	4n	15 8	7 0
	4n+1	Two	First	8 bits	4n		15 8
			Second	8 bits	4n+2	7 0	
	4n+2	One	First	16 bits	4n+2	15 8	7 0
	4n+3	Two	First	8 bits	4n+2		15 8
			Second	8 bits	4n+4	7 0	
32 bits	4n	Two	First	16 bits	4n	31 24	23 16
			Second	16 bits	4n+2 (p)	15 8	7 0
	4n+1	Three	First	8 bits	4n		31 24
			Second	16 bits	4n+2	23 16	15 8
			Third	8 bits	4n+4	7 0	
	4n+2	Two	First	16 bits	4n+2	31 24	23 16
			Second	16 bits	4n+4	15 8	7 0
	4n+3	Three	First	8 bits	4n+2		31 24
			Second	16 bits	4n+4	23 16	15 8
			Third	8 bits	4n+6	7 0	

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 16.9 Data Alignment (Big Endian) in 16-Bit Bus Space

(3) 8-Bit Bus Space

When an 8-bit bus space is selected by the BSIZE[1:0] bits in CSnCR, the address buses A23 to A0 are enabled to output address signals in byte units.
In 8-bit bus space, only the WR0# pin is valid regardless of write access mode, and always outputs the low level during write access. The WR1# to WR3# pins and the BC0# to BC3# pins are not used.
Page access can occur in access to data in 16- or 32-bit units. Specifically, page access can occur when an access does not spread over a 32-bit boundary. The situations in which page access occurs are indicated by the letter (p) in Figure 16.10 and Figure 16.11.
In 8-bit bus space, the valid positions of data external to the chip are D7 to D0 and WR0# is used as the control signal, regardless of the endian mode.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		WR1#/BC1# / WR0#/BC0# / RD# Data Bus D15–D8	D7–D0
8 bits	4n	One	First	8 bits	4n			7 0
	4n+1	One	First	8 bits	4n+1			7 0
	4n+2	One	First	8 bits	4n+2			7 0
	4n+3	One	First	8 bits	4n+3			7 0
16 bits	4n	Two	First	8 bits	4n			7 0
			Second	8 bits	4n+1	(p)		15 8
	4n+1	Two	First	8 bits	4n+1			7 0
			Second	8 bits	4n+2	(p)		15 8
	4n+2	Two	First	8 bits	4n+2			7 0
			Second	8 bits	4n+3	(p)		15 8
	4n+3	Two	First	8 bits	4n+3			7 0
			Second	8 bits	4n+4			15 8
32 bits	4n	Four	First	8 bits	4n			7 0
			Second	8 bits	4n+1	(p)		15 8
			Third	8 bits	4n+2	(p)		23 16
			Fourth	8 bits	4n+3	(p)		31 24
	4n+1	Four	First	8 bits	4n+1			7 0
			Second	8 bits	4n+2	(p)		15 8
			Third	8 bits	4n+3	(p)		23 16
			Fourth	8 bits	4n+4			31 24
	4n+2	Four	First	8 bits	4n+2			7 0
			Second	8 bits	4n+3	(p)		15 8
			Third	8 bits	4n+4			23 16
			Fourth	8 bits	4n+5	(p)		31 24
	4n+3	Four	First	8 bits	4n+3			7 0
			Second	8 bits	4n+4			15 8
			Third	8 bits	4n+5	(p)		23 16
			Fourth	8 bits	4n+6	(p)		31 24

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 16.10 Data Alignment (Little Endian) in 8-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		WR1#/BC1# RD# D15 D8	WR0#/BC0# D7 D0
8 bits	4n	One	First	8 bits	4n			7 0
	4n+1	One	First	8 bits	4n+1			7 0
	4n+2	One	First	8 bits	4n+2			7 0
	4n+3	One	First	8 bits	4n+3			7 0
16 bits	4n	Two	First	8 bits	4n			15 8
			Second	8 bits	4n+1	(p)		7 0
	4n+1	Two	First	8 bits	4n+1			15 8
			Second	8 bits	4n+2	(p)		7 0
	4n+2	Two	First	8 bits	4n+2			15 8
			Second	8 bits	4n+3	(p)		7 0
	4n+3	Two	First	8 bits	4n+3			15 8
			Second	8 bits	4n+4			7 0
32 bits	4n	Four	First	8 bits	4n			31 24
			Second	8 bits	4n+1	(p)		23 16
			Third	8 bits	4n+2	(p)		15 8
			Fourth	8 bits	4n+3	(p)		7 0
	4n+1	Four	First	8 bits	4n+1			31 24
			Second	8 bits	4n+2	(p)		23 16
			Third	8 bits	4n+3	(p)		15 8
			Fourth	8 bits	4n+4			7 0
	4n+2	Four	First	8 bits	4n+2			31 24
			Second	8 bits	4n+3	(p)		23 16
			Third	8 bits	4n+4			15 8
			Fourth	8 bits	4n+5	(p)		7 0
	4n+3	Four	First	8 bits	4n+3			31 24
			Second	8 bits	4n+4			23 16
			Third	8 bits	4n+5	(p)		15 8
			Fourth	8 bits	4n+6	(p)		7 0

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 16.11 Data Alignment (Big Endian) in 8-Bit Bus Space

16.4.2 Data Alignment Control for SDRAM Area

(1) 32-Bit Bus Space

When a 32-bit width is selected for a bus space by the BSIZE[1:0] bits in SDCCR, the address buses A23 to A2 are enabled to output address signals in units of 32 bits, and the address buses A1 and A0 are disabled (always output the low level).

The external data is accessed using the D31 to D24, D23 to D16, D15 to D8, and D7 to D0 pins. Either 8-, 16-, or 32-bit data can be accessed at a time. The valid byte position is indicated by DQM0 to DQM3 signals.

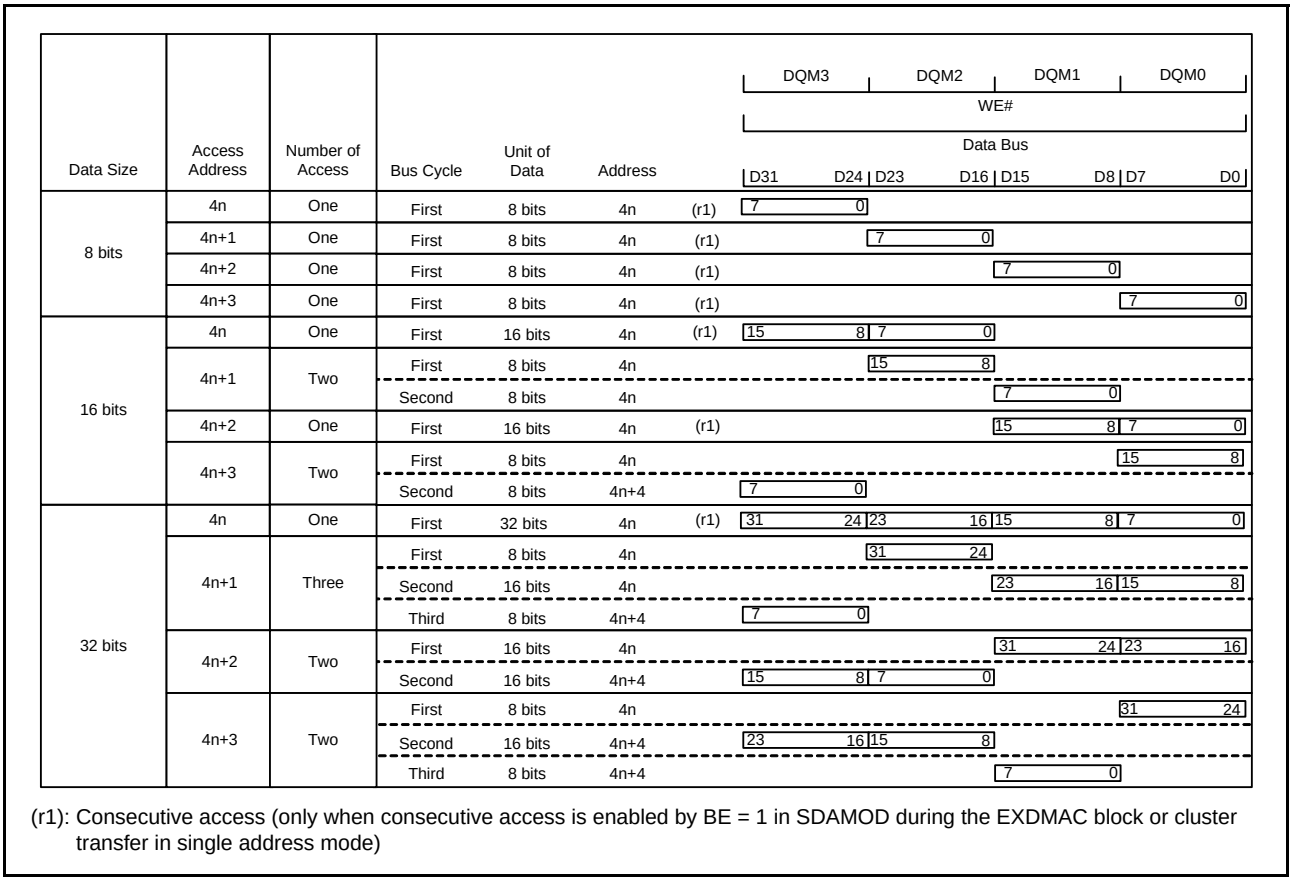
In 32-bit bus space, the valid positions of data external to the chip and of SDRAM control signals (DQM0 to DQM3) differ according to whether the endian is big or little. Figure 16.12 and Figure 16.13 show data alignment control when the endian is little and big, respectively.

In 32-bit bus space, consecutive access can occur in access to data in 8-, 16-, or 32-bit units. Specifically, consecutive access can occur when a single round of bus access is generated in response to a single transfer request. The situations in which consecutive access occurs are indicated by the letter “(r1)” in Figure 16.12 and Figure 16.13. Figure 16.18 shows a consecutive access example.

Data Size	Access Address	Access Address	Bus Cycle	Unit of Data	Address		DQM3 / DQM2 / DQM1 / DQM0 / WE# / Data Bus D31 D24 D23 D16 D15 D8 D7 D0
8 bits	4n	One	First	8 bits	4n	(r1)	7 0
	4n+1	One	First	8 bits	4n	(r1)	7 0
	4n+2	One	First	8 bits	4n	(r1)	7 0
	4n+3	One	First	8 bits	4n	(r1)	7 0
16 bits	4n	One	First	16 bits	4n	(r1)	15 8 7 0
	4n+1	Two	First	8 bits	4n		7 0
			Second	8 bits	4n		15 8
	4n+2	One	First	16 bits	4n	(r1)	15 8 7 0
	4n+3	Two	First	8 bits	4n		7 0
			Second	8 bits	4n+4		15 8
32 bits	4n	One	First	32 bits	4n	(r1)	31 24 23 16 15 8 7 0
	4n+1	Three	First	8 bits	4n		7 0
			Second	16 bits	4n		23 16 15 8
			Third	8 bits	4n+4		31 24
	4n+2	Two	First	16 bits	4n		15 8 7 0
			Second	16 bits	4n+4		31 24 23 16
	4n+3	Two	First	8 bits	4n		7 0
			Second	16 bits	4n+4		23 16 15 8
			Third	8 bits	4n+4		31 24

(r1): Consecutive access (only when consecutive access is enabled by BE = 1 in SDAMOD during the EXDMAC block or cluster transfer in single address mode)

Figure 16.12 Data Alignment (Little Endian) in 32-Bit Bus Space



Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	
8 bits	4n	One	First	8 bits	4n	(r1)
	4n+1	One	First	8 bits	4n	(r1)
	4n+2	One	First	8 bits	4n	(r1)
	4n+3	One	First	8 bits	4n	(r1)
16 bits	4n	One	First	16 bits	4n	(r1)
	4n+1	Two	First	8 bits	4n	
			Second	8 bits	4n	
	4n+2	One	First	16 bits	4n	(r1)
	4n+3	Two	First	8 bits	4n	
			Second	8 bits	4n+4	
32 bits	4n	One	First	32 bits	4n	(r1)
	4n+1	Three	First	8 bits	4n	
			Second	16 bits	4n	
			Third	8 bits	4n+4	
	4n+2	Two	First	16 bits	4n	
			Second	16 bits	4n+4	
	4n+3	Two	First	8 bits	4n	
			Second	16 bits	4n+4	
			Third	8 bits	4n+4	

(r1): Consecutive access (only when consecutive access is enabled by BE = 1 in SDAMOD during the EXDMAC block or cluster transfer in single address mode)

Figure 16.13 Data Alignment (Big Endian) in 32-Bit Bus Space

(2) 16-Bit Bus Space

When a 16-bit width is selected for a bus space by the BSIZE[1:0] bits in SDCCR, the address buses A23 to A1 are enabled to output address signals in units of 16 bits, and the address buses A0 is disabled (always output the low level). The valid byte position is indicated by DQM0 and DQM1 signals. DQM2 and DQM3 signals are not used.

In 16-bit bus space, the external data is accessed using the D15 to D8 and D7 to D0 pins and DQM0 and DQM1 control signals. Either 8- or 16-bit data can be accessed at a time.

In 16-bit bus space, the valid positions of data external to the chip and of control signals differ according to whether the endian is big or little. Figure 16.14 and Figure 16.15 show data alignment control when the endian is little and big, respectively.

In 16-bit bus space, consecutive access can occur in access to data in 8- or 16-bit units. Specifically, consecutive access can occur when a single round of bus access is generated in response to a single transfer request. The situations in which consecutive access occurs are indicated by the letter "(r1)" in Figure 16.14 and Figure 16.15. Figure 16.18 shows a consecutive access example.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		DQM1 / DQM0 / WE# / Data Bus (D15 D8 D7 D0)
8 bits	4n	One	First	8 bits	4n	(r1)	7 0
	4n+1	One	First	8 bits	4n	(r1)	7 0
	4n+2	One	First	8 bits	4n+2	(r1)	7 0
	4n+3	One	First	8 bits	4n+2	(r1)	7 0
16 bit	4n	One	First	16 bits	4n	(r1)	15 8 7 0
	4n+1	Two	First	8 bits	4n		7 0
			Second	8 bits	4n+2		15 8
	4n+2	One	First	16 bits	4n+2	(r1)	15 8 7 0
	4n+3	Two	First	8 bits	4n+2		7 0
			Second	8 bits	4n+4		15 8
32 bits	4n	Two	First	16 bits	4n		15 8 7 0
			Second	16 bits	4n+2		31 24 23 16
	4n+1	Three	First	8 bits	4n		7 0
			Second	16 bits	4n+2		23 16 15 8
			Third	8 bits	4n+4		31 24
	4n+2	Two	First	16 bits	4n+2		15 8 7 0
			Second	16 bits	4n+4		31 24 23 16
	4n+3	Three	First	8 bits	4n+2		7 0
			Second	16 bits	4n+4		23 16 15 8
			Third	8 bits	4n+6		31 24

(r1): Consecutive access (only when consecutive access is enabled by BE = 1 in SDAMOD during the EXDMAC block or cluster transfer in single address mode)

Figure 16.14 Data Alignment (Little Endian) in 16-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		DQM1 / DQM0 / WE# / Data Bus (D15 D8 D7 D0)
8 bits	4n	One	First	8 bits	4n	(r1)	7 0
	4n+1	One	First	8 bits	4n	(r1)	7 0
	4n+2	One	First	8 bits	4n+2	(r1)	7 0
	4n+3	One	First	8 bits	4n+2	(r1)	7 0
16 bits	4n	One	First	16 bits	4n	(r1)	15 8 7 0
	4n+1	Two	First	8 bits	4n		15 8
			Second	8 bits	4n+2		7 0
	4n+2	One	First	16 bits	4n+2	(r1)	15 8 7 0
	4n+3	Two	First	8 bits	4n+2		15 8
			Second	8 bits	4n+4		7 0
32 bits	4n	Two	First	16 bits	4n		31 24 23 16
			Second	16 bits	4n+2		15 8 7 0
	4n+1	Three	First	8 bits	4n		31 24
			Second	16 bits	4n+2		23 16 15 8
			Third	8 bits	4n+4		7 0
	4n+2	Two	First	16 bits	4n+2		31 24 23 16
			Second	16 bits	4n+4		15 8 7 0
	4n+3	Three	First	8 bits	4n+2		31 24
			Second	16 bits	4n+4		23 16 15 8
			Third	8 bits	4n+6		7 0

(r1): Consecutive access (only when consecutive access is enabled by BE = 1 in SDAMOD during the EXDMAC block or cluster transfer in single address mode)

Figure 16.15 Data Alignment (Big Endian) in 16-Bit Bus Space

(3) 8-Bit Bus Space

When an 8-bit width is selected for a bus space by the BSIZE[1:0] bits in SDCCR, the address buses A23 to A0 are enabled to output address signals in units of 8 bits.

In 8-bit bus space, the external data is accessed using the D7 to D0 pins and DQM0 control signal. Eight-bit data can be accessed at a time; 16-bit data is accessed with two 8-bit accesses and 32-bit data is accessed with four 8-bit accesses. Figure 16.16 and Figure 16.17 show data alignment control when the endian is little and big, respectively.

In 8-bit bus space, consecutive access can occur in access to data in 8-bit units. Specifically, consecutive access can occur when a single round of bus access is generated in response to a single transfer request. The situations in which consecutive access occurs are indicated by the letter "(r1)" in Figure 16.16 and Figure 16.17. Figure 16.18 shows a consecutive access example.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		DQM1 / DQM0 / WE# / Data Bus D15 D7 D0
8 bits	4n	One	First	8 bits	4n	(r1)	7 0
	4n+1	One	First	8 bits	4n+1	(r1)	7 0
	4n+2	One	First	8 bits	4n+2	(r1)	7 0
	4n+3	One	First	8 bits	4n+3	(r1)	7 0
16 bits	4n	Two	First	8 bits	4n		7 0
			Second	8 bits	4n+1		15 8
	4n+1	Two	First	8 bits	4n+1		7 0
			Second	8 bits	4n+2		15 8
	4n+2	Two	First	8 bits	4n+2		7 0
			Second	8 bits	4n+3		15 8
	4n+3	Two	First	8 bits	4n+3		7 0
			Second	8 bits	4n+4		15 8
32 bits	4n	Four	First	8 bits	4n		7 0
			Second	8 bits	4n+1		15 8
			Third	8 bits	4n+2		23 16
			Fourth	8 bits	4n+3		31 24
	4n+1	Four	First	8 bits	4n+1		7 0
			Second	8 bits	4n+2		15 8
			Third	8 bits	4n+3		23 16
			Fourth	8 bits	4n+4		31 24
	4n+2	Four	First	8 bits	4n+2		7 0
			Second	8 bits	4n+3		15 8
			Third	8 bits	4n+4		23 16
			Fourth	8 bits	4n+5		31 24
	4n+3	Four	First	8 bits	4n+3		7 0
			Second	8 bits	4n+4		15 8
			Third	8 bits	4n+5		23 16
			Fourth	8 bits	4n+6		31 24

(r1): Consecutive access (only when consecutive access is enabled by BE = 1 in SDAMOD during the EXDMAC block or cluster transfer in single address mode)

Figure 16.16 Data Alignment (Little Endian) in 8-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		DQM1 / DQM0 / WE# / Data Bus D15 D8 D7 D0
8 bits	4n	One	First	8 bits	4n	(r1)	7 0
	4n+1	One	First	8 bits	4n+1	(r1)	7 0
	4n+2	One	First	8 bits	4n+2	(r1)	7 0
	4n+3	One	First	8 bits	4n+3	(r1)	7 0
16 bits	4n	Two	First	8 bits	4n		15 8
			Second	8 bits	4n+1		7 0
	4n+1	Two	First	8 bits	4n+1		15 8
			Second	8 bits	4n+2		7 0
	4n+2	Two	First	8 bits	4n+2		15 8
			Second	8 bits	4n+3		7 0
	4n+3	Two	First	8 bits	4n+3		15 8
			Second	8 bits	4n+4		7 0
32 bits	4n	Four	First	8 bits	4n		31 24
			Second	8 bits	4n+1		23 16
			Third	8 bits	4n+2		15 8
			Fourth	8 bits	4n+3		7 0
	4n+1	Four	First	8 bits	4n+1		31 24
			Second	8 bits	4n+2		23 16
			Third	8 bits	4n+3		15 8
			Fourth	8 bits	4n+4		7 0
	4n+2	Four	First	8 bits	4n+2		31 24
			Second	8 bits	4n+3		23 16
			Third	8 bits	4n+4		15 8
			Fourth	8 bits	4n+5		7 0
	4n+3	Four	First	8 bits	4n+3		31 24
			Second	8 bits	4n+4		23 16
			Third	8 bits	4n+5		15 8
			Fourth	8 bits	4n+6		7 0

(r1): Consecutive access (only when consecutive access is enabled by BE = 1 in SDAMOD during the EXDMAC block or cluster transfer in single address mode)

Figure 16.17 Data Alignment (Big Endian) in 8-Bit Bus Space

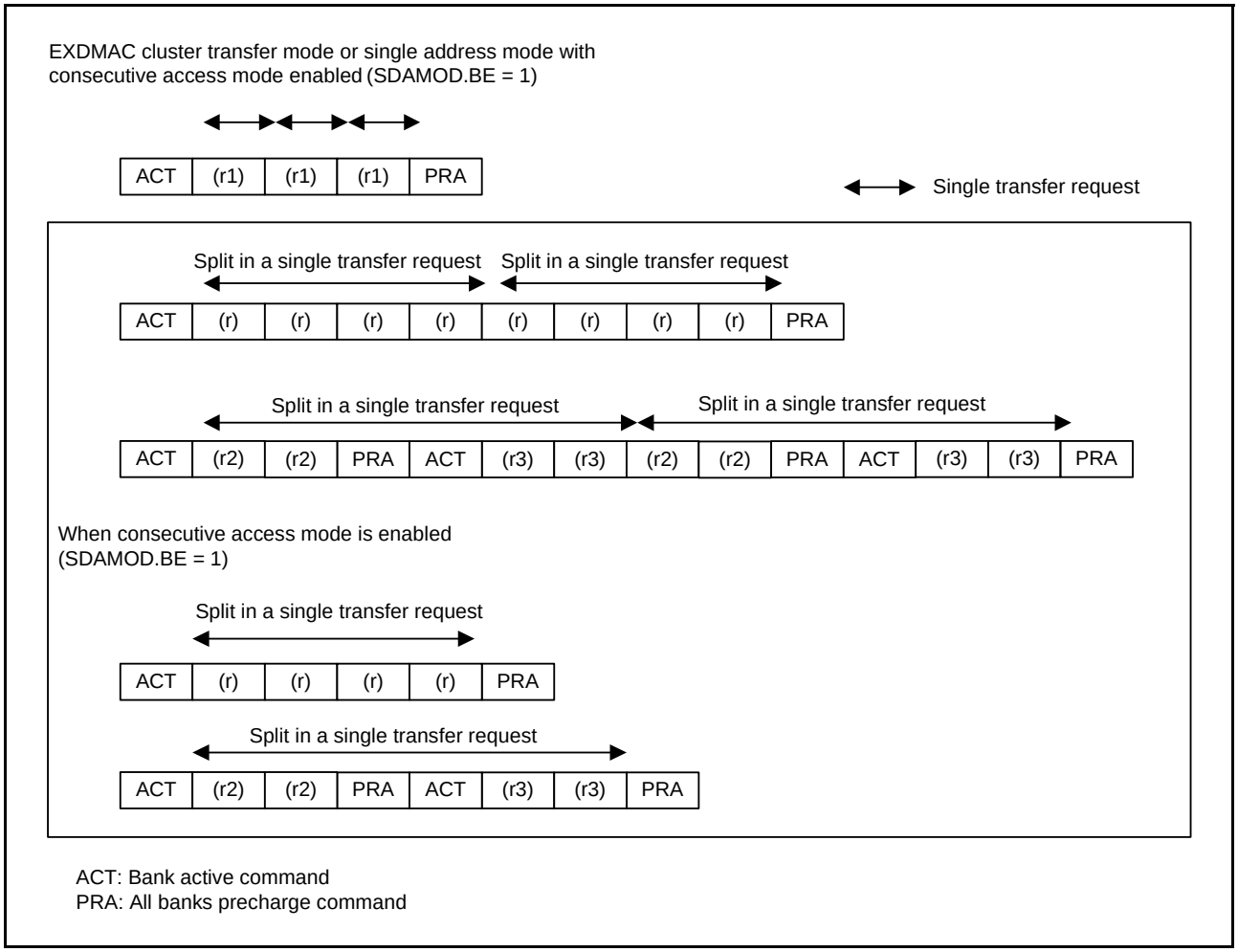


Figure 16.18 Consecutive Access Example

16.5 Operation of CS Area Controller

16.5.1 Separate Bus

The various periods in the timing charts are described below.
The CS area controller (CSC) operates in synchronization with the external bus clock (BCLK). The operation cycles, such as wait cycles specified with the CSC register, are counted on BCLK. In the following description, frequencies of BCLK and BCLK pin output are the same, unless otherwise noted.
Access via the external bus starts at the same point as the output of a rising edge on the BCLK pin. However, if the external bus clock (BCLK) and the output on the BCLK pin are at different frequencies so that a single request from a bus master for transfer leads to two or more rounds of access via the external bus, the wait settings may cause the start of access for the second and subsequent rounds to coincide with the falling edge of the output on the BCLK pin (see Figure 16.26 to Figure 16.30). If recovery cycles are inserted for bus access, the setting for the number of recovery cycles may also cause the start of access for the second and subsequent rounds to coincide with the falling edge of the output on the BCLK pin (see Figure 16.46).

(a) Tw1 to Twn (Clock Cycles of Waiting for a Normal Read Cycle or Normal Write Cycle)

The period Tw1 to Twn is made up of the number of clock cycles between the start of access via the external bus clock and one cycle before the strobe signal is valid. The number of cycles are selectable within the range from zero to 31. Within this period, the timing of CSn#, RD#, and WRn# assertion (placing the signals at the low level) is determined by the respective wait settings. Specifically, the periods of waiting are controlled by the CS assert wait select bits (CSON[2:0]), the RD assert wait select bits (RDON[2:0]), the WR assert wait select bits (WRON[2:0]), and the write data output wait select bits (WDON[2:0]) in CSn wait control register 2 (CSnWCR2). The number of clock cycles for each of these periods of waiting is selectable as a value from zero to seven counted from the start of external bus access. Selectable numbers of cycles are also within the overall number of clock cycles of waiting for reading or writing.

(b) Tend (Clock Cycle where the Strobe Signal is Valid)

Tend is the next clock cycle after completion of the period of waiting for a normal cycle of reading or writing or for a cycle of page reading or page writing. If each wait select bit for a normal cycle of reading or writing or for a cycle of page reading or page writing is zero, the clock cycle where bus access starts is the clock cycle where the strobe signal is valid. The RD# and WRn# signals are negated in the next clock cycle after the cycle where the strobe signal is valid. In the case of read access, the clock cycle where the strobe signal is valid becomes the clock cycle where the data to be read are sampled.
If an external wait is enabled, the wait signal is sampled at the time of the cycle where the strobe signal is valid. The bus cycle is extended if the wait signal is at the low level. The bus cycle is completed in the next clock cycle if the wait signal is at the high level. Tend indicates the cycle where sampling of the wait signal starts.
After the first cycle where the strobe signal is valid during page access, second and subsequent page access operations (point (e) below) start in the next cycle except in cases of write access where a setting (other than zero) for write-data output extension clock cycles (point (d) below) has been made. If the setting for the RD or WR assertion wait is a value other than zero, the RD# and WRn# signals are negated in the next clock cycle. If the setting is zero, assertion continues. Furthermore, the CSn# signal continues to be asserted rather than being negated.

(c) Tn1 to Tnm (Clock Cycles of CS Extension)

In the case of normal access, Tn1 to Tnm represent the clock cycles of the period following the cycle where the strobe signal is valid (Tend) up to negation of the CSn# signal. For read or write access, the timing of negation can be controlled by the read-access CS extension cycle select bits (CSROFF[2:0]) and the write-access CS extension cycle select bits (CSWOFF[2:0]) in the CSn wait control register 2 (CSnWCR2), respectively.
The number of cycles are counted from the cycle following the cycle where the strobe signal is valid.
In the case of page access, Tn1 to Tnm represent the clock cycles of the period for the cycle following the last cycle where the strobe signal is valid up to negation of the CSn# signal.

For write access, setting the write data output extension cycle select bits (WDOFF) controls extension of the period where the address and output data are valid.

(d) Tdw1 to Tdwn (Write-Data Output Extension Clock Cycles)

For write access, if the setting for write-data output extension wait is a value other than zero, clock cycles of write-data output extension are inserted from the cycle that follows the cycle where the strobe signal is valid (Tend).
In the case of normal access, this is inserted within the period of clock cycles of CS extension (point (c) above).
In the case of page access, this is inserted within the period of the cycle where the strobe signal is valid and subsequent page access or within the period of clock cycles of CS extension (point (c) above). Valid address and data output are extended over this period, and the WRn# signal is negated.

(e) Tpw1 to Tpwn (Page-Read Cycle Wait or Page-Write Cycle Wait)

For the second and subsequent bus cycles during page access, the values for a page-read cycle wait or page-write cycle wait are used instead of the settings for a normal read or write cycle wait. Setting the WR assert wait select bits becomes enabled in the same way as for the first round of access. How the setting for RD assertion controls operation depends on the setting for page-read access mode (the PRMOD bit in CSnMOD) as described below.

CSnMOD.PRMOD = 0: A wait until RD assertion is inserted in the same way as for the first round of access, and the RD# signal is negated.
CSnMOD.PRMOD = 1: Although a wait until RD assertion is inserted in the same way as for normal-access compatibility mode, the RD# signal continues to be asserted over this period.

(f) Tr1 to Trn (Recovery Cycles)

Recovery cycles can be inserted from the point where a bus cycle is completed (CSn# signal negation). The number of recovery cycles can be controlled by the setting of the read recovery (RRCV) or write recovery (WRCV) bits in the CSn recovery cycle register (CSnREC). Both numbers of recovery cycles are counted from the end of a bus cycle (CSn# negation) and can be selected from 0 to 15 cycles. For details on recovery cycles, see section 16.5.4, Insertion of Recovery Cycles.

(1) Normal Access

When the PRENB and PWENB bits in CSnMOD are set to 0 to disable page-read and page-write access, respectively, all bus accesses will take the form of normal read and write operations.
Even when the PRENB and PWENB bits in CSnMOD are set to 1 to enable page-read and page-write access, respectively, bus access other than page access will take the form of normal read and write operations.
Figure 16.19 to Figure 16.21 show the normal access operations.

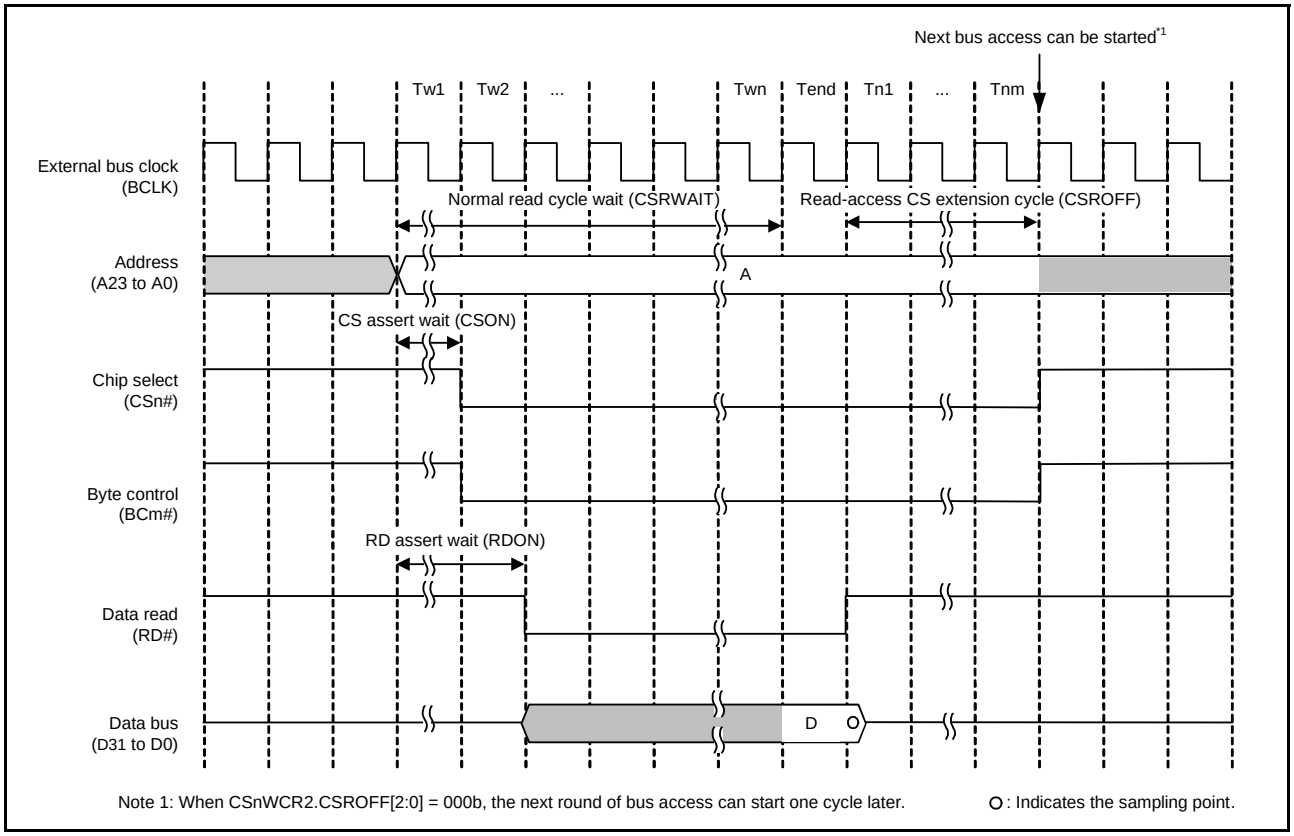


Figure 16.19 Bus Timing (Normal-Read Operation) (n = 0 to 7, m = 0 to 3)

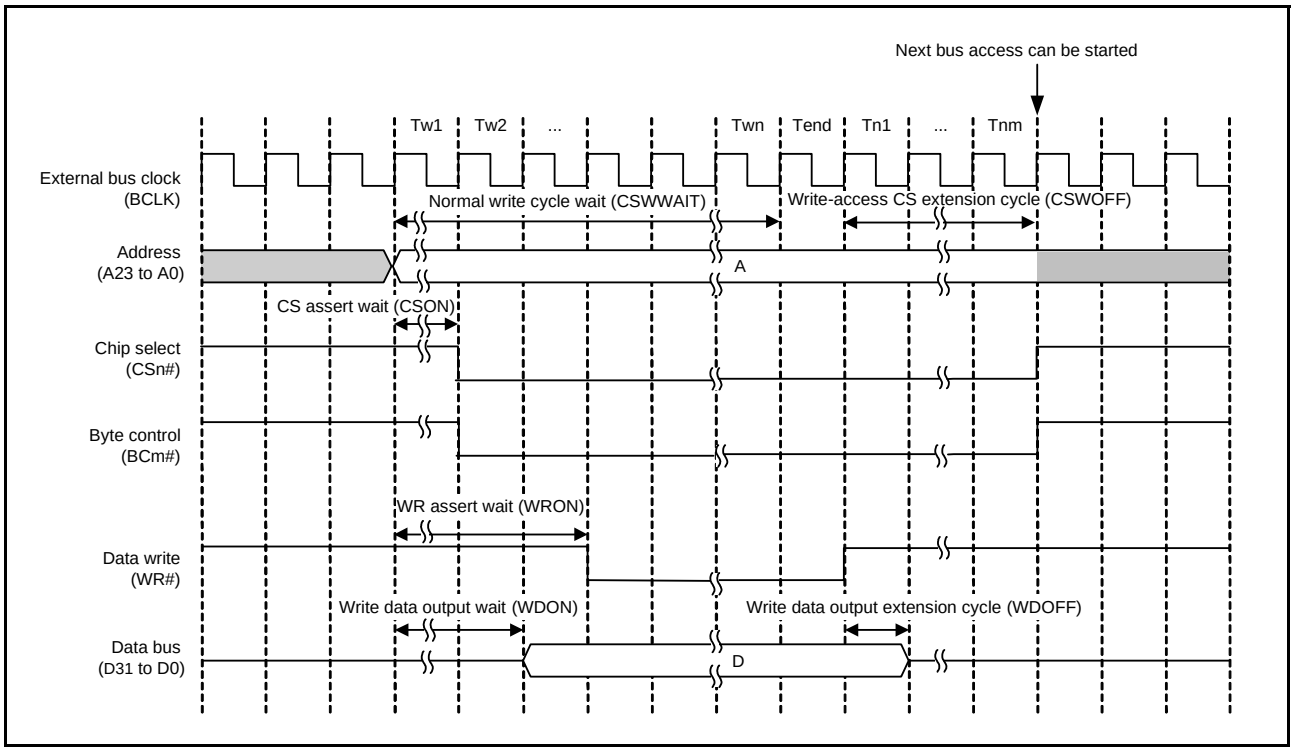


Figure 16.20 Bus Timing (Normal-Write Operation, Single Write Strobe Mode) (n = 0 to 7, m = 0 to 3)

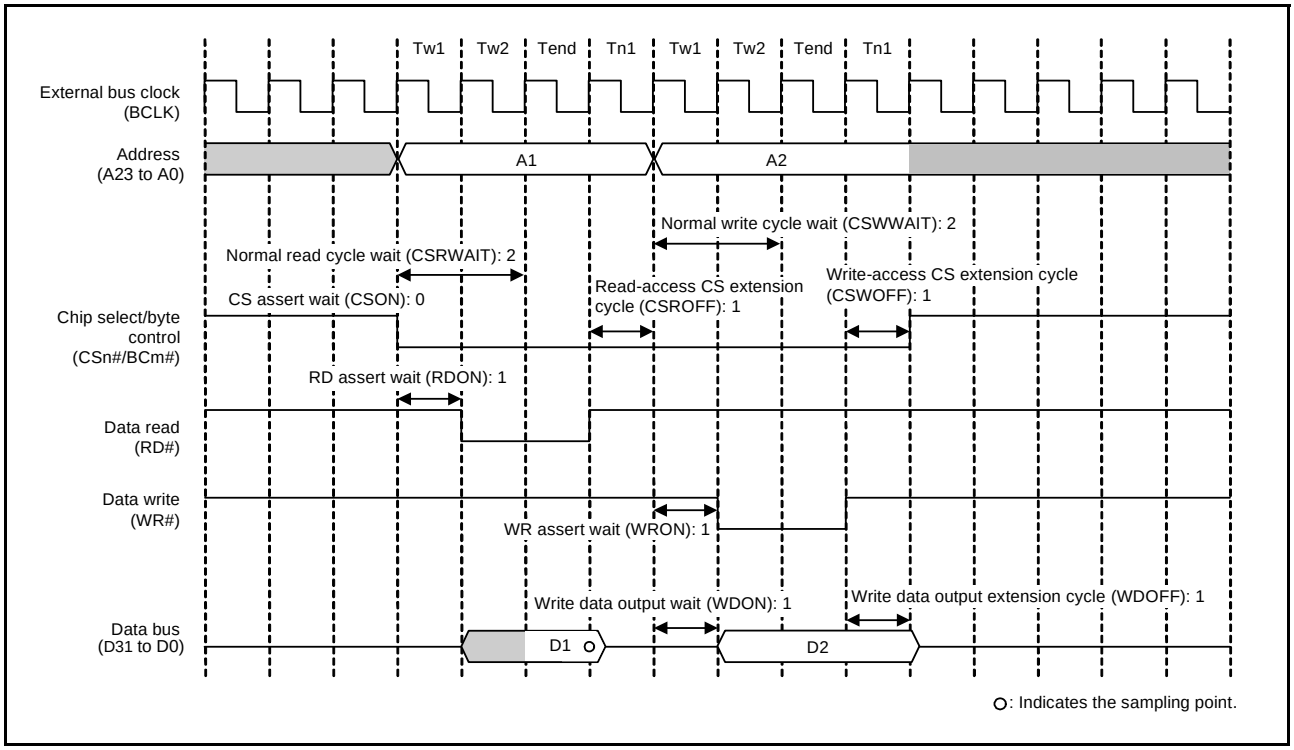


Figure 16.21 Example of Normal Access Operation (Read/Write) (n = 0 to 7, m = 0 to 3)

When two or more rounds of external bus access are required in response to a single request for transfer from a bus master, normal access operations (steps (a) to (d) above) are repeated. Figure 16.22 and Figure 16.23 show examples of operations when two rounds of bus access are generated in response to a single request for transfer. If the recovery cycle insertion condition is satisfied, recovery cycles (step (f) above) are also inserted in the second and subsequent external bus accesses (see Figure 16.44).

The values of the wait control registers are example settings. In practice, set the register bits according to the specifications of connected devices.

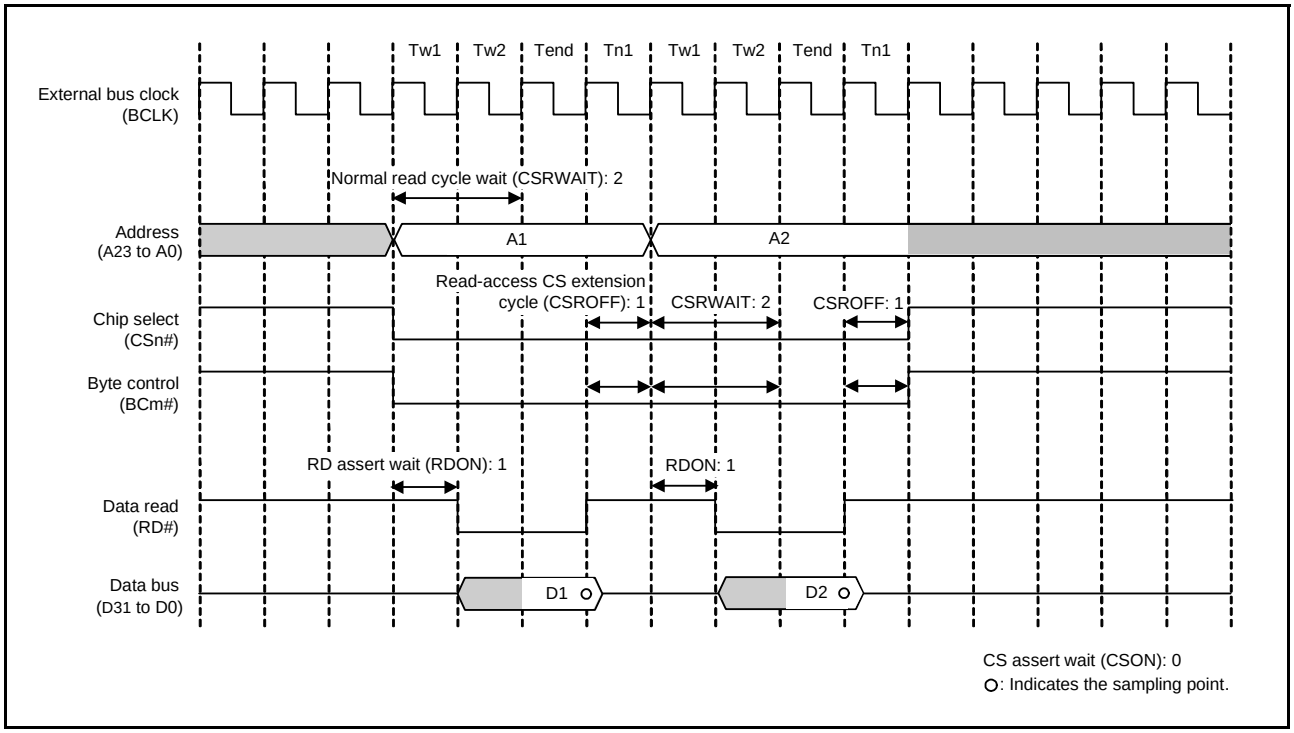


Figure 16.22 Example of Normal-Read Operation (when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7, m = 0 to 3)

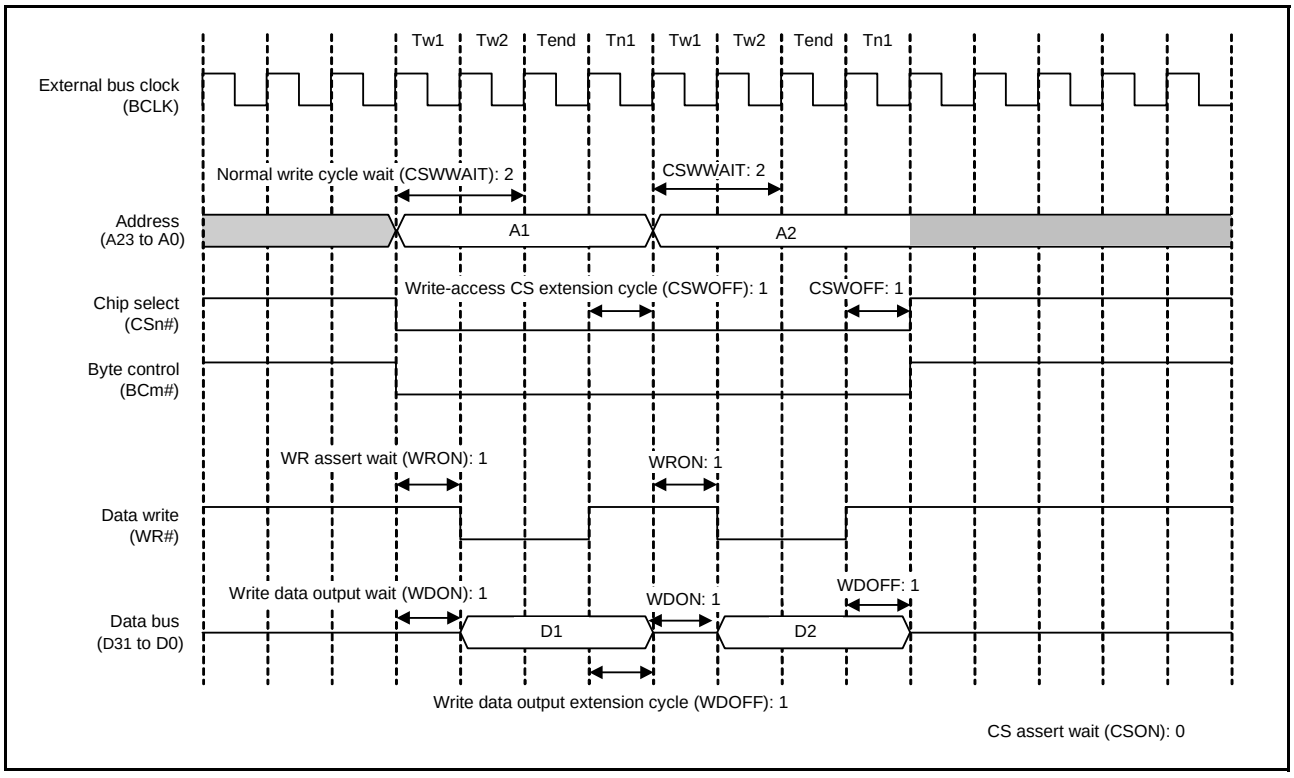


Figure 16.23 Example of Normal-Write Operation (when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer, in Single Write Strobe Mode) (n = 0 to 7, m = 0 to 3)

Figure 16.24 and Figure 16.25 show examples of normal read and write accesses to a 32-bit bus space in 16 bits.

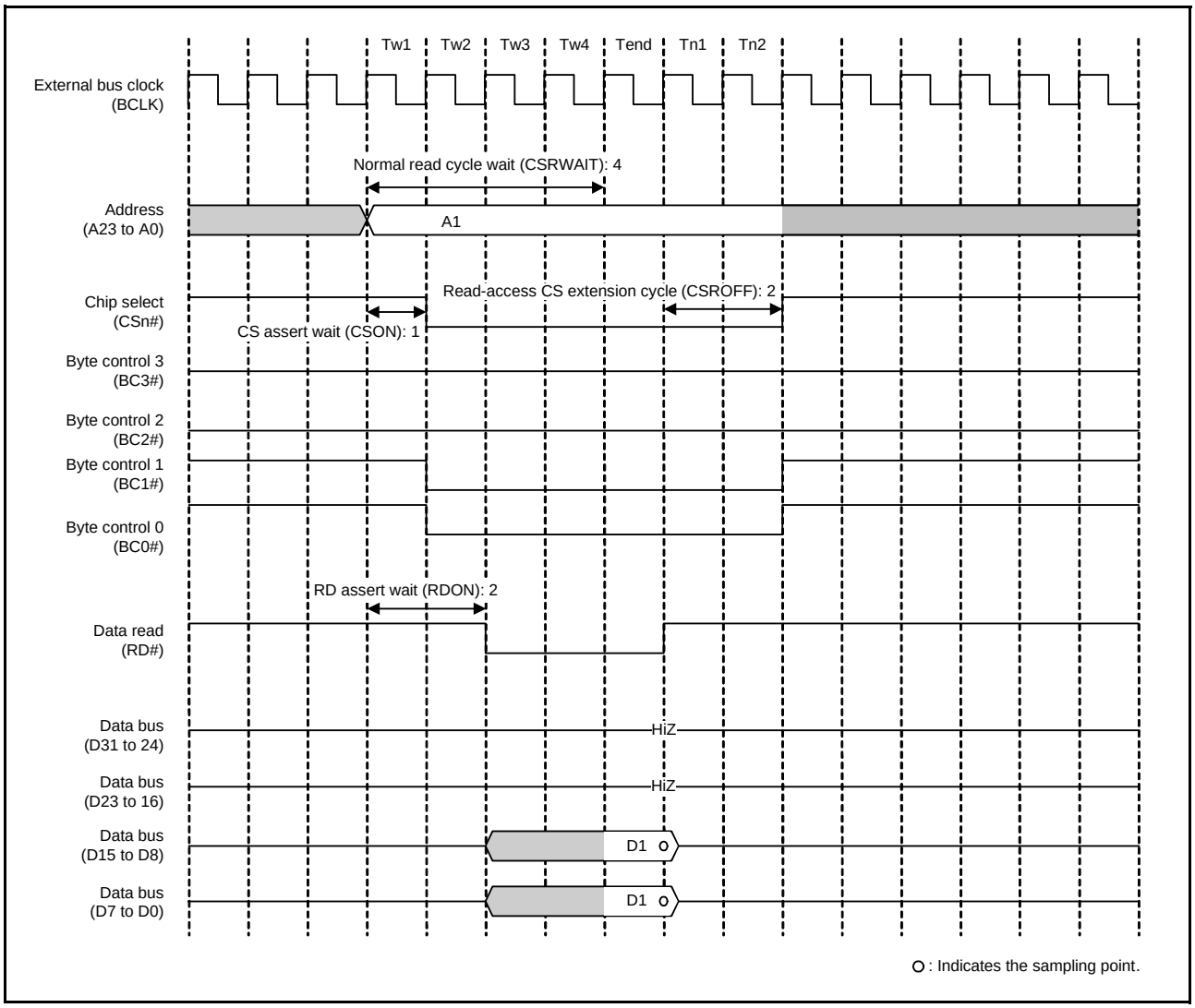


Figure 16.24 Example of Normal-Read Operation (when 32-Bit Bus Space is Accessed in 16 Bits) (n = 0 to 7)

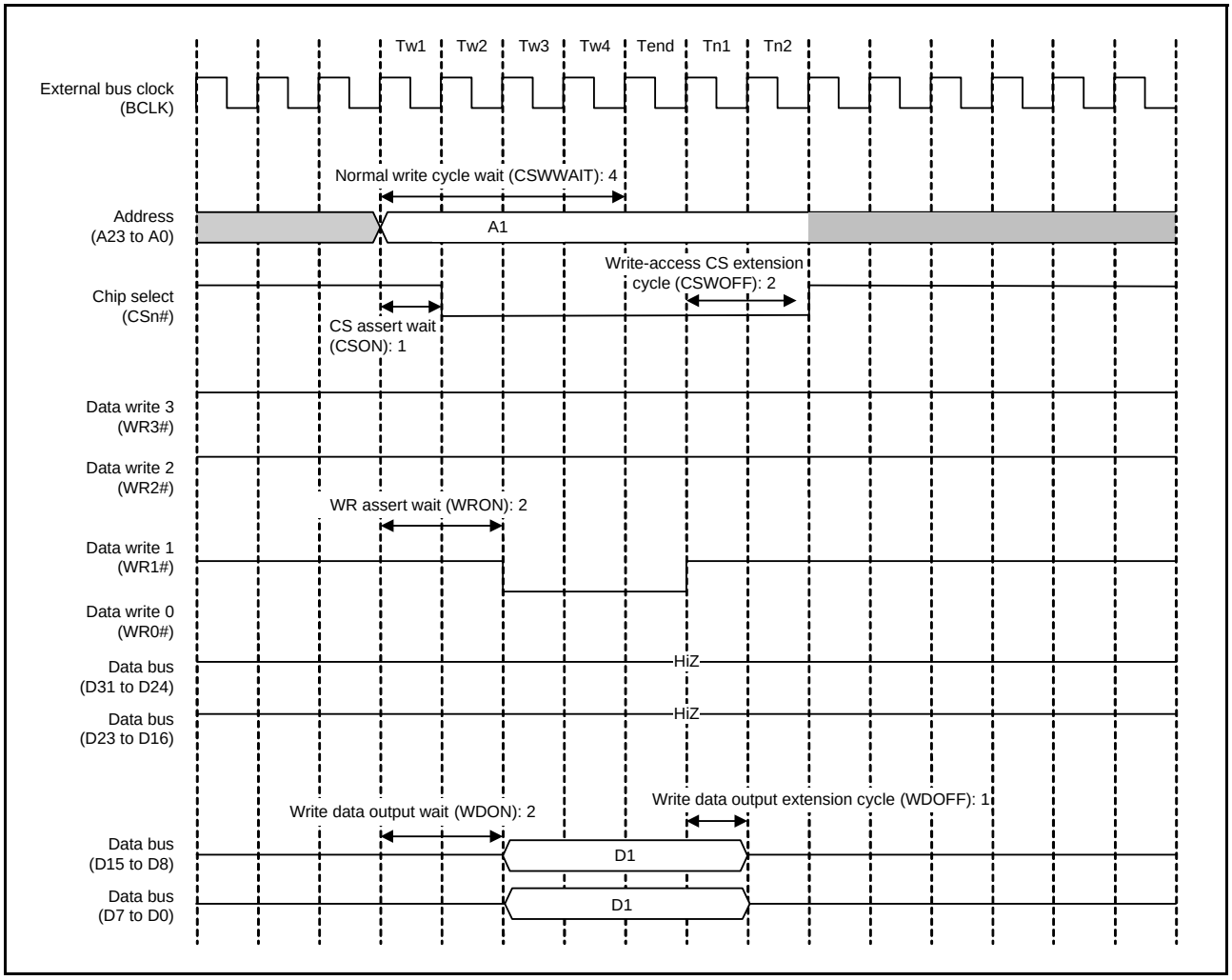


Figure 16.25 Example of Normal-Write Operation (when 32-Bit Bus Space is Accessed in 16 Bits, in Byte Strobe Mode) (n = 0 to 7)

Figure 16.26 to Figure 16.30 show examples of normal accesses made with the 1/2 BCLK selected with the BCLK pin output select bit.

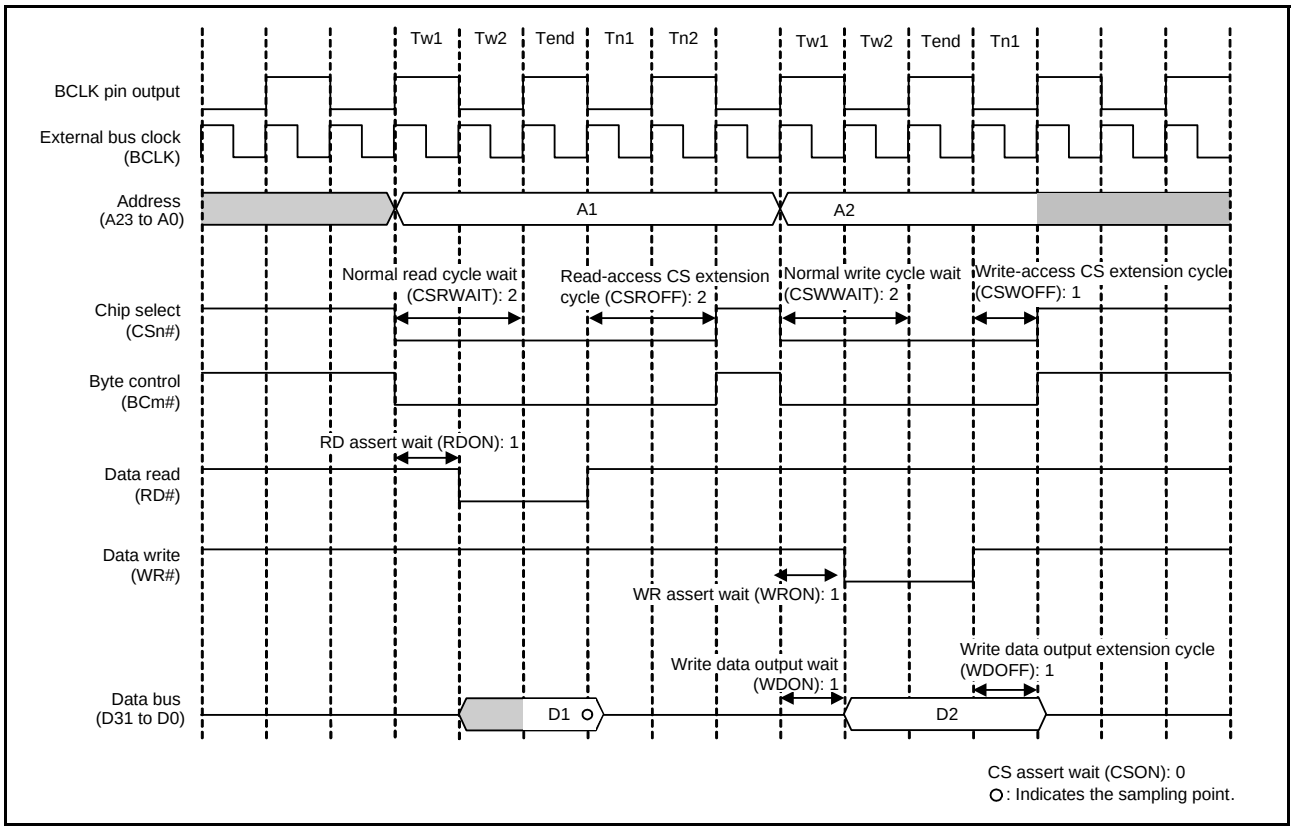


Figure 16.26 Example of Normal Access (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit) (n = 0 to 7, m = 0 to 3)

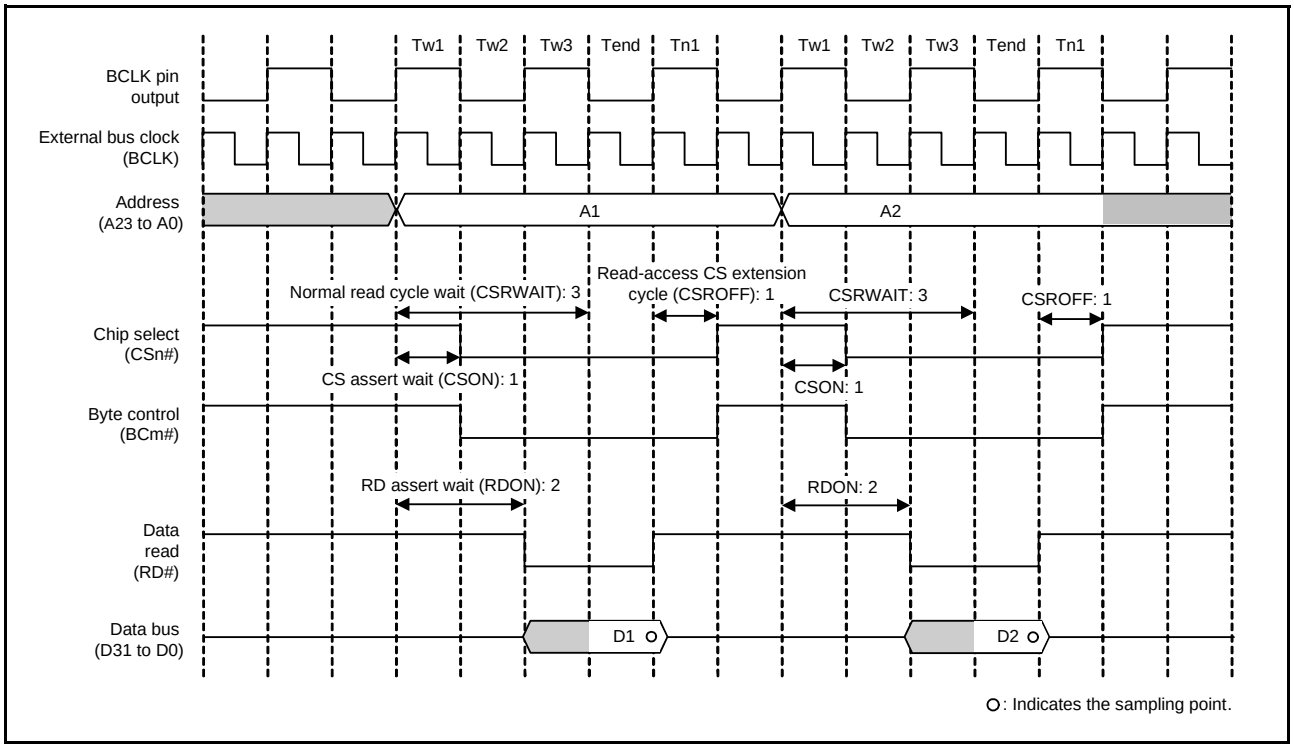


Figure 16.27 Example of Normal-Read Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit) (n = 0 to 7, m = 0 to 3)

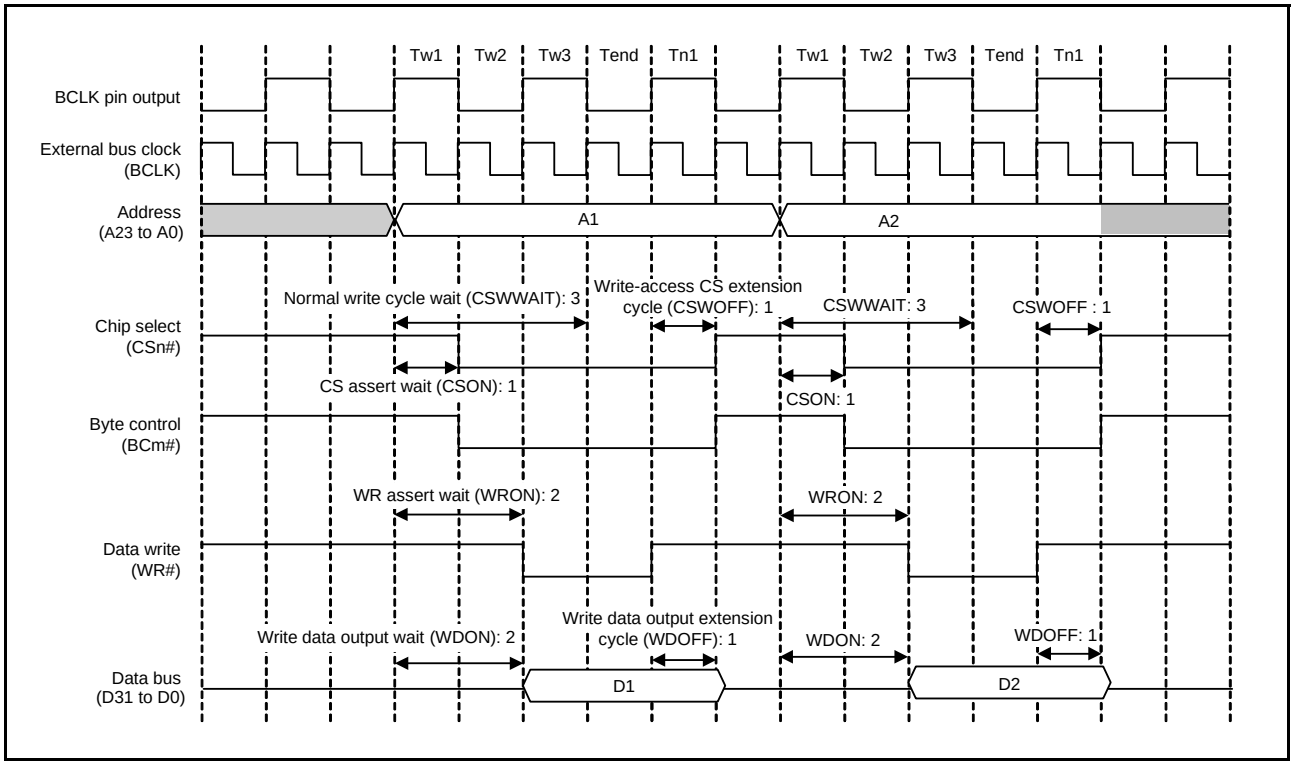


Figure 16.28 Example of Normal-Write Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit) (n = 0 to 7, m = 0 to 3)

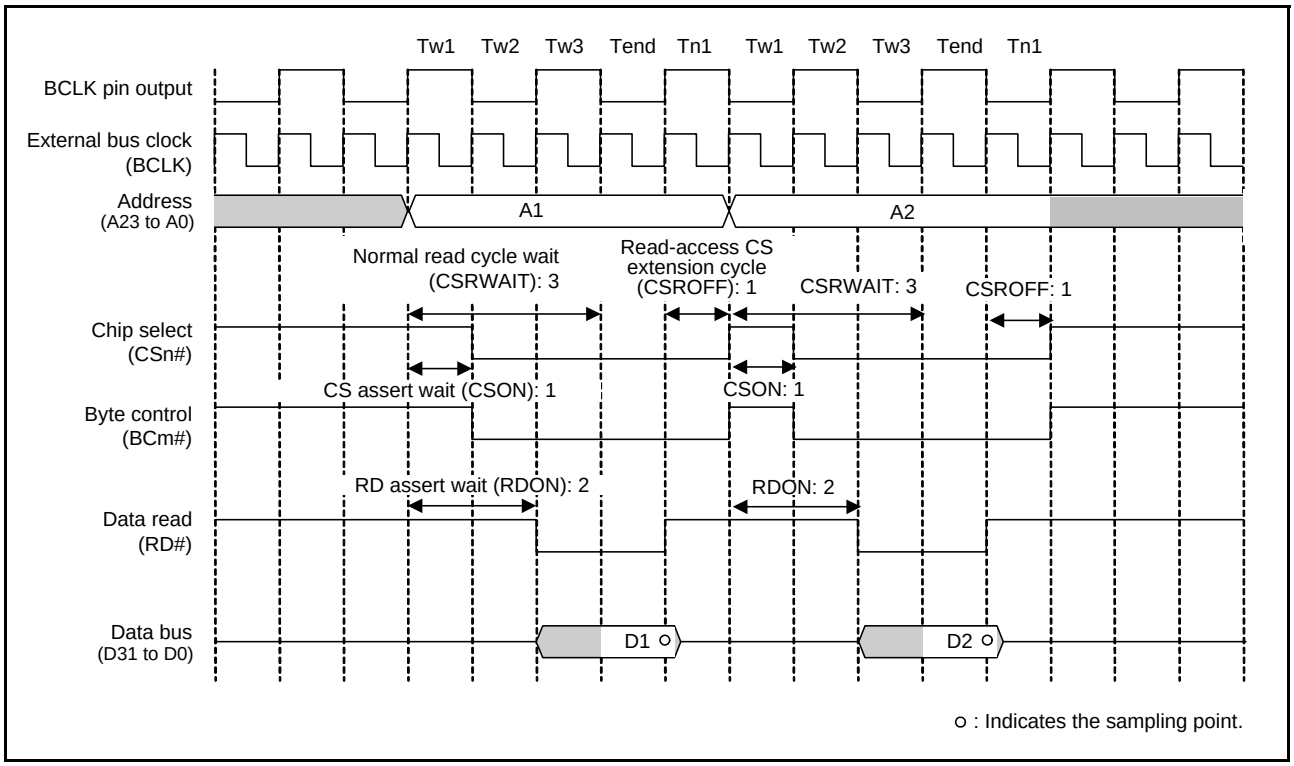


Figure 16.29 Example of Normal-Read Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7, m = 0 to 3)

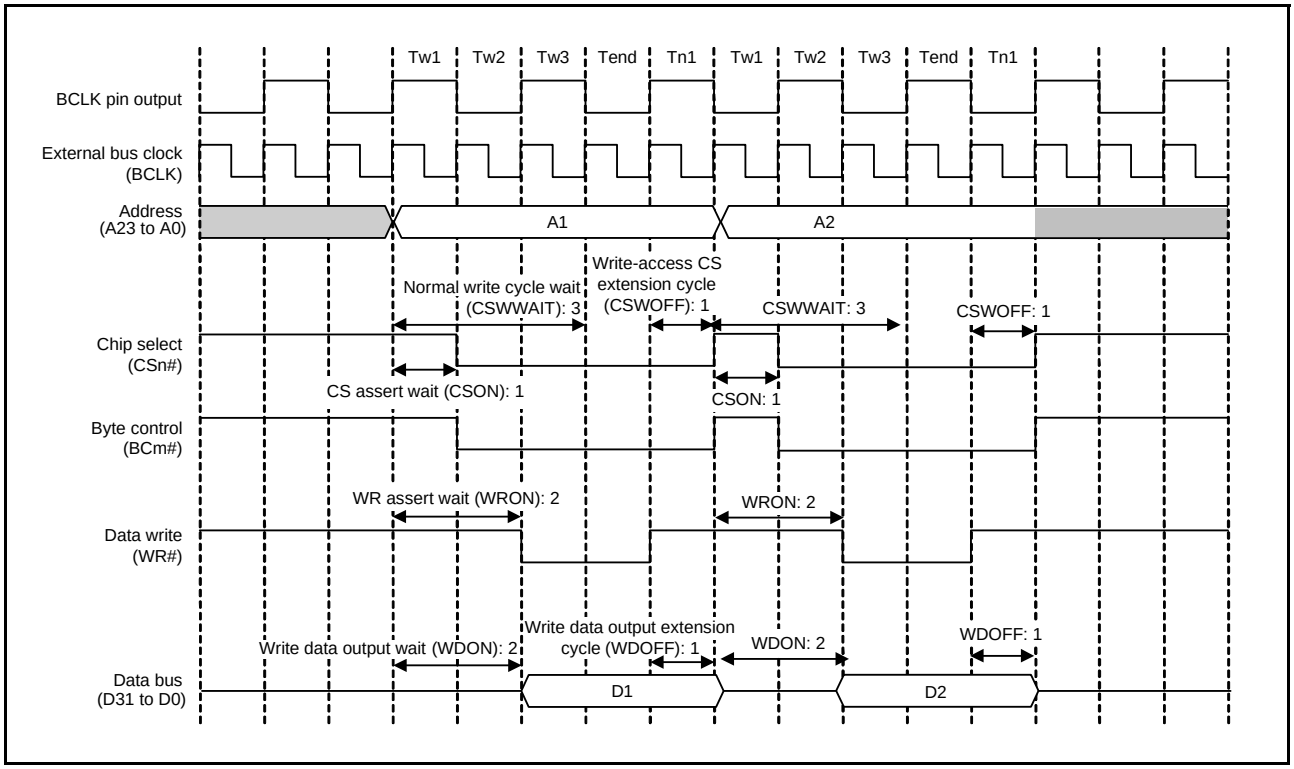


Figure 16.30 Example of Normal-Write Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7, m = 0 to 3)

(2) Page Access

When the PRENB and PWENB bits in CSnMOD are set to 1 to enable page-read and page-write access, respectively, the bus access for page access operations becomes page reading and writing. Specifically, page access can occur when two or more rounds of external bus access are required for a single transfer request from the bus master. However, normal access is made when the split accesses are not aligned or the access spreads over the 32-bit boundary. See Figure 16.8 to Figure 16.11 for the conditions under which page access occurs.

Figure 16.31 and Figure 16.32 show examples of page access operations.

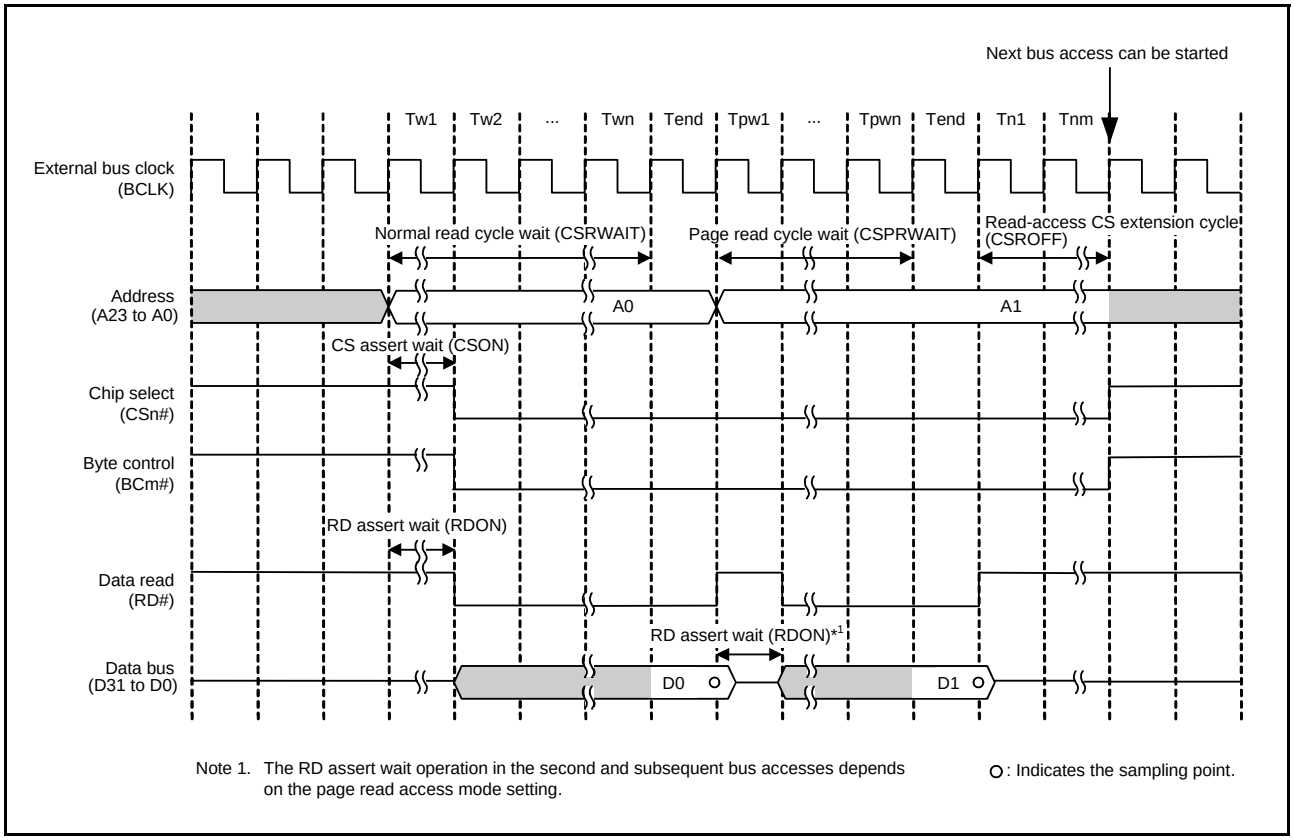


Figure 16.31 Page-Read Access Timing (n = 0 to 7, m = 0 to 3)

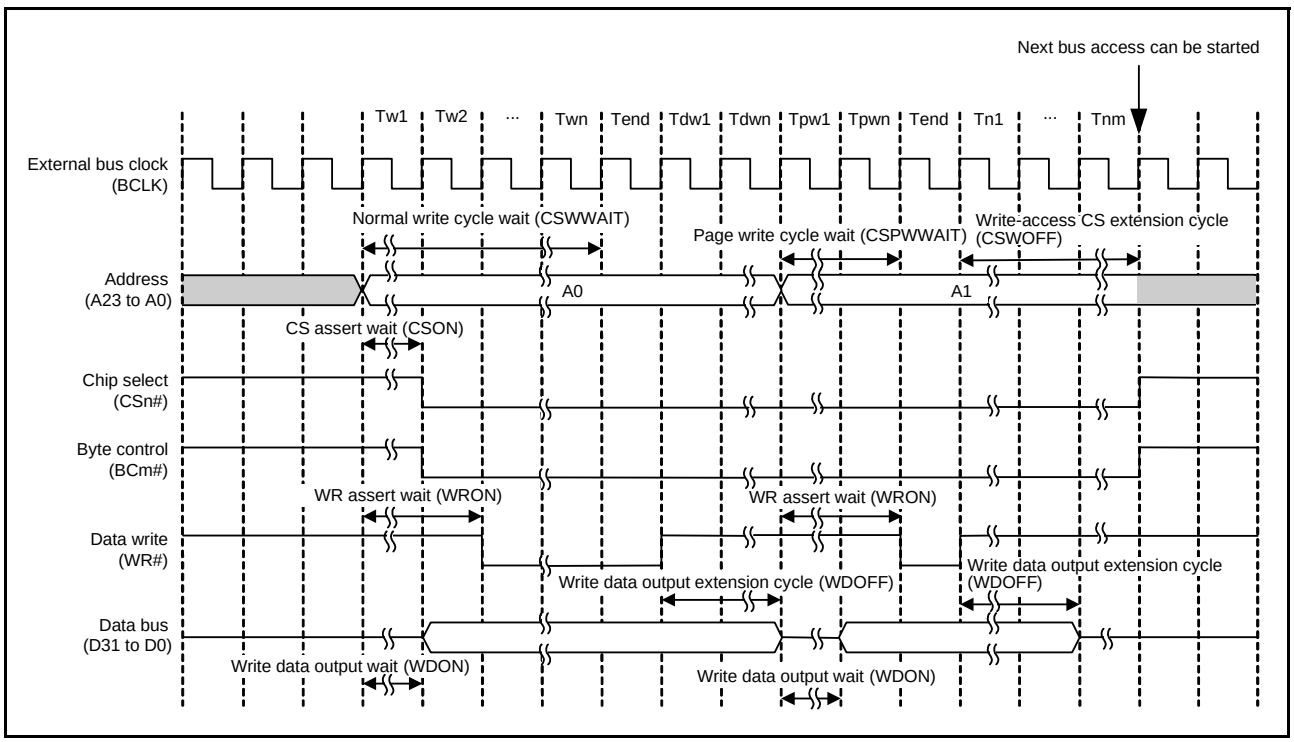


Figure 16.32 Page-Write Access Timing (n = 0 to 7, m = 0 to 3)

Figure 16.33 and Figure 16.34 show examples of operations for access to a 16-bit bus space in 32 bits. The values of the wait control registers are example settings. In practice, the register settings will correspond to the specifications of connected devices.

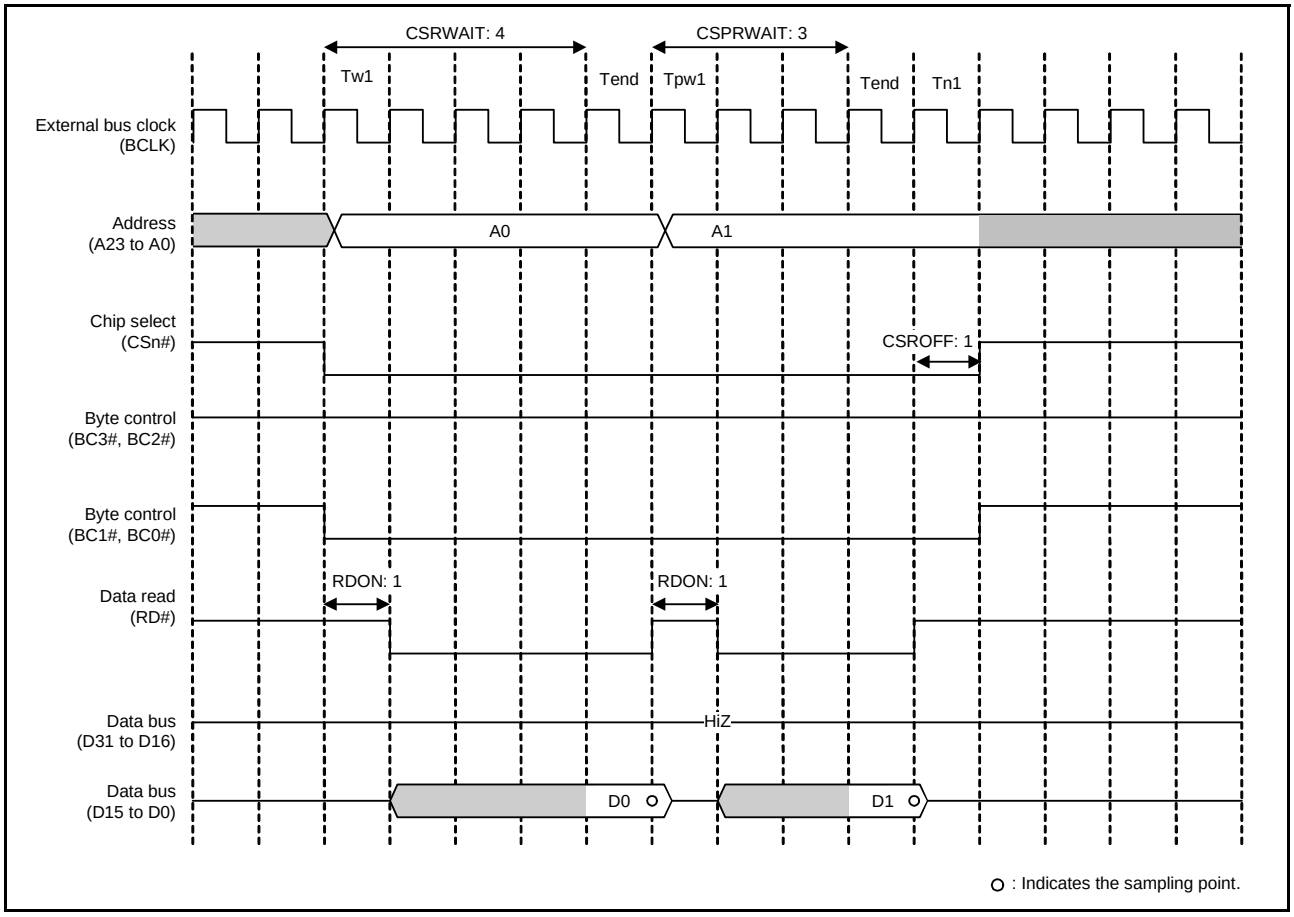


Figure 16.33 Example of Page-Read Access Operation (when 16-Bit Bus Space is Accessed in 32 Bits) (n = 0 to 7)

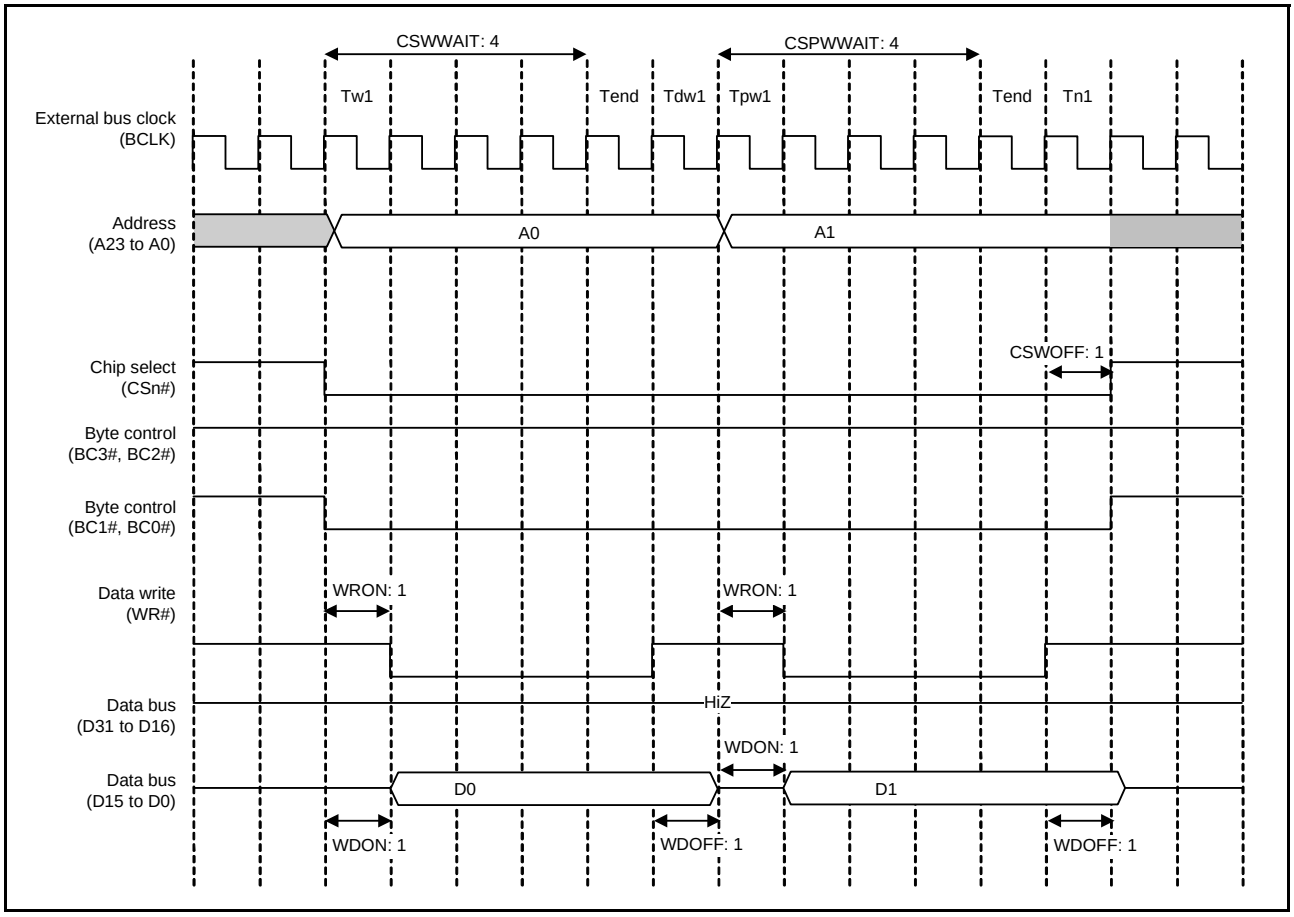


Figure 16.34 Example of Page-Write Access Operation (when 16-Bit Bus Space is Accessed in 32 Bits, in Single Write Strobe Mode) (n = 0 to 7)

Figure 16.35 and Figure 16.36 show examples of page access operations performed with the 1/2 BCLK selected with the BCLK pin output select bit.

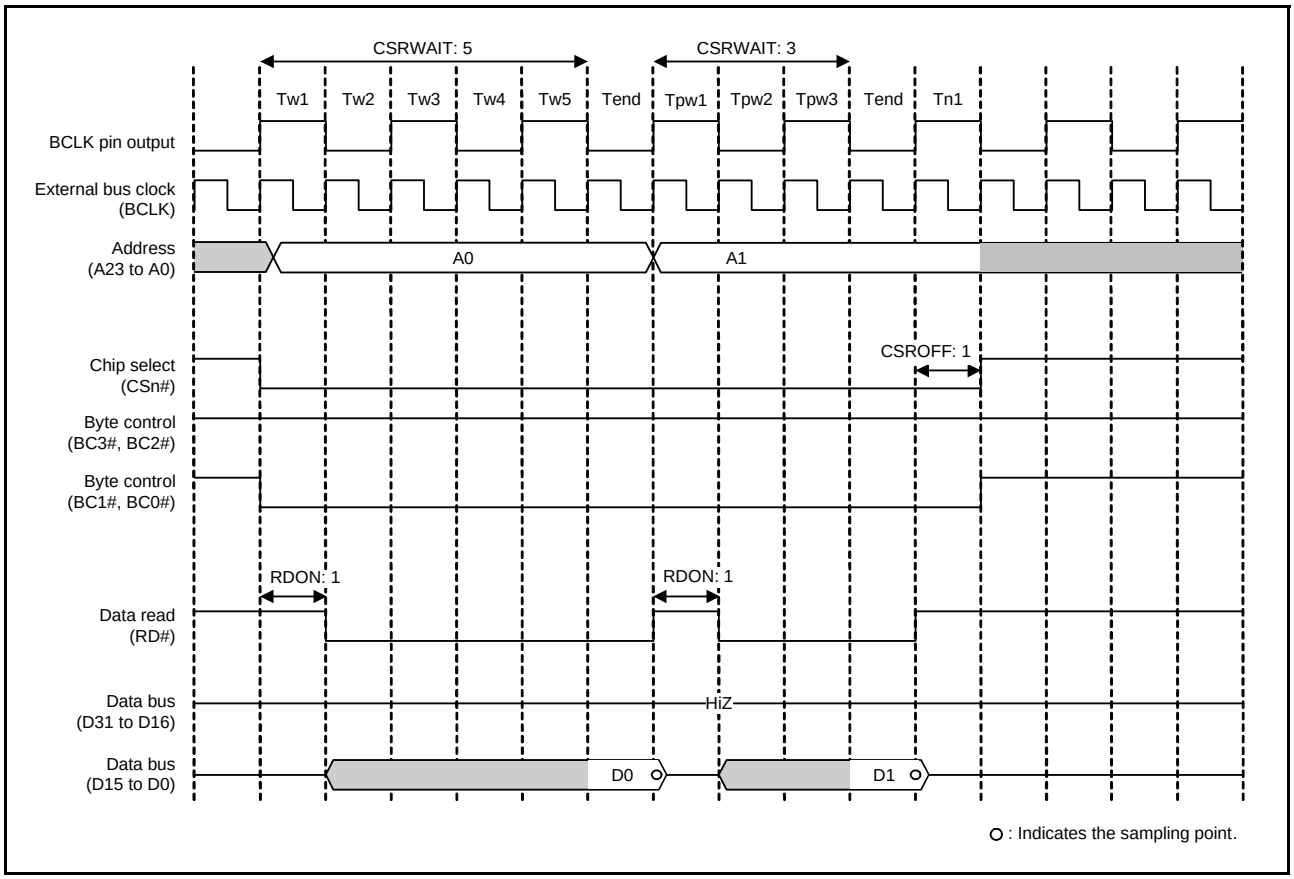


Figure 16.35 Example of Page Read Access Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7)

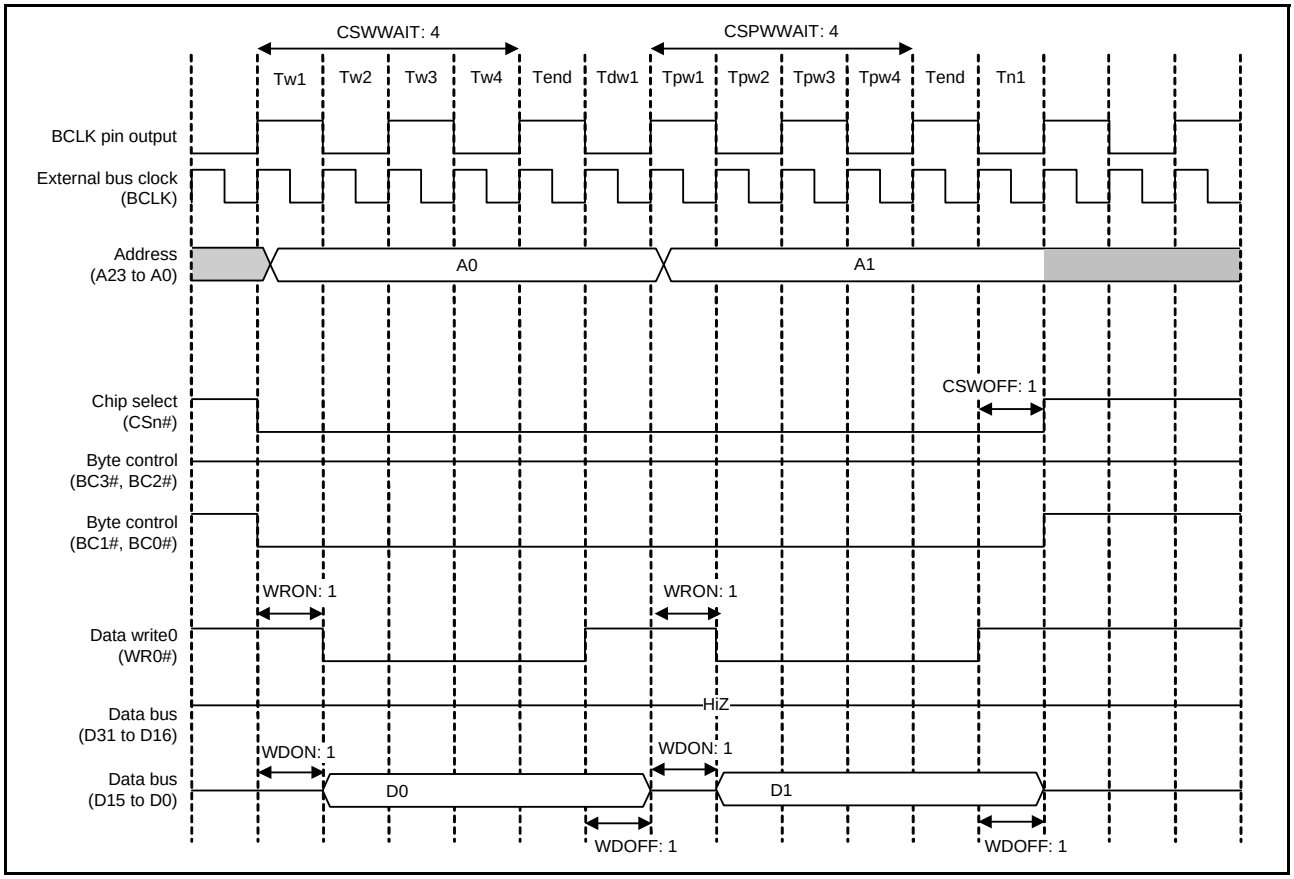


Figure 16.36 Example of Page Write Access Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer, in Single Write Strobe Mode) (n = 0 to 7)

16.5.2 Address/Data Multiplexed Bus

When the address/data multiplexed I/O interface select bit (MPXEN) in CSnCR is set to 1, addresses and data can be multiplexing input/output to/from the D15 to D0 pins in the corresponding area. Using this function enables direct connection of this LSI to peripheral LSIs requiring address/data multiplexing. When 8-bit width is selected with the BSIZE[1:0] bits in CSnCR, D7 to D0 are multiplexed with A7 to A0. When 16-bit width is selected, D15 to D0 are multiplexed with A15 to A0. In the address/data multiplexed I/O space, accesses are controlled with the ALE, RD#, WRn#, and BCn# signals.
Byte strobe mode or single-write strobe mode is selectable in the same way as for a separate bus. However, with regard to the BCn# signals within the address cycle, the byte-control signal is output for the data being read or written.
During the address/data multiplexed I/O space access, after the number of wait cycles specified by the address cycle wait select bits (AWAIT[1:0]) in CSnWCR2 is inserted in the address output cycle, data access is performed.

- Ta1 to Tan (Address Cycle Wait)

The period Ta1 to Tan is valid only when the address/data multiplexed I/O space is specified. This period is made up of the number of clock cycles between the start of external bus access and one cycle before the address latch (ALE) signal is negated. The number of cycles are selectable within the range from zero to three. Addresses are output until the next cycle of ALE signal negation (address cycle). The timing of ALE signal is the same as that of CS# assertion. After the address cycle, a data cycle is started. CSnWCR1 and CSnWCR2 should be set so that an address cycle and a data cycle do not overlap.

Page access to the address/data multiplexed I/O space is invalid. When the PRENB or PWENB bit in CSnMOD is set to 1 to enable page-read or page-write access, these settings are ignored and normal read or write operation is performed.

Figure 16.37 to Figure 16.39 show examples of operations with the address/data multiplexed I/O interface.

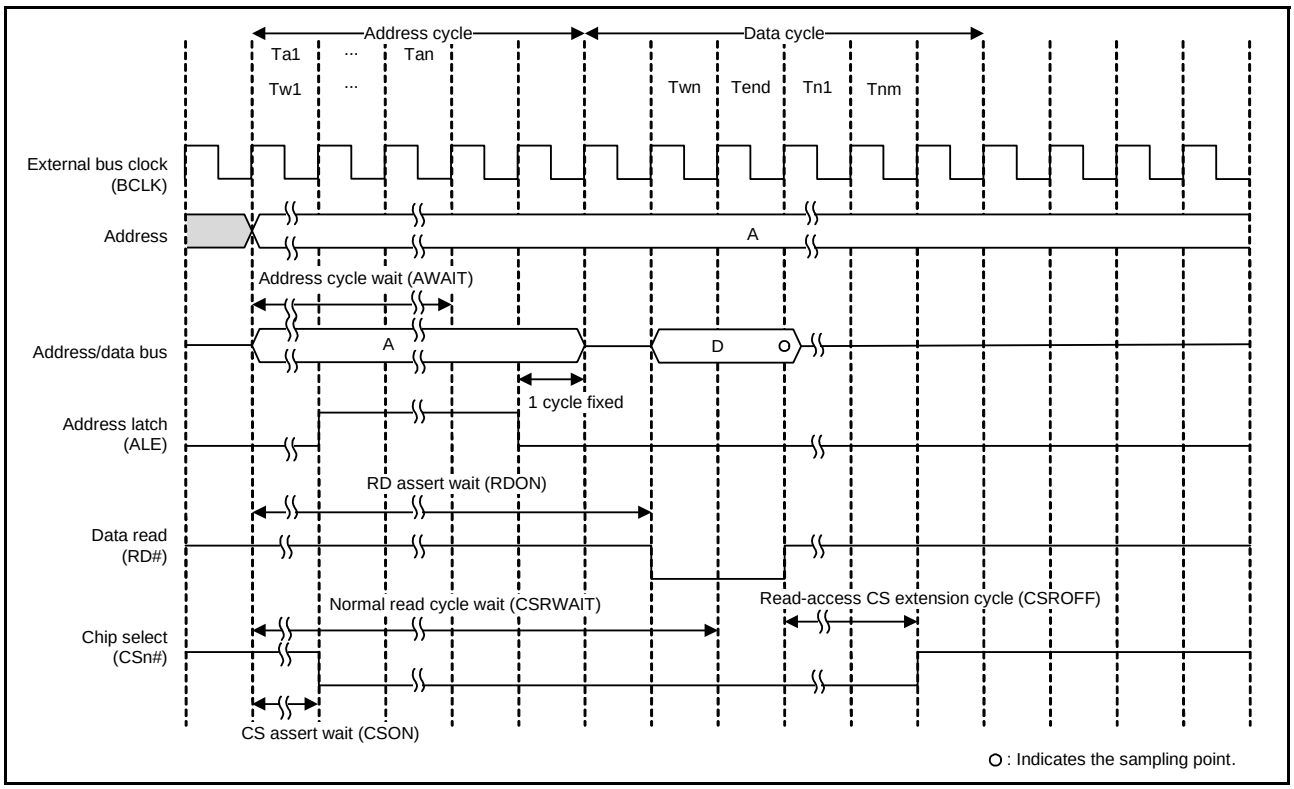


Figure 16.37 Example of Read Access Operation with Address/Data Multiplexed I/O Interface

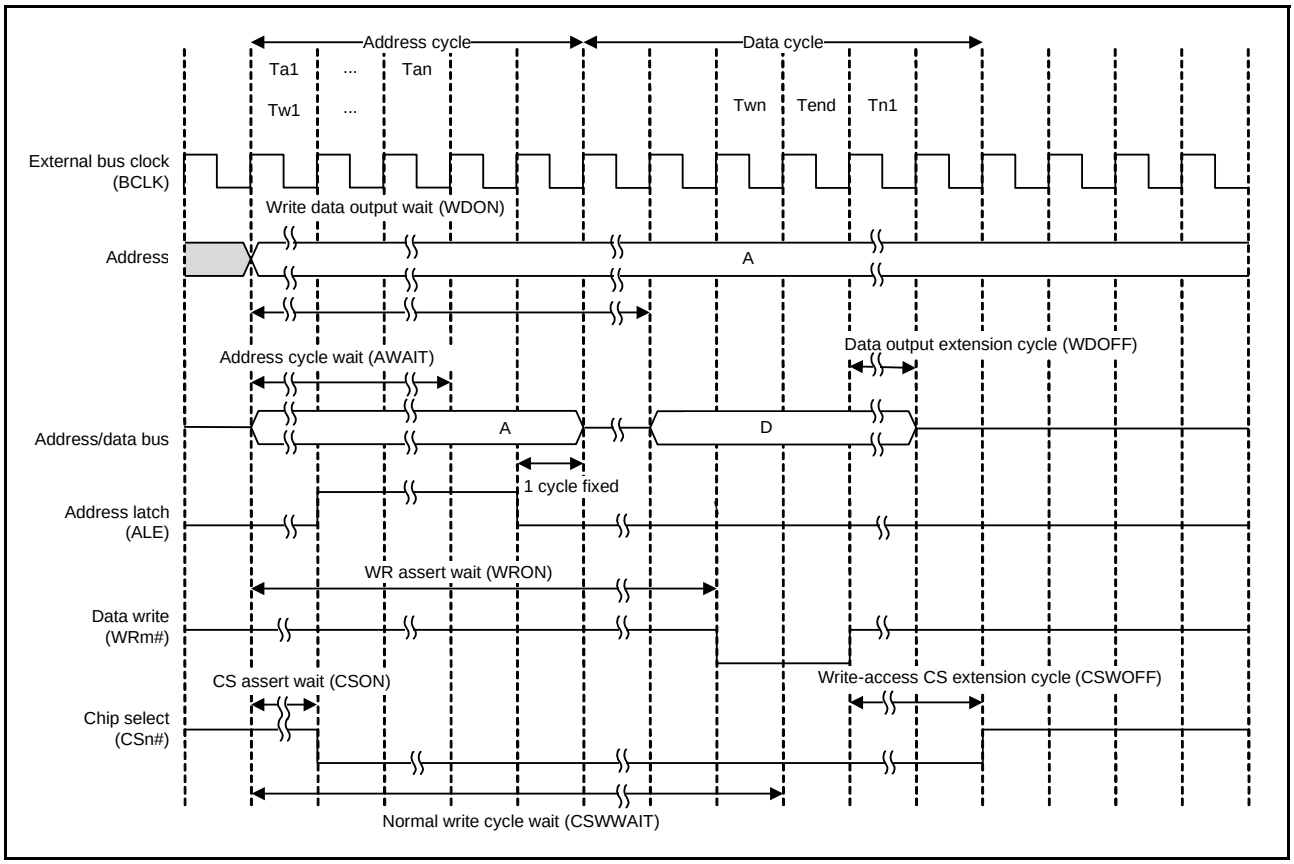


Figure 16.38 Example of Write Access Operation with Address/Data Multiplexed I/O Interface (m = 0, 1)

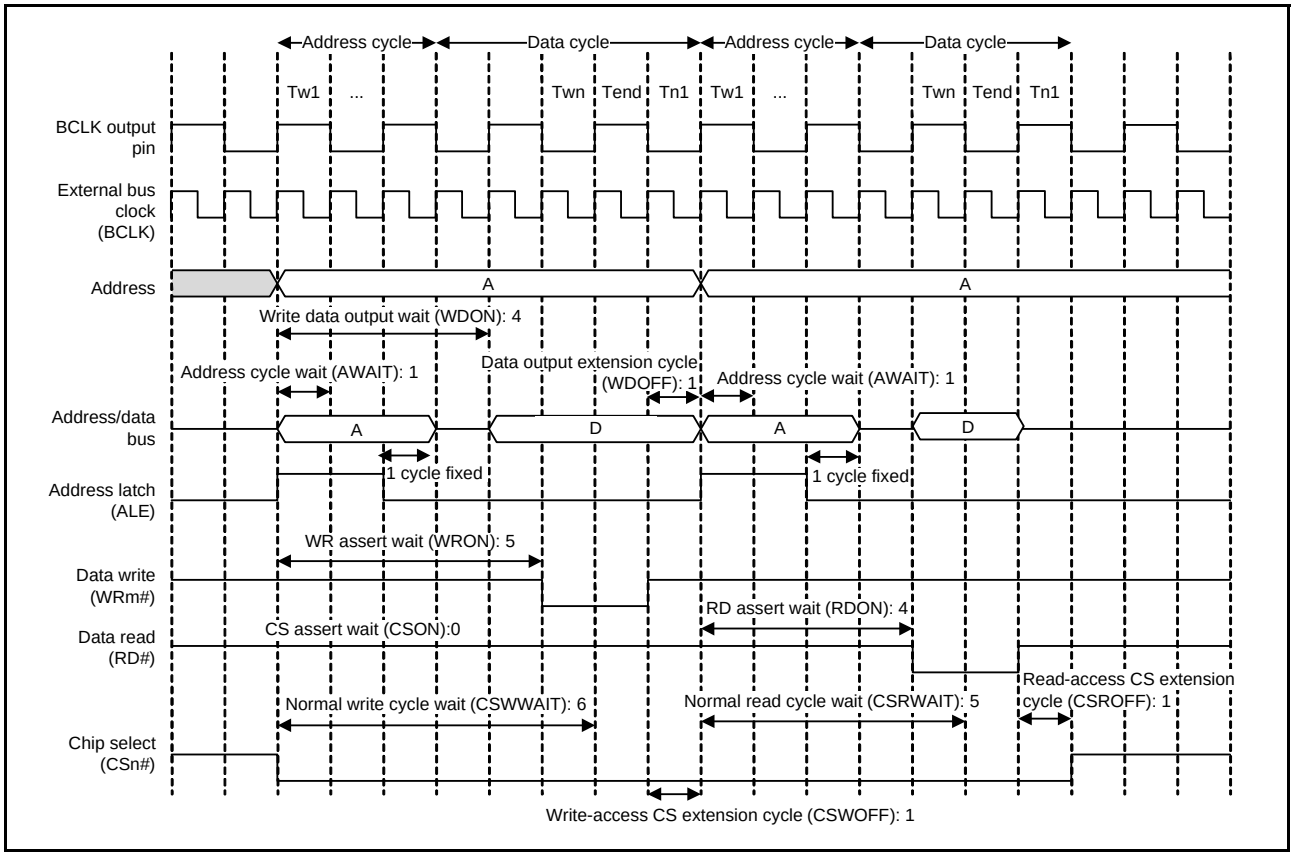


Figure 16.39 Example of Bus Timing with Address/Data Multiplexed I/O Interface (m = 0, 1)

16.5.3 External Wait Function

Wait cycles can be extended by the WAIT# signal over the length of normal access cycle wait (specified by the CSRWAIT[4:0] and CSWWAIT[4:0] bits in CSnWCR1) and page access cycle wait (specified by the CSPRWAIT[2:0] and CSPWWAIT[2:0] bits in CSnWCR1).
When external wait is enabled (the EWENB bit = 1 in CSnMOD), wait cycles are inserted while the WAIT# signal is held low. When external wait is disabled (the EWENB bit = 0 in CSnMOD), the WAIT# signal has no effect.
All wait cycles specified in CSnWCR1 are inserted independently of the WAIT# signal.

(1) Normal Access

Sampling of the WAIT# signal begins upon completion of the wait cycle (Tend) specified in CSnWCR1. The bus cycle is extended while the WAIT# signal is held low. The wait cycle ends (Tend) at the next cycle after the WAIT# signal becomes high.

(2) Page Access

The first access operation is the same as the normal access operation. Sampling of the WAIT# signal begins upon completion of the wait cycle (Tend) specified in the CSnWCR1 register. The bus cycle is extended while the WAIT# signal is held low. The wait cycle ends (Tend) at the next cycle after the WAIT# signal becomes high.
With respect to the second and subsequent accesses, sampling of the WAIT# signal begins upon completion of the wait cycle of the page access (Tend). The wait cycle of the page access is extended while the WAIT# signal is held low, and ends (Tend) at the next cycle after the WAIT# signal becomes high.
Figure 16.40 and Figure 16.41 show examples of external wait insertion timing with the separate bus interface.

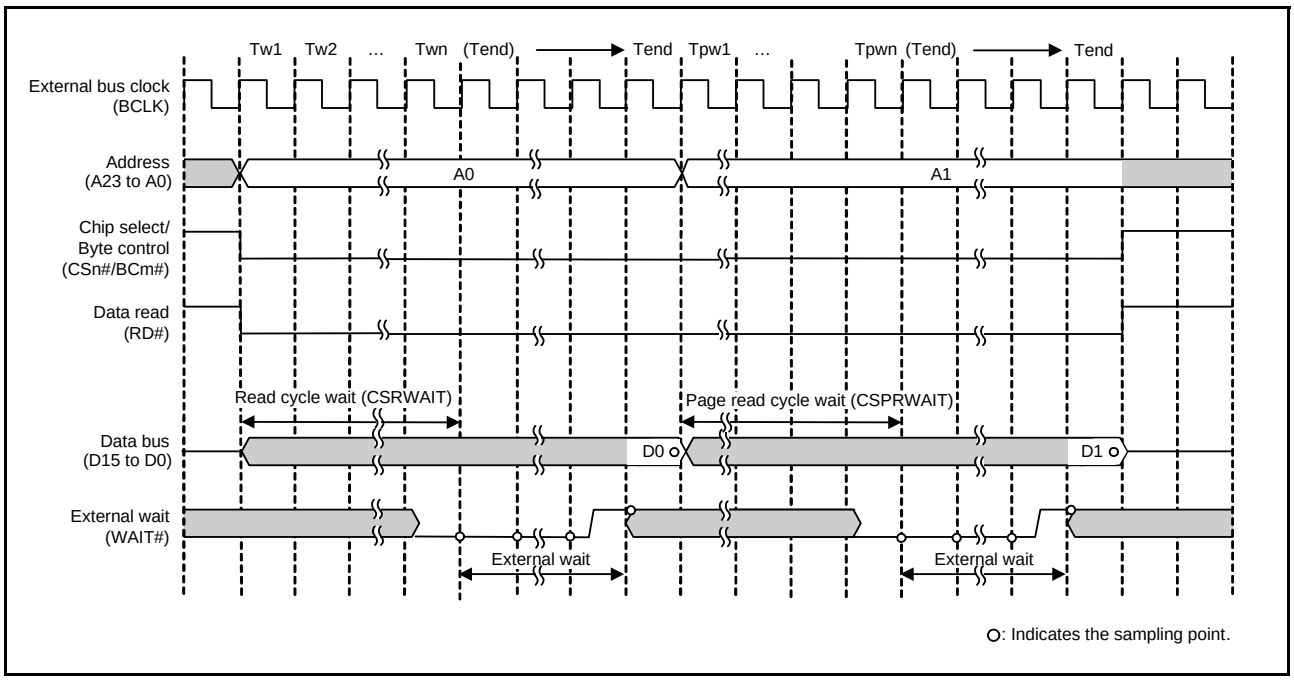


Figure 16.40 Example of External Wait Timing (Page-Read Access to 16-Bit Bus Space) (n = 0 to 7, m = 0, 1)

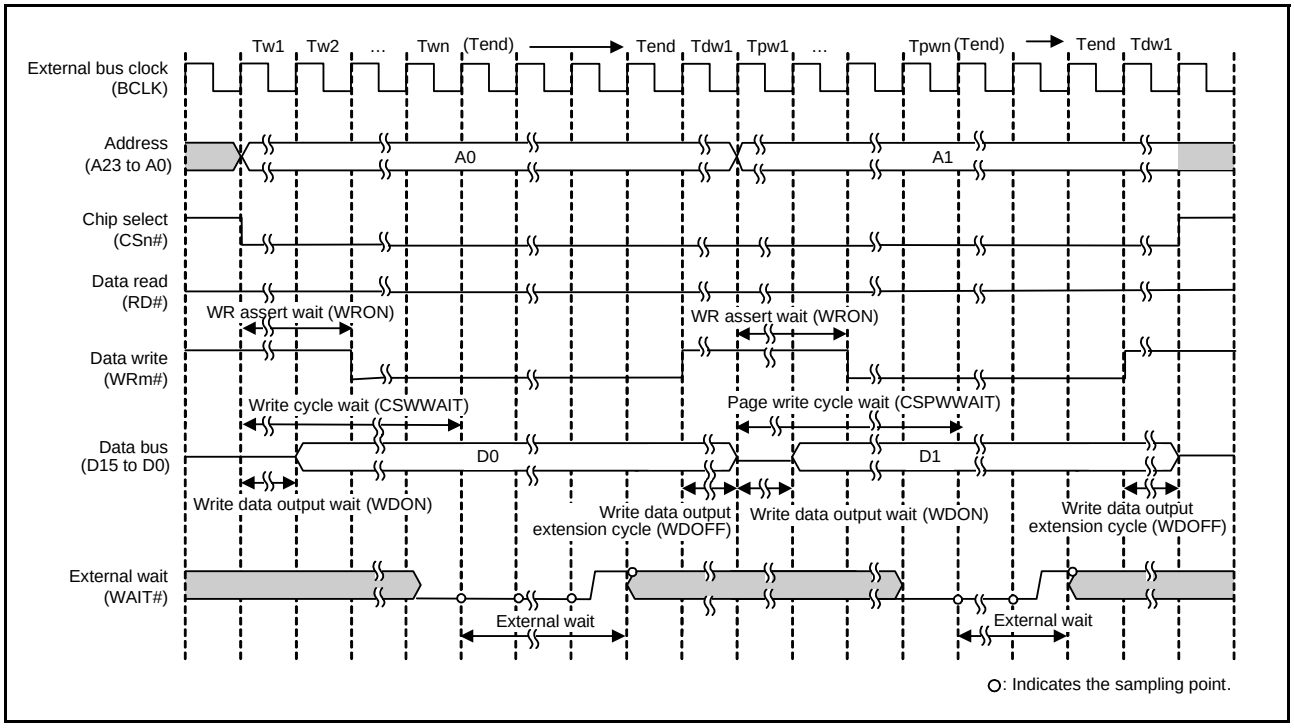


Figure 16.41 Example of External Wait Timing (Page-Write Access to 16-Bit Bus Space, in Byte Strobe Mode) (n = 0 to 7, m = 0, 1)

(3) Address/Data Multiplexed I/O Interface

In a data cycle with the address/data multiplexed I/O interface, programmed waits and pin waits using the WAIT pin can be inserted in the same way as that with the separate bus interface.

Address cycles are not affected by the wait control settings. Figure 16.42 shows an example of external wait insertion timing with the address/data multiplexed I/O interface.

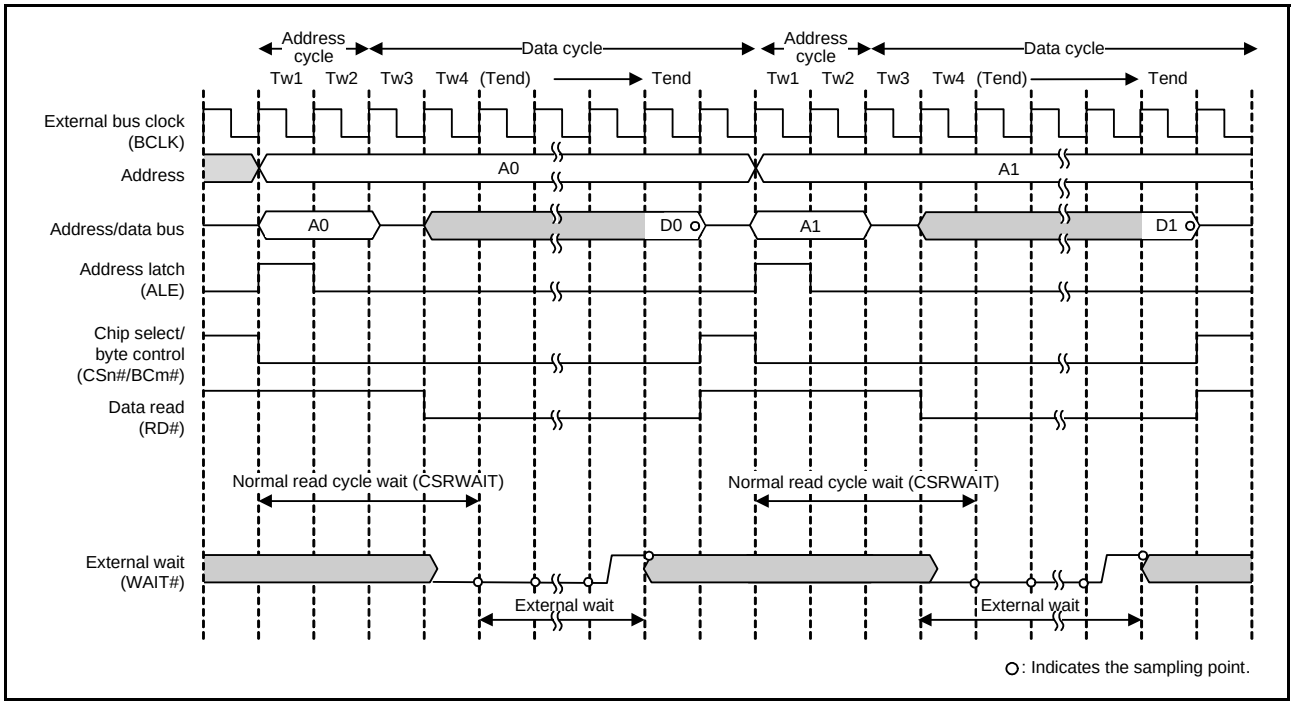


Figure 16.42 Example of External Wait Insertion Timing with Address/Data Multiplexed I/O Interface (m = 0, 1)

16.5.4 Insertion of Recovery Cycles

Recovery cycles can be inserted between consecutive rounds of external bus access by setting the recovery cycle insertion enable bit in CSRECEN to 1.
The number of recovery cycles to be inserted after read cycles and write cycles can be separately set for each area using CSnREC. When the preceding bus cycle is a write access, the number of write recovery cycles should be set with the WRCV[3:0] bits for the area. When the preceding bus cycle is a read access, the number of read recovery cycles should be set with the RRCV[3:0] bits for the area. For example, when CS1 read access occurs after CS0 read access, the number of recovery cycles to be inserted between them is set by the RRCV[3:0] bits in CS0REC.
Recovery cycles can be inserted on any of the following eight conditions. The recovery cycle insertion can be enabled or disabled with the RCVENj (j = 0 to 7) in CSRECEN when the preceding bus access is a separate bus access, and with RCVENMj (j = 0 to 7) when the preceding bus access is an address/data multiplexed bus access.

- After a read access to the external bus, a read access is made to the external bus in the same area.
- After a read access to the external bus, a read access is made to the external bus in a different area.
- After a read access to the external bus, a write access is made to the external bus in the same area.
- After a read access to the external bus, a write access is made to the external bus in a different area.
- After a write access to the external bus, a read access is made to the external bus in the same area.
- After a write access to the external bus, a read access is made to the external bus in a different area.
- After a write access to the external bus, a write access is made to the external bus in the same area.
- After a write access to the external bus, a write access is made to the external bus in a different area.

The recovery cycle starts at the end of the preceding bus cycle, i.e. when the CSn# signal (n = 0 to 7) is negated. A high-level period of the CSn# signal is inserted for the specified recovery cycle period starting from this point.
The CSn# signal for the next round of bus access is asserted immediately after the end of recovery cycles in the fastest case. Even if the next request for access to an external address space is generated during the recovery period, the next round of access over the external bus will start immediately after the end of recovery cycles.
When two or more external bus access cycles are required for a single transfer request from a bus master and the recovery cycle insertion condition is satisfied, recovery cycles are also inserted between these bus access cycles.
However, when page read access is enabled (CSnMOD.PRENB = 1) or page write access is enabled (CSnMOD.PWENB = 1), recovery cycles are not inserted except after the last bus access cycle of the transfer even if the recovery cycle insertion condition is satisfied (Figure 16.45).
Similarly, during normal accesses with page access enabled, recovery cycles are not inserted between bus access cycles but inserted only after the last bus access cycle of the transfer.
With the address/data multiplexed I/O interface, when the recovery cycle insertion condition is satisfied, recovery cycles are inserted between bus access cycles regardless of the page access enable setting.

Figure 16.43 to Figure 16.45 show examples of recovery cycle insertion with the separate bus interface.

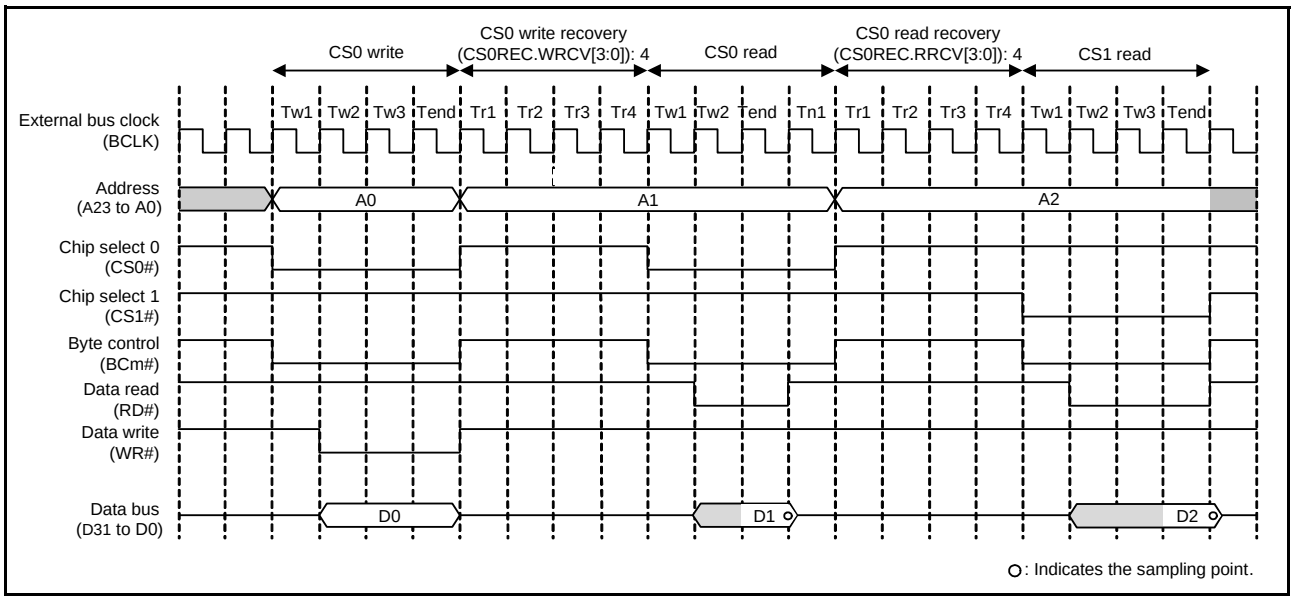


Figure 16.43 Example of Recovery Cycle Insertion with Separate Bus Interface (m = 0 to 3)

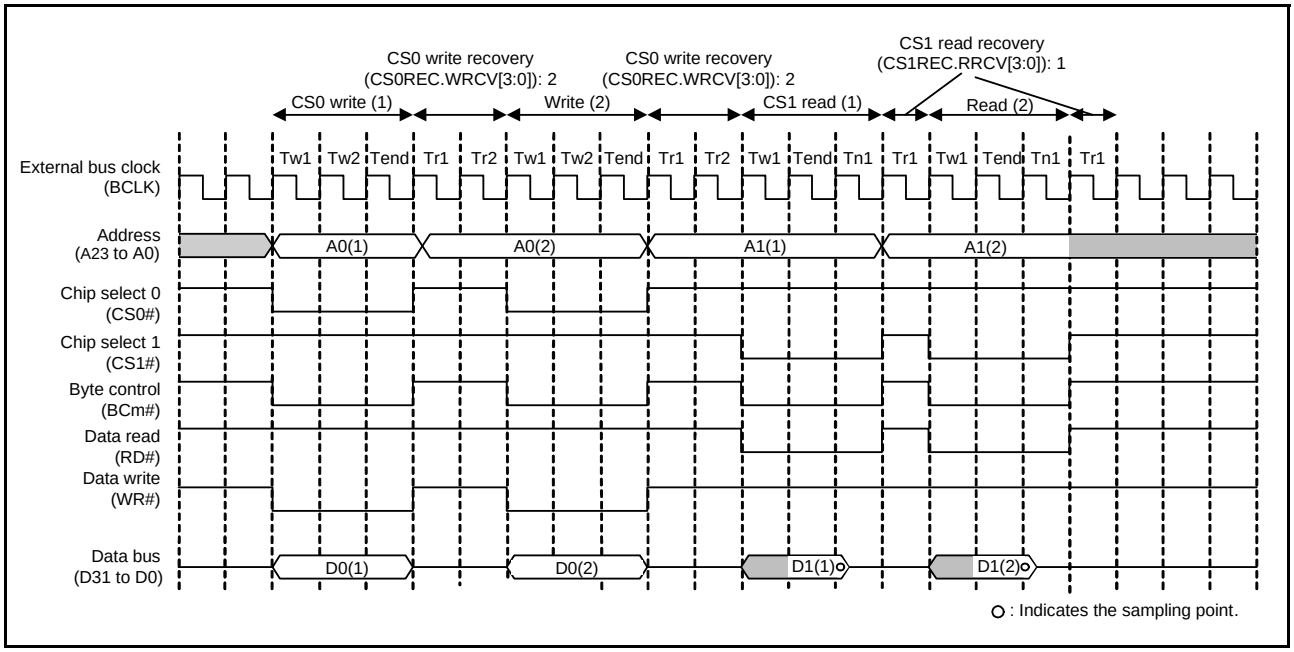


Figure 16.44 Example of Recovery Cycle Insertion When a Bus Access is Split (with Separate Bus Interface, Normal Access) (m = 0 to 3)

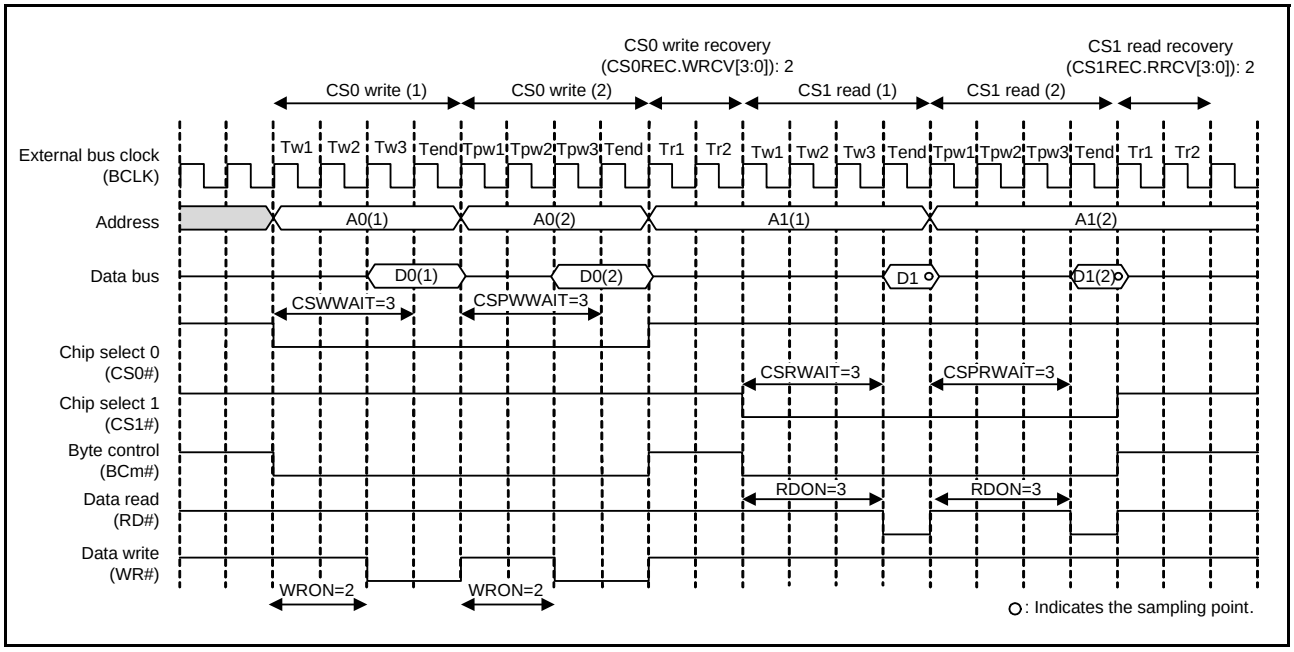


Figure 16.45 Example of Recovery Cycle Insertion When a Bus Access is Split (with Separate Bus Interface, Page Access) (m = 0 to 3)

Figure 16.46 shows examples of operations when the BCLK pin output selection bits are set for frequency-division of BCLK by 2.

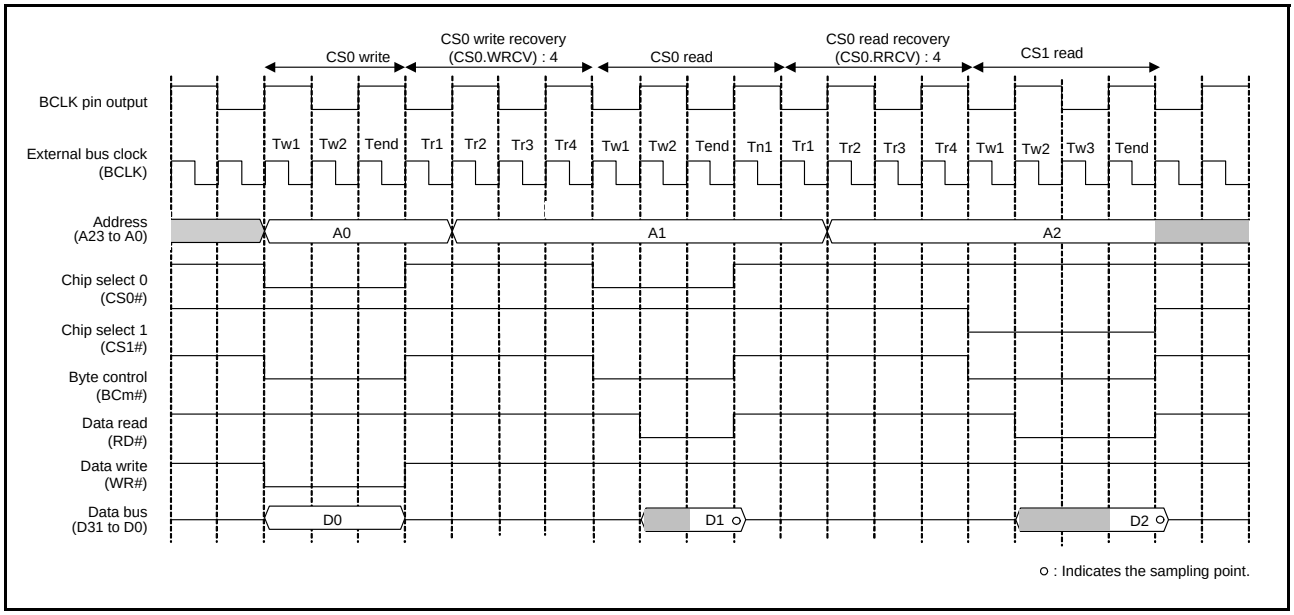


Figure 16.46 Example of Operation for Recovery Cycles when the BCLK Pin Output Selection Bits are Set for Frequency-Division of BCLK by 2 (For the Case of Normal Access through a Separate Bus Interface) (m = 0 to 3)

With the address/data multiplexed I/O interface, recovery cycles are inserted in the same way as that with the separate bus interface. Figure 16.47 and Figure 16.48 show examples of recovery cycle insertion with the address/data multiplexed I/O interface.

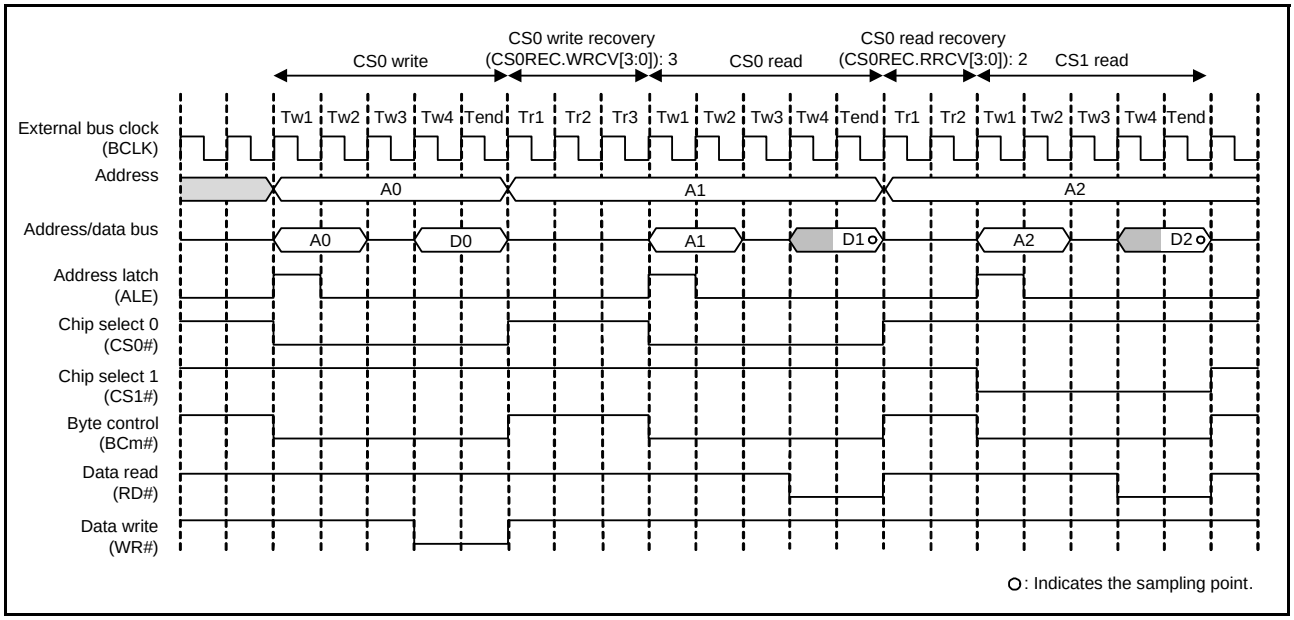


Figure 16.47 Example of Recovery Cycle Insertion with Address/Data Multiplexed I/O Interface (m = 0, 1)

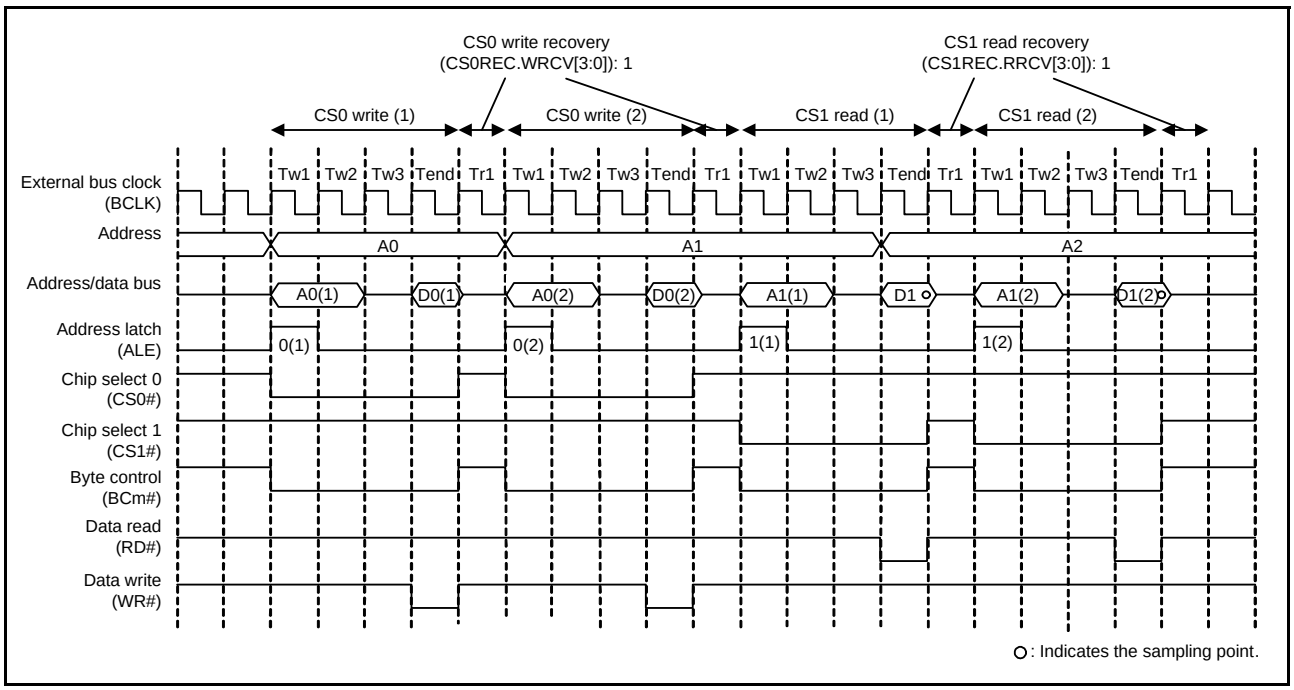


Figure 16.48 Example of Recovery Cycle Insertion When a Bus Access is Split with Address/Data Multiplexed I/O Interface (m = 0, 1)

16.5.5 No Access State

When no external address space is accessed, CSn#, BCn#, WRn#, and RDn# signals are high, ALE signal is low, and D31 to D0 are in the high-impedance state.

16.5.6 Write Buffer Function (External Bus)

The internal main bus is released by writing data to the write buffer before the write access is completed, which allows the next round of bus access to start. However, if the following round of bus access is to an external address space or to a register of the external bus controller, it is suspended until the external bus operations already in progress are completed. Figure 16.49 shows an example of operation when the write-buffer function is in use. When this function is in use, if the next operation after an external write is internal access, the internal access (access to on-chip memory or a peripheral module) is executed in parallel with the external write, i.e. without waiting for completion of the latter operation.

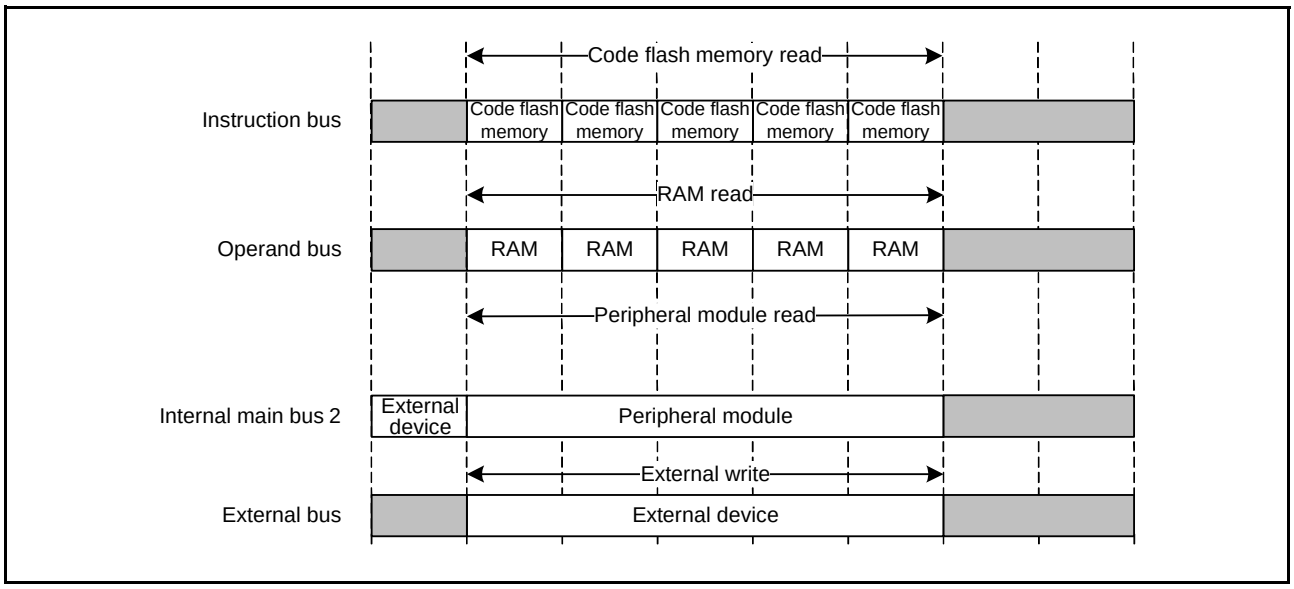


Figure 16.49 Example of Operation when the Write-Buffer Function is in Use

16.5.7 Limitations

(1) Limitations on Using Separate Bus Interface

- Limitations that apply to various bits of CSn wait control register 1 (CSnWCR1) and CSn wait control register 2 (CSnWCR2) at the times of normal and page accesses are listed in Table 16.10.
  Even if the setting of the page-read access enable bit in the CSn mode register or the page-write access enable bit in the CSn mode register selects permission (CSnMOD.PRENB = 1 or CSnMOD.PWENB = 1), the first round of access for page access and the access that does not fall within the scope of page access are normal access operation, and thus limitations on normal access must be satisfied.

Table 16.10 Limitations at the Time of Normal and Page Access

Limitations at the Time of Normal Access		Limitations at the Time of Page Access	
Reading	**Writing**	**Reading**	**Writing**
CSON[2:0] ≤ CSRWAIT RDON[2:0] ≤ CSRWAIT CSON[2:0] ≤ RDON	1 ≤ WDON[2:0] CSON[2:0] ≤ CSWWAIT WRON[2:0] ≤ CSWWAIT WDON[2:0] ≤ CSWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON	CSON[2:0] ≤ CSPRWAIT RDON[2:0] ≤ CSPRWAIT CSON[2:0] ≤ RDON	1 ≤ WDON[2:0] CSON[2:0] ≤ CSPWWAIT WRON[2:0] ≤ CSPWWAIT WDON[2:0] ≤ CSPWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON

- When two or more external bus access cycles are required for a single transfer request from a bus master and the recovery cycle insertion condition is satisfied with page read access enabled (CSnMOD.PRENB = 1) or page write access enabled (CSnMOD.PWENB = 1), recovery cycles are not inserted between bus access cycles but inserted only after the last bus access cycle of the transfer.

(2) Limitations on Using Address/Data Multiplexed Bus Interface

- In the address/data multiplexed I/O space, page accesses are invalid. If a page access setting is specified, the setting is ignored and the normal read or write operation is performed.
- When the address/data multiplexed I/O interface is set, the BSIZE[1:0] bits in CSnCR should not be set to the 32-bit bus space. If set, the operation cannot be guaranteed.

Table 16.11 Limitations at the Time of Normal and Page Access

Limitations at the Time of Normal Access	
Reading	**Writing**
CSON[2:0] ≤ CSRWAIT	CSON[2:0] ≤ CSWWAIT
RDON[2:0] ≤ CSRWAIT	WRON[2:0] ≤ CSWWAIT
CSON[2:0] ≤ RDON	WDON[2:0] ≤ CSWWAIT
AWAIT[1:0] + 2 ≤ RDON	WDOFF[2:0] ≤ CSWOFF
CSON[2:0] ≤ AWAIT	WDON[2:0] ≤ WRON
	CSON[2:0] ≤ WRON
	AWAIT[1:0] + 2 ≤ WRON
	AWAIT[1:0] + 2 ≤ WDON
	CSON[2:0] ≤ AWAIT

(3) Limitation when a Pin is Multiplexed between A0 and BC0# Functions

When the A0 and BC0# pin functions share the same pin, setting the single write strobe mode is prohibited in the 8-bit bus space; otherwise the operation is not guaranteed.

(4) Limitations when 1/2 BCLK is Selected with BCLK Pin Output Select Bit

When 1/2 BCLK is selected through the BCLK pin output select bit, the external bus access cycle starts at the rising edge of the BCLK pin output. However, when two or more external bus access cycles are generated for a single transfer request from a bus master, the second or subsequent external bus access cycle may start at the falling edge of the BCLK pin output depending on the wait cycle settings. Make appropriate register settings according to the specifications of the device to be connected.

(5) Limitations on EXDMAC Single Address Transfer Mode

- During the transfer in EXDMAC single address mode, the EDACK signal can be negated one cycle before the RD# signal is negated for read access or one cycle after the WR# signal is negated for write access through the settings of the EDACKn pin negate wait bit in the EXDMA output set register (EDMOMD.DACKW). Here, the CS# signal assertion and negation timing should be set so that the EDACK signal is enabled while the CS# signal is asserted. Table 16.12 and Table 16.13 show the limitations on the CSnWCR1 and CSnWCR2 register setting during the EXDMAC transfer in single address mode.
- To enable the EDACK signal output during the EXDMAC transfer in single address mode, the external wait function should be disabled (EWENB bit in CSnMOD = 0).
- When the external data read continuous assertion mode is specified (PRMOD bit in CSnMOD = 1) for page read access, the transfer in EXDMAC single address mode is prohibited; if such an attempt is made, correct operation is not guaranteed.
- In the address/data multiplexed I/O space, the transfer in EXDMAC single address mode is prohibited; if such an attempt is made, correct operation is not guaranteed.

Table 16.12 Limitations on EXDMAC Single Address Transfer Mode (EDMOMD.DACKW = 0)

Limitations at the Time of Normal Access		Limitations at the Time of Page Access	
Reading	**Writing**	**Reading**	**Writing**
CSON[2:0] ≤ CSRWAIT RDON[2:0] ≤ CSRWAIT CSON[2:0] ≤ RDON 1 ≤ RDON	CSON[2:0] ≤ CSWWAIT WRON[2:0] ≤ CSWWAIT WDON[2:0] ≤ CSWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON 1 ≤ WRON	CSON[2:0] ≤ CSPRWAIT RDON[2:0] ≤ CSPRWAIT CSON[2:0] ≤ RDON 1 ≤ RDON	CSON[2:0] ≤ CSPWWAIT WRON[2:0] ≤ CSPWWAIT WDON[2:0] ≤ CSPWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON 1 ≤ WRON

Table 16.13 Limitations on EXDMAC Single Address Transfer Mode (EDMOMD.DACKW = 1)

Limitations at the Time of Normal Access		Limitations at the Time of Page Access	
Reading	**Writing**	**Reading**	**Writing**
CSON[2:0] ≤ CSRWAIT RDON[2:0] < CSRWAIT CSON[2:0] ≤ RDON 1 ≤ RDON	CSON[2:0] ≤ CSWWAIT WRON[2:0] ≤ CSWWAIT WDON[2:0] ≤ CSWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON 1 ≤ WRON 1 ≤ WDOFF	CSON[2:0] ≤ CSPRWAIT RDON[2:0] < CSPRWAIT CSON[2:0] ≤RDON 1 ≤ RDON	CSON[2:0] ≤ CSPWWAIT WRON[2:0] ≤ CSPWWAIT WDON[2:0] ≤CSPWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON 1 ≤ WRON 1 ≤ WDOFF

(6) Prohibition of Access that Spans Areas of Address Space

Single access that spans several areas of the address space is prohibited, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single word or longword access.

Single access that spans two areas of the address space is also prohibited in the EXDMAC block transfer in single address mode or cluster transfer, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single access in the EXDMAC block transfer in single address mode or cluster transfer.

(7) Restrictions on RMPA and String-Manipulation Instructions

- Although the external space has a per-area ending-switching facility (only for data), the allocation of data to be handled by RMPA or string-manipulation instructions to an area where the endian differs from that of the chip is prohibited, and operation is not guaranteed if this restriction is not observed. If data to be handled by RMPA or string-manipulation instructions are allocated to the external space, they must be allocated to areas where the endian setting is the same as that for the chip.
- The allocation of data to be handled by RMPA or string-manipulation instructions to I/O registers is prohibited, and operation is not guaranteed if this restriction is not observed.

(8) Restriction on Instruction Code

When the endian setting for each area is different from that for the chip, no instruction code can be allocated in the area. The instruction code should be allocated to the external space whose endian setting is the same as that for the chip.

16.6 SDRAM Area Controller Operation

The following sections describe how the SDRAM area controller (SDRAMC) is enabled and the SDRAM bus width is set, which is followed by the description of SDRAMC operations including read, write, auto-refresh, self-refresh, initialization sequence, and mode register setting.

16.6.1 Enabling/Disabling SDRAM Access and Setting SDRAM Bus Width

SDRAM access can be enabled or disabled using the SDC control register (SDCCR). The SDRAM bus width can also be set using SDCCR.
Even when the operation of the SDRAM address space is disabled, refresh operation is available as long as self-refresh or auto-refresh operation is enabled.

16.6.2 No Access State

When no external address space is accessed, SDCS#, WEn#, RAS#, and CAS# signals are high.

16.6.3 Insertion of Recovery Cycles

When access to the SDRAM area follows access to the CS area, data recovery cycles are inserted for the CS area controller (CSC). If the number of recovery cycles for the CSC is 0, the ACT command for the next SDRAM access is issued immediately after negation of CSn# signal at the earliest. If the number of recovery cycles are not 0, the ACT command is issued two cycles after the specified recovery cycle period elapsed after negation of CSn# signal at the earliest. Since no data conflicts can occur during access to the SDRAM area, there is no need to set data recovery cycles for the SDRAM (fixed to zero cycle).

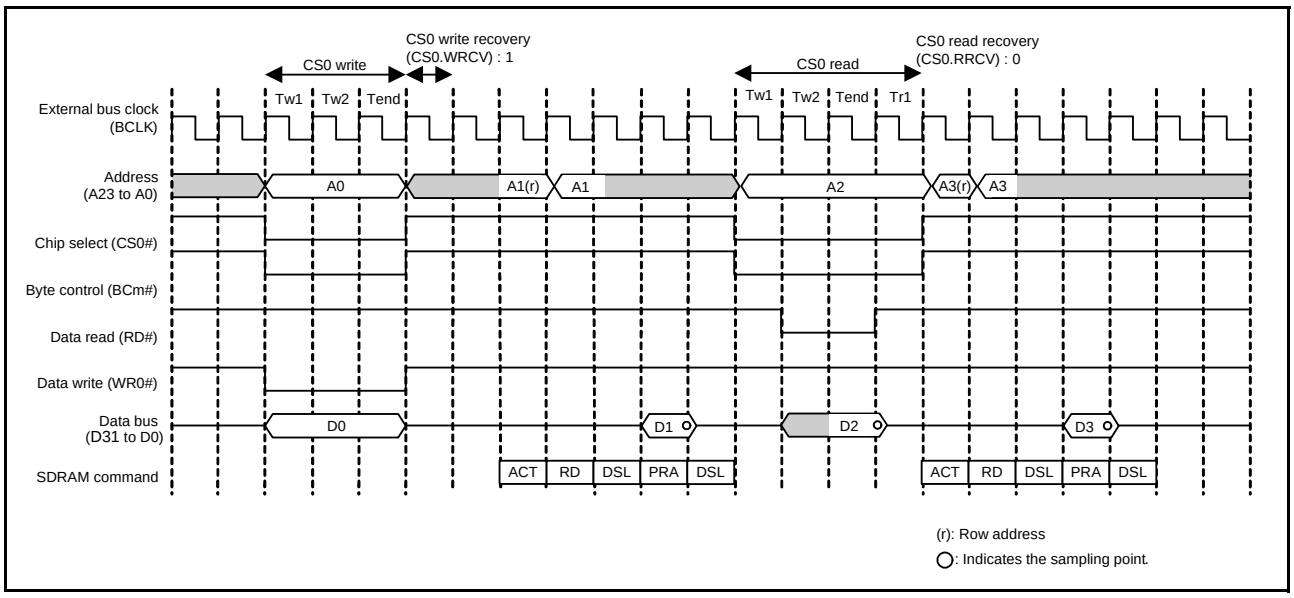


Figure 16.50 Example of Recovery Timing (for SDRAM Access)

16.6.4 Write Buffer Function

In write access, the internal main bus is released by writing data to the write buffer before the write access is completed, which allows the next round of bus access to start. However, if the following round of bus access is to an external address space or to a register of the external bus controller, it is suspended until the external bus operations already in progress are completed.

16.6.5 SDRAM Commands

The SDRAMC issues a command for each bus cycle to control SDRAM. Commands are defined by combination of SDCS#, RAS#, CAS#, WE#, CKE, and other signals.
Table 16.14 lists the commands issued by the SDRAMC.

Table 16.14 List of SDRAMC Commands

							CKE			
Name	**Abbreviation**	**Command**	**SDCS#**	**RAS#**	**CAS#**	**WE#**	**n-1**	**n**	**BA1**	**BA0**
DESL	DSL	Device deselect	H	x	x	x	H	x	x	x
ACTV	ACT	Bank active	L	L	H	H	H	x	V	V
READ	RD	Read	L	H	L	H	H	x	V	V
WRIT	WRI	Write	L	H	L	L	H	x	V	V
PALL	PRA	All bank precharge	L	L	H	L	H	x	x	x
REF	RFA	Auto-refresh	L	L	L	H	H	x	x	x
MRS	MRS	Mode register set	L	L	L	L	H	x	L	L
SELF	RFS	Self-refresh entry	L	L	L	H	H	L	x	x
SELFX	RFX	Self-refresh end	H	x	x	x	L	H	x	x

Note: H: High level, L: Low level, V: Valid, x: Don't care. (High level or low level)
n: Command issue cycle, n – 1: One cycle before the command is issued.

16.6.6 Conditions for Setting SDRAMC Registers

SDRAMC registers should be modified only when all the corresponding conditions are satisfied as shown in Table 16.15.

Table 16.15 Conditions for Register Modification

Function/Operation	**Registers**	**Conditions**
Self-refresh	SDSELF*1	• SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Auto-refresh operation is enabled. (SDRFEN.RFEN = 1)
Auto-refresh	SDRFCR	Self-refresh operation is disabled. (SDSELF.SFEN = 0)
	SDRFEN	Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Initialization sequence	SDIR*1	The SDICR has not been set yet, and the same conditions as SDICR modification should be satisfied.
	SDICR*1	• SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled. (SDRFEN.RFEN = 0) • Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Address register	SDADR	• SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled. (SDRFEN.RFEN = 0) • Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Timing register	SDTR	• During self-refresh operation (SDSELF.SFEN = 1) or • SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled. (SDRFEN.RFEN = 0) • Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Mode register	SDMOD*1	• SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Access mode register	SDAMOD	• SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled. (SDRFEN.RFEN = 0) • Self-refresh operation is disabled. (SDSELF.SFEN = 0)

Note 1. Before modification, confirm that all the status bits in SDSR are 0.
Note 2. After writing 0 to the EXENB bit, confirm that the EXENB bit is set to 0.

16.6.7 Self-Refresh

Transition to or recovery from self-refresh mode can be controlled using the SDRAM self-refresh control register (SDSELF).

Immediately before transition to self-refresh mode, auto-refresh operation is performed. In self-refresh mode, the CKE signal is low. Immediately after recovery from self-refresh mode, the auto-refresh cycle is started.

Figure 16.51 and Figure 16.52 show timing examples of transition to self-refresh mode and recovery from self-refresh mode, respectively.

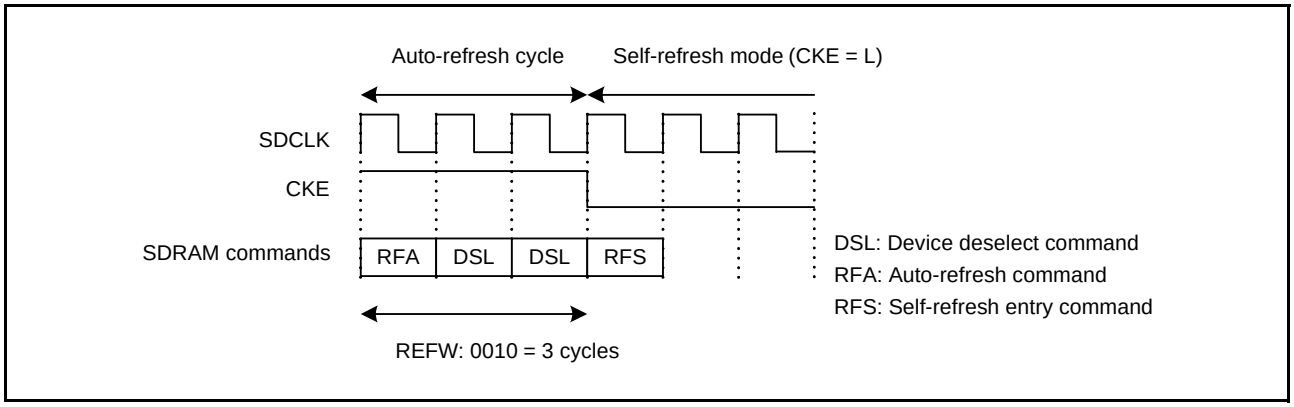


Figure 16.51 Timing Example of Transition to Self-Refresh Mode (when SDRFCR.REFW[3:0] = 0010b: 3 Cycles)

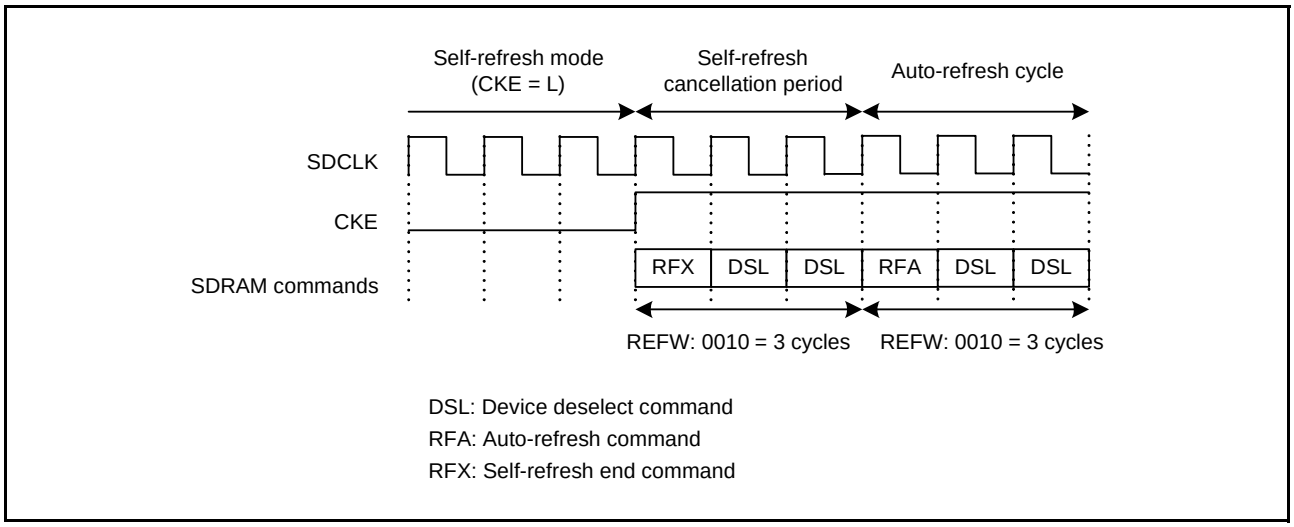


Figure 16.52 Timing Example of Recovery from Self-Refresh Mode

(1) Self-Refresh in All-Module-Clock Stop Mode

When causing transition to self-refresh mode in all-module-clock stop mode, first cause transition to self-refresh mode according to the procedure shown in section 16.6.12.2, Procedure for Transition to and Recovery from Self-Refresh Mode, and then make settings to cause transition to all-module-clock stop mode.

After canceling all-module-clock stop mode, follow the procedure shown in section 16.6.12.2, Procedure for Transition to and Recovery from Self-Refresh Mode.

For details of transition and cancellation of all-module-clock stop mode, refer to section 11, Low Power Consumption.

(2) Self-Refresh in Software Standby Mode

When causing transition to self-refresh mode in software standby mode, first cause transition to self-refresh mode

according to the procedure shown in section 16.6.12.2, Procedure for Transition to and Recovery from Self-Refresh Mode, and then make settings to cause transition to software standby mode. In software standby mode, set the output port enable bit (OPE) in the standby control register (SBYCR) to 1 to hold the output state of the address bus and bus control signals.
After canceling software standby mode, follow the procedure shown in section 16.6.12.2, Procedure for Transition to and Recovery from Self-Refresh Mode.
For details of transition and cancellation of software standby mode, refer to section 11, Low Power Consumption.

(3) Self-Refresh in Deep Software Standby Mode

Transition to deep software standby mode is performed via software standby mode. On transition to deep software standby mode from software standby mode, the state of pins remains unchanged. Therefore, transition to self-refresh mode in deep software standby mode can be made according to the same procedure as that in software standby mode. In deep software standby mode, however, additional setting is necessary to cause transition to self-refresh mode; it is necessary to set the I/O port keep bit (IOKEEP) in the deep software standby control register (DPSBYCR) to 1.
Since the SDRAMC is internally reset by an internal reset signal when deep software standby mode is canceled, the SDRAM control registers need to be set again. After canceling software standby mode, follow the procedure shown below to cancel self-refresh mode. Figure 16.53 shows self-refresh timing in deep software standby mode.
For details of transition and cancellation of deep software standby mode, refer to section 11, Low Power Consumption.

1. In deep software standby mode, the CKE signal output remains low according to the IOKEEP setting in DPSBYCR.
2. Start clock supply to SDRAMC.
3. Set the SDRAM control registers (SDCMOD, SDAMOD, SDADR, and SDTR) again, which have been initialized by an internal reset upon transition to deep software standby mode, and then enable auto-refresh operation (RFEN bit in SDRFEN = 1).
4. Check that all the status bits in SDSR are set to 0 and set the SFEN bit in SDSELF to 1 to set self-refresh mode again.
5. Modify port settings for the SDRAM interface according to the procedure below.
   (1) Set the enable bits for the SDRAM pins (PFBCR1.MDSDE and PFBCR1.DQM1E in PFBCR1) to 1 to set the ports for SDRAM again.
   (2) Set the enable bit for the SDCLK pin (PFBCR1.SDCLKE in PFBCR1) to 1 to enable SDCLK pin output again.
   (3) Set the IOKEEP bit in DPSBYCR to 0 to release the I/O ports from the held state.
6. Set the PSTOP0 bit in SCKCR to 0 to start clock supply to the SDRAM via the SDCLK pin.
7. Check that all the status bits in SDSR are set to 0 and set the SFEN bit in SDSELF to 0 to cancel self-refresh mode.

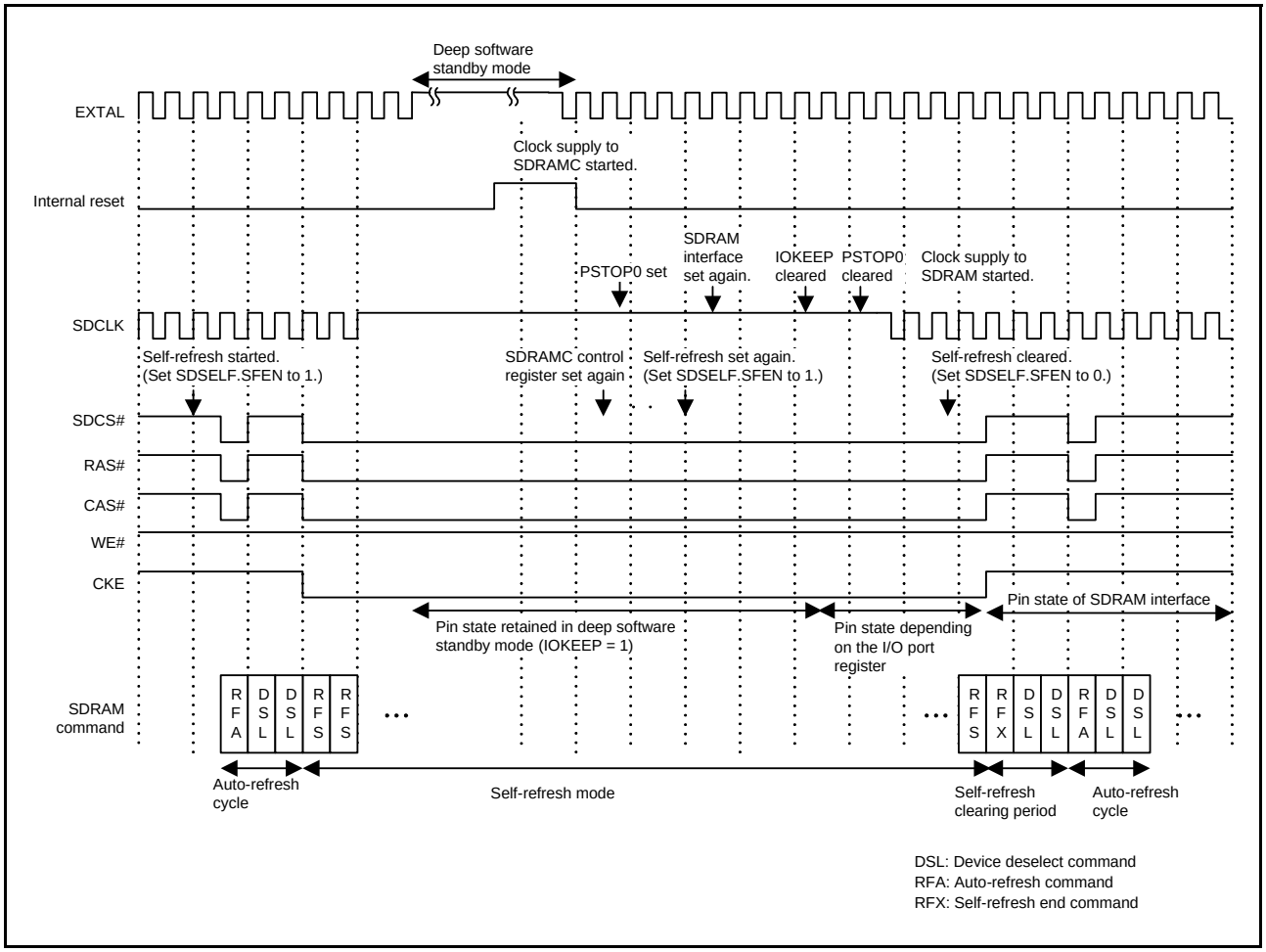


Figure 16.53 Timing Example of Self-Refresh Cycle (in Deep Software Standby Mode)

16.6.8 Auto-Refresh

The auto-refresh cycle can be started by setting the auto-refresh operation enable bit (RFEN) in the SDRAM auto-refresh control register (SDRFEN) to 1. Once the cycle is started, refresh requests are generated at fixed intervals determined by the refresh counter to start the auto-refresh cycle. However, since refresh requests are not accepted during read/write access, the auto-refresh cycle may be suspended. If an auto-refresh request is issued during consecutive access to the SDRAM, the auto-refresh cycle starts after bus access in response to a single transfer request from the bus master is completed.

If an SDRAM access and a refresh request are generated at the same time, the refresh request takes precedence. A CS area access and a refresh request can be made at the same if the SDCS#, RAS#, CAS#, WE#, and CKE signals, which are necessary for issuing the refresh command, are exclusively provided for SDRAM access.

When the RFEN bit in SDRFEN is set to 1 again after the auto-refresh cycle is started, a refresh request is generated. However, if a request is made during read/write access, a request is actually generated when access is completed.

The refresh counter is halted during self-refresh operation. After recovery from self-refresh mode, the auto-refresh cycle is started and the counter value is reset thus resuming the counter operation.

Figure 16.54 shows an example of the timing of an auto-refresh cycle.

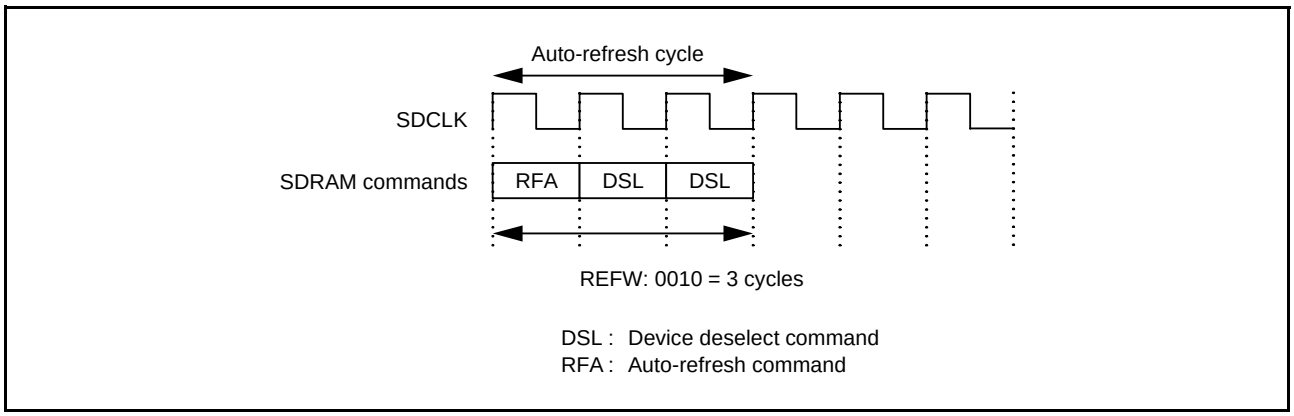


Figure 16.54 Timing Example of Auto-Refresh Cycle (1)

Figure 16.55 and Figure 16.56 show examples of operation when an auto-refresh request is generated during single access and continuous access, respectively.

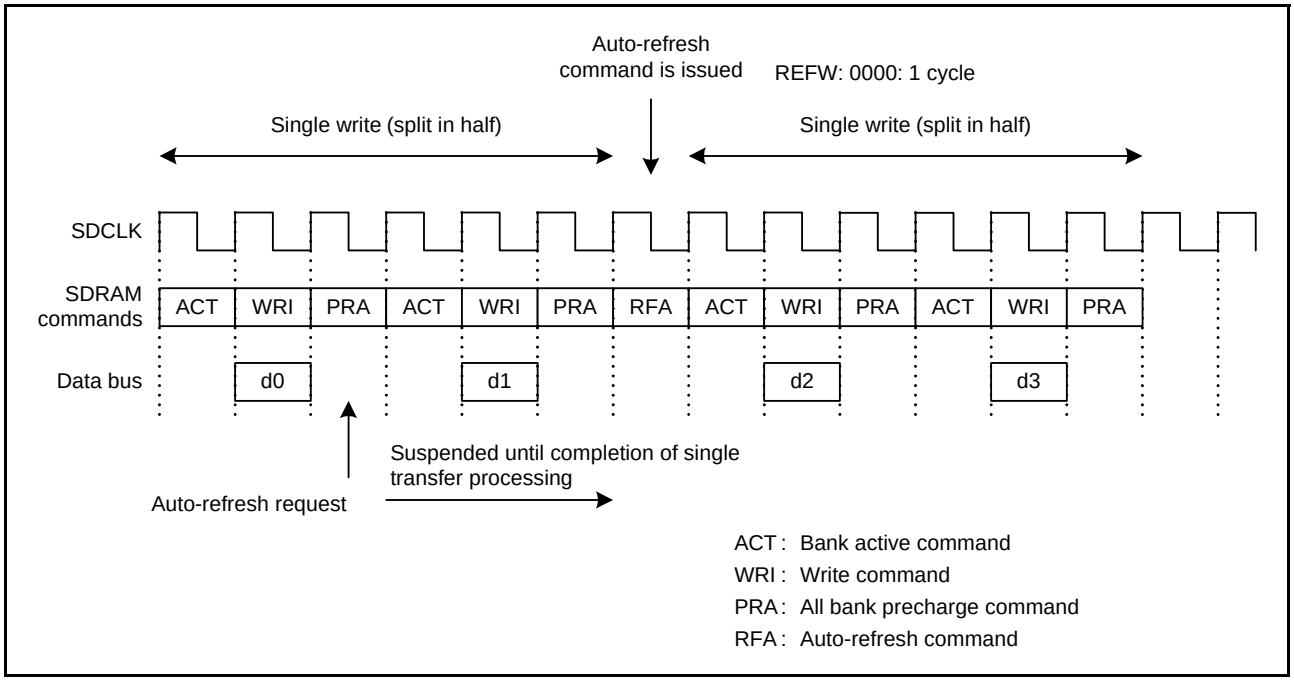


Figure 16.55 Timing Example of Auto-Refresh Cycle (2) (Auto-Refresh Request is Made during Single Access)

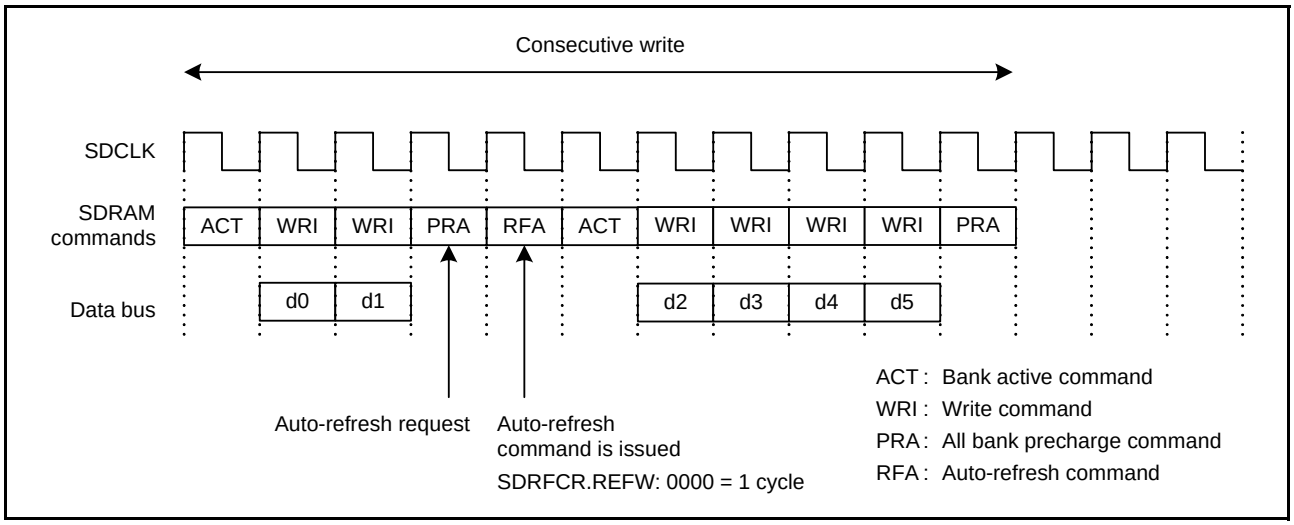


**Figure 16.56 Timing Example of Auto-Refresh Cycle (3)
(Auto-Refresh Request is Made during Continuous Access)**

16.6.9 Initialization Sequencer

The SDRAMC has a sequencer to issue SDRAM initialization commands. After a reset, the initialization sequence must be activated without fail; the operation is not guaranteed if the SDRAM is not initialized.
The SDRAM initialization sequencer issues an all-bank-precharge command followed by auto-refresh commands n times (n = 1 to 15). The SDRAM initialization sequence timing can be set using the SDRAM initialization register (SDIR). The SDRAM initialization sequence can be activated using the SDRAM initialization sequence control register (SDICR). These registers should be set only when the conditions listed in Table 16.15, Conditions for Register Modification.
Figure 16.57 shows a timing example of the SDRAM initialization sequence. When the ARFC[3:0] bits in SDIR are set so that auto-refresh operation is performed two or more times, auto-refresh cycles are repeated in the initialization sequence accordingly.

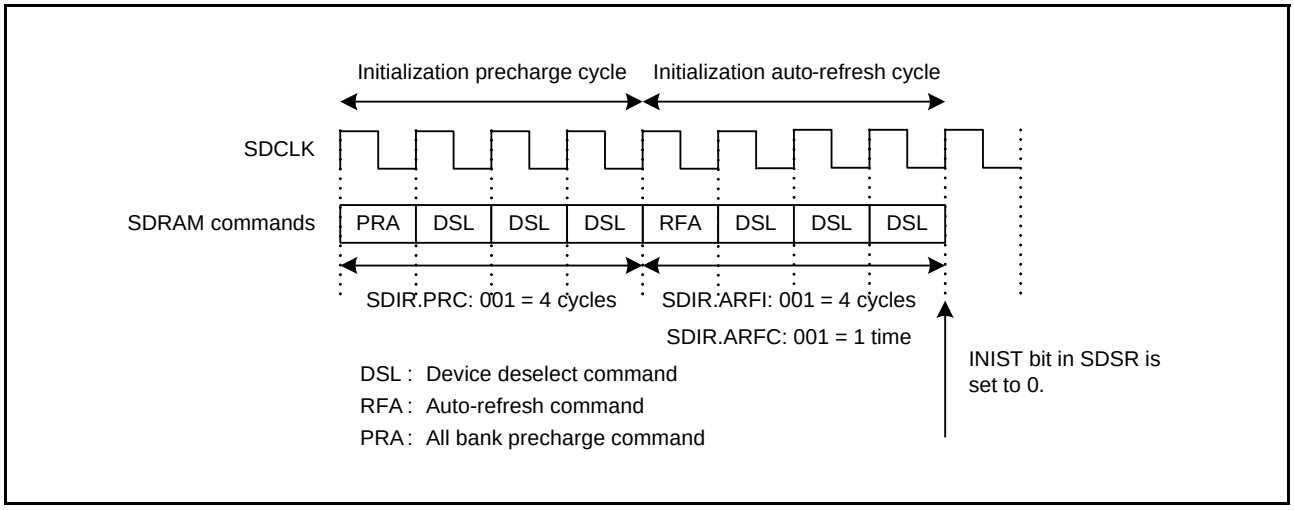


Figure 16.57 Timing Example of SDRAM Initialization Sequence

16.6.10 Read/Write Access

The SDRAMC controls read/write access in the following two modes.

- Single access mode: the row address is output each time data is accessed
- Consecutive access mode: when the same row address is accessed consecutively, only the column address is changed after the row address is output, enabling quick data access.

Consecutive SDRAM access is enabled by setting the continuous access enable bit (BE) in SDRAM access mode register (SDAMOD) to 1 in EXDMAC cluster transfer or block transfer in single address mode.
If the data size for a single transfer by the EXDMAC is less than the width of the external bus, and when bus access for a single transfer request ends once, in the same way as for non-aligned access, operation with consecutive access becomes possible.
When the above condition is not satisfied, the setting for consecutive-access mode is prohibited, and operation is not guaranteed if the setting is made.
Furthermore, setting the SDRAMC column-latency setting bits (CL[2:0]) in SDTR to 1 (CL = 1) in consecutive-access mode is prohibited, and operation is not guaranteed if this setting is made.
When the BE bit in SDAMOD is 0, single access is used in both cluster transfer for the EXDMAC and block transfer in single address mode.

(1) Single Access

Figure 16.58 and Figure 16.59 show timing examples of single read and single write, respectively. The specific access timing depends on the SDRAM timing register (SDTR) settings. For details, refer to section 16.6.12.3, Timing Register Settings and Access Timing.

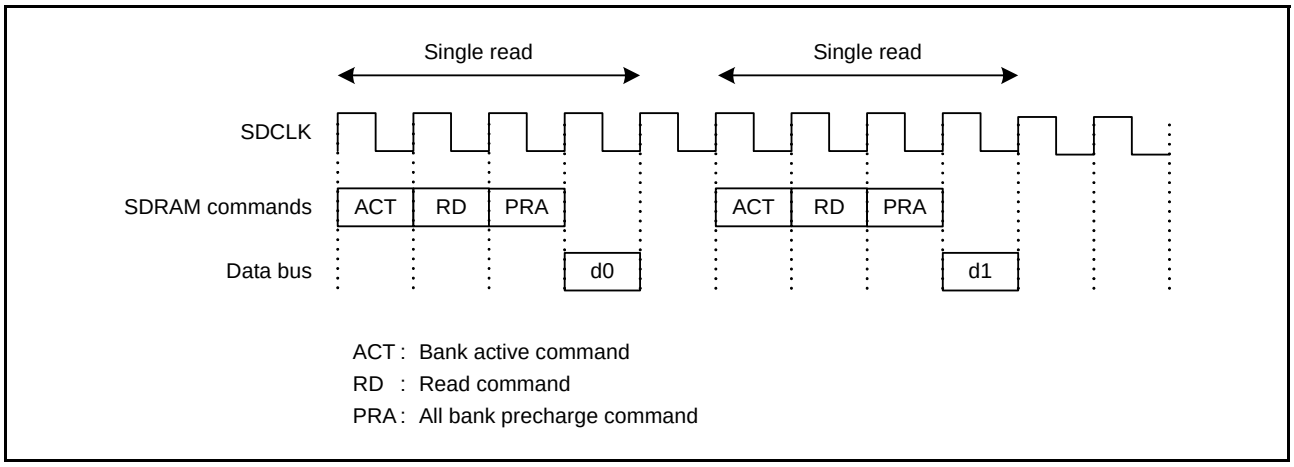


Figure 16.58 Timing Example of Single Read (SDTR.CL[2:0] = 010b: 2 Cycles)

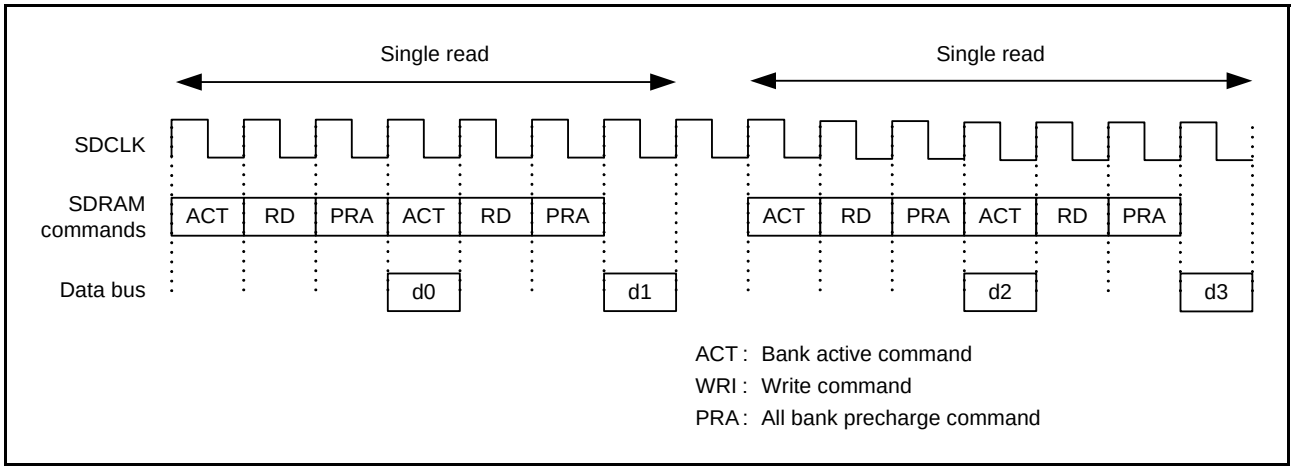


Figure 16.59 Timing Example of Single Read (Cluster Transfer by EXDMAC or Block Transfer in Single Address Mode with SDAMOD.BE = 0 and SDTR.CL[2:0] = 010b: 2 Cycles)

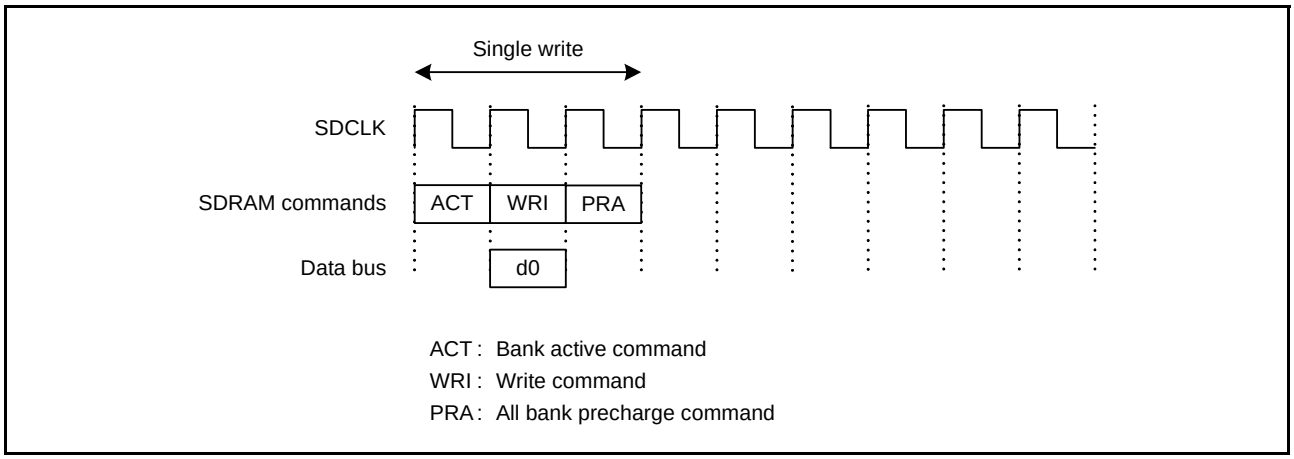


Figure 16.60 Timing Example of Single Write (when the Shortest Timing is Set)

(2) Consecutive Access

Figure 16.61 and Figure 16.62 show timing examples of consecutive read and consecutive write for four data, respectively.

When the SDRAM row address changes during transfer, the pertinent row is automatically deactivated or activated appropriately.

Figure 16.63 shows a timing example of consecutive write in which the row address changes.

The specific access timing depends on the SDRAM timing register (SDTR) settings. For details, refer to section 16.6.12.3, Timing Register Settings and Access Timing.

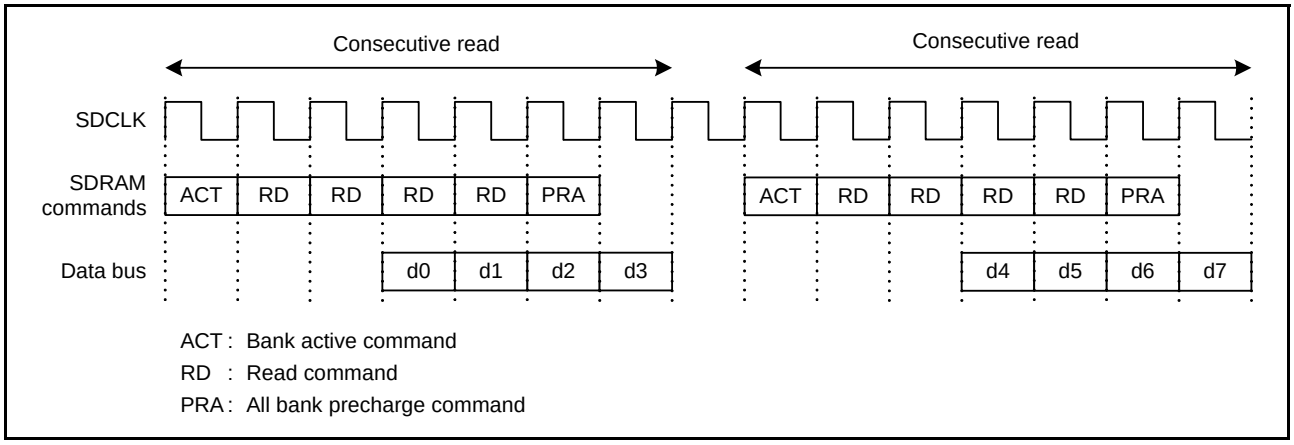


Figure 16.61 Timing Example of Consecutive Read (SDAMOD.BE = 1 and SDTR.CL[2:0] = 010b: 2 Cycles)

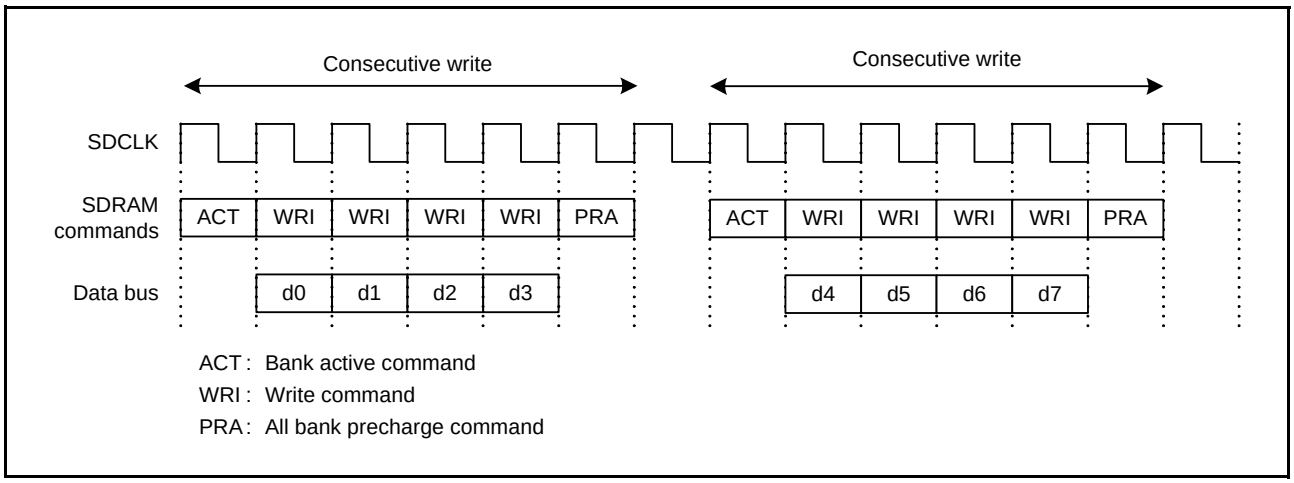


Figure 16.62 Timing Example of Consecutive Write (SDAMOD.BE = 1, when the Earliest Timing is Set)

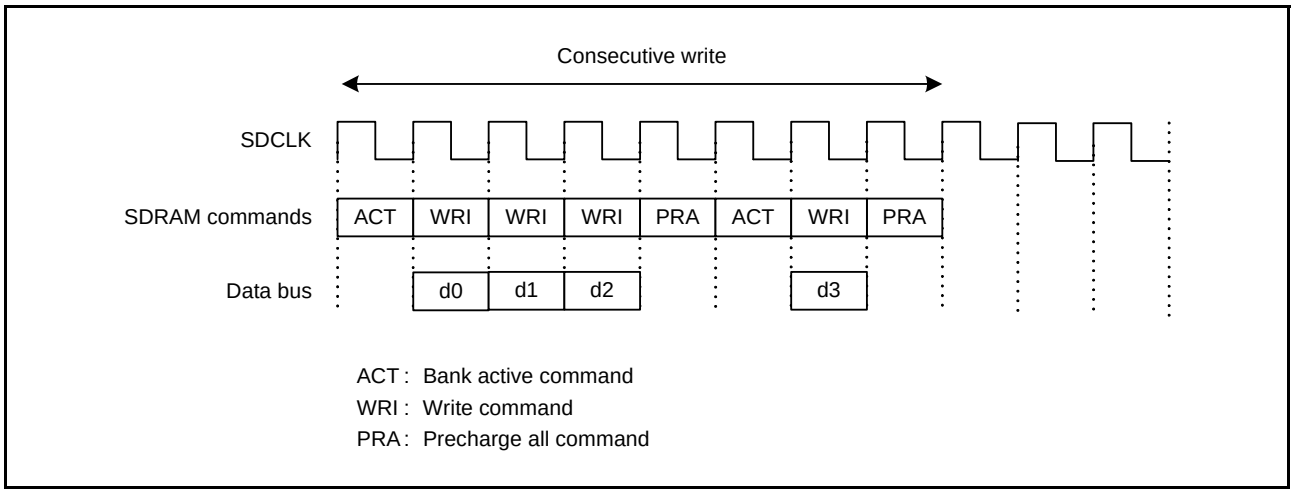


Figure 16.63 Timing Example of Consecutive Write (SDAMOD.BE = 1, when the Earliest Timing is Set) in which the Row Address Changes

16.6.11 Setting Mode Register

Setting the SDRAM mode register (SDMOD) allows the mode register set command to be issued to SDRAM and the value set in the MR[14:0] bits in SDMOD to be output to the lower bits of the address; specifically, to the A14 to A0 for 8-bit bus width, A15 to A1 for 16-bit bus width, and A16 to A2 for 32-bit bus width. Therefore, set the SDCCR.BSIZE[1:0] bits before setting the mode register, to determine the data bus width of the SDRAM.
Figure 16.64 shows the mode register setting timing.

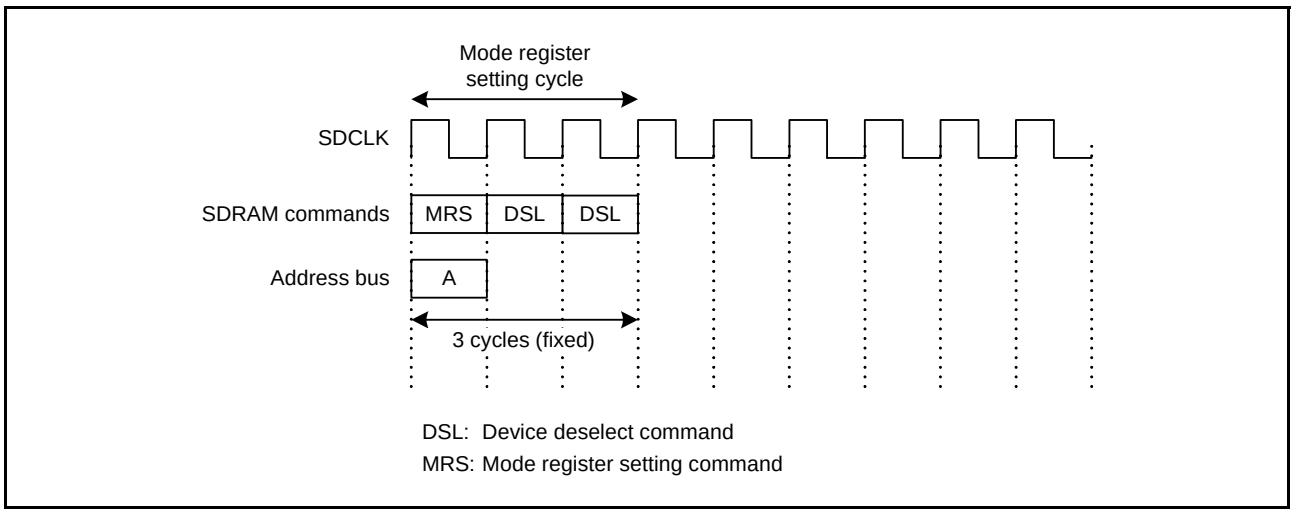


Figure 16.64 Mode Register Setting Timing

16.6.12 SDRAMC Setting Examples

This section describes the SDRAMC setting procedure, timing register setting examples, and procedure for transition to and recovery from self-refresh mode.

16.6.12.1 SDRAMC Access Procedure

Figure 16.65 shows the SDRAMC setting procedure.
The shown specifications including a power-up sequence may be different from that from the specifications of the SDRAM actually used; the system should be designed after reviewing the specifications of the SDRAM.

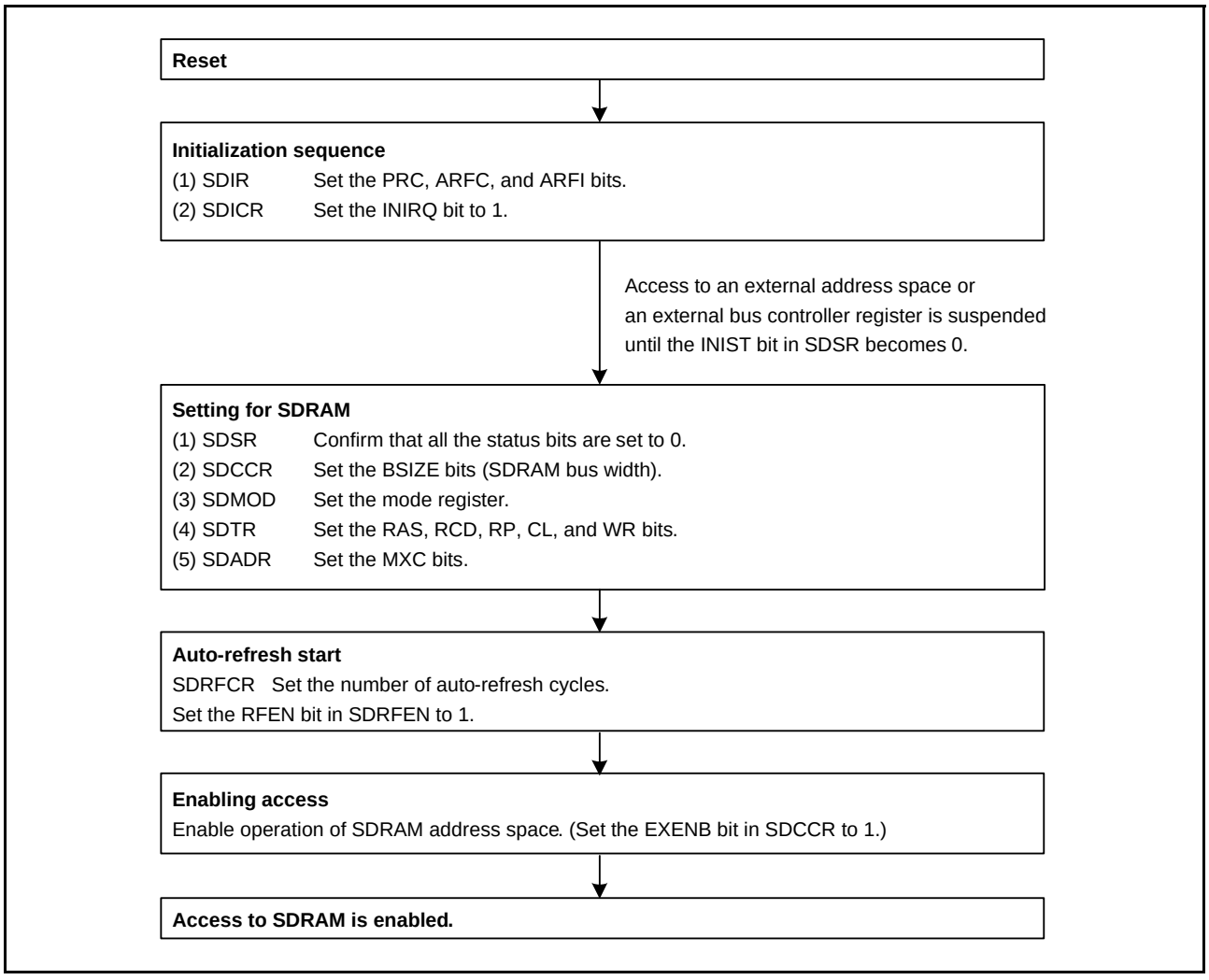


Figure 16.65 SDRAMC Setting Procedure

16.6.12.2 Procedure for Transition to and Recovery from Self-Refresh Mode

Figure 16.66 shows the procedure for transition to and recovery from self-refresh mode.

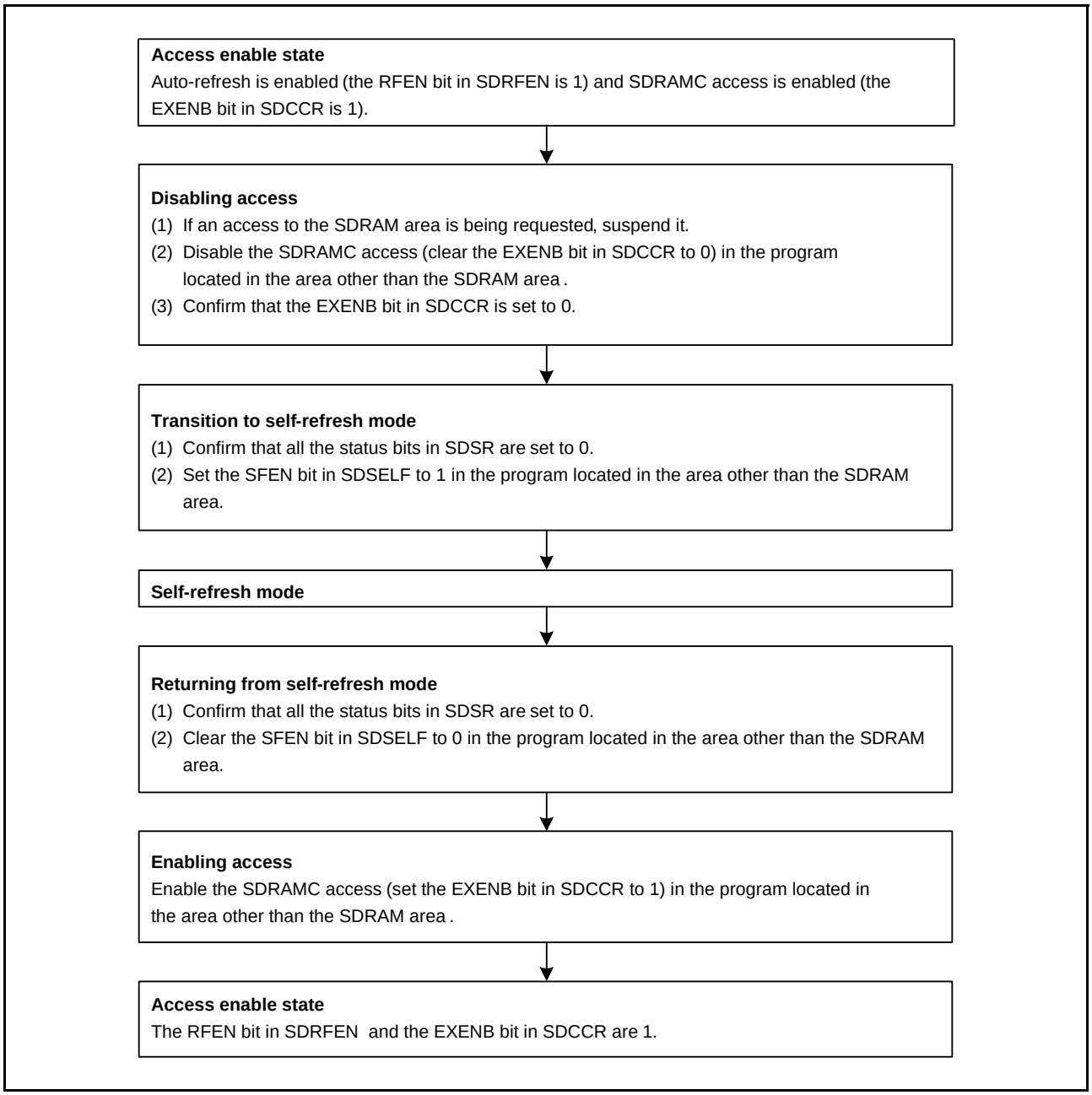


Figure 16.66 Procedure for Transition to and Recovery from Self-Refresh Mode

Note: Transition to and recovery from self-refresh mode requires SDRAM access to be disabled. Accordingly, transition to and recovery from self-refresh mode cannot be made during SDRAM access. The instructions below should be followed in programming.
- Before making transition to self-refresh mode, disable the access to the SDRAM area.
- During transition to self-refresh mode, self-refresh operation, and recovery from self-refresh mode, do not allow any operand access or instruction fetch (including prefetch) to the SDRAM area to be generated.

Figure 16.67 shows the procedure for transition to and recovery from self-refresh mode in deep software standby mode.

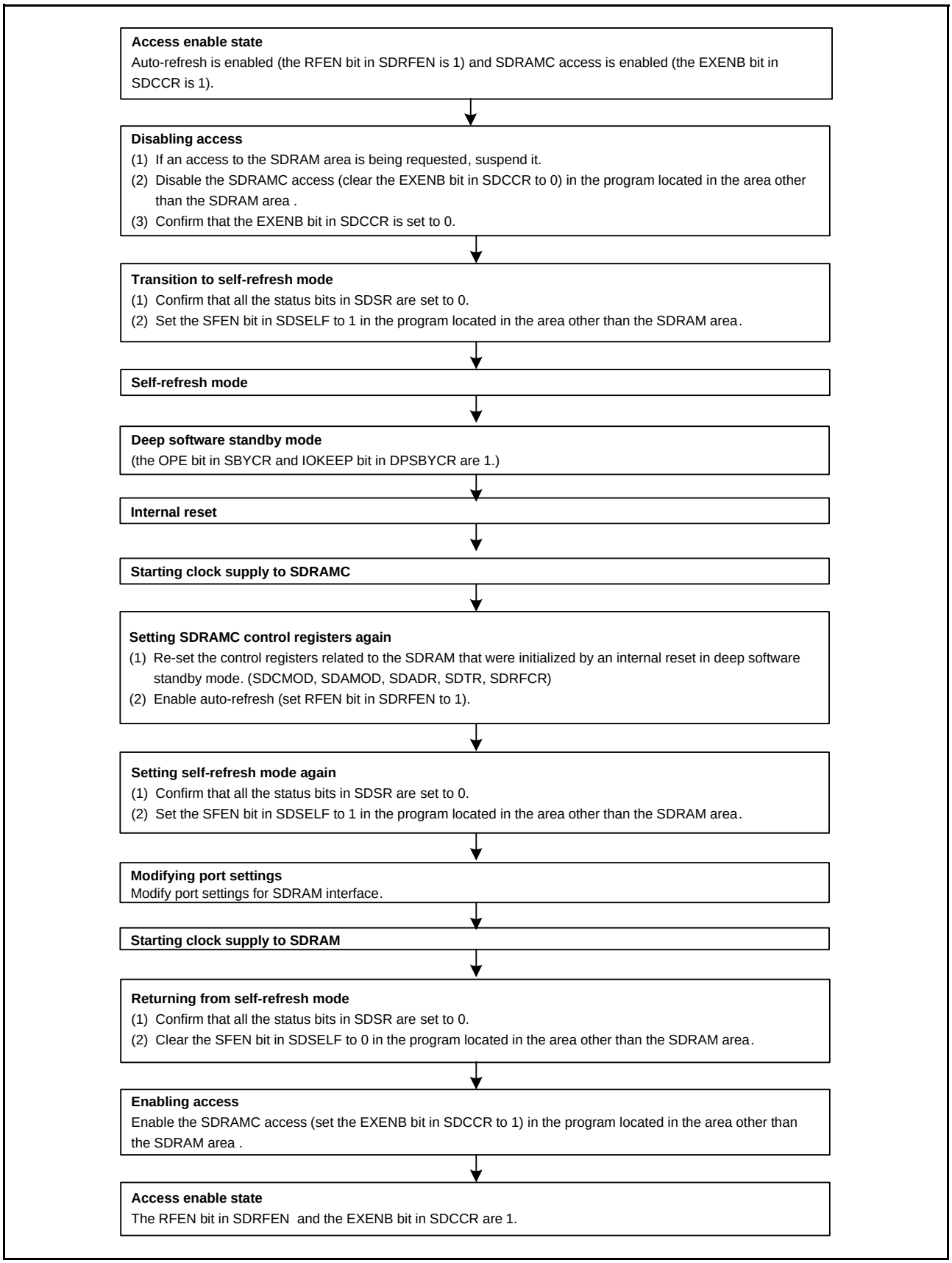


Figure 16.67 Procedure for Transition to and Recovery from Self-Refresh Mode in Deep Software Standby Mode

16.6.12.3 Timing Register Settings and Access Timing

This section describes the relationship between the read/write timing and the settings of the SDRAM timing register (SDTR).

(1) Single Read Timing Examples

Figure 16.68 to Figure 16.72 show the relationship between the single read timing and the SDTR register settings. Table 16.16 shows the correspondence between the figures and the SDTR register settings.
During read access, the next bus access is enabled two cycles after the read data becomes valid at the earliest. However, if two or more accesses occur for one transfer request, the next bus access is enabled one cycle after the read data becomes valid at the earliest, as shown in Figure 16.72.

Table 16.16 Correspondence between Target Figures and SDTR Register Settings (Single Read Timing)

Figure No.	RAS[2:0] Settings	Number of Cycles	RCD[1:0] Settings	Number of Cycles	RP[2:0] Settings	Number of Cycles	CL[2:0] Settings	Number of Cycles
Figure 16.68	010	3	00	1	001	2	010	2
Figure 16.69	000	1	01	2	001	2	010	2
Figure 16.70	000	1	01	2	001	2	011	3
Figure 16.71, Figure 16.72	010	3	00	1	000	1	010	2

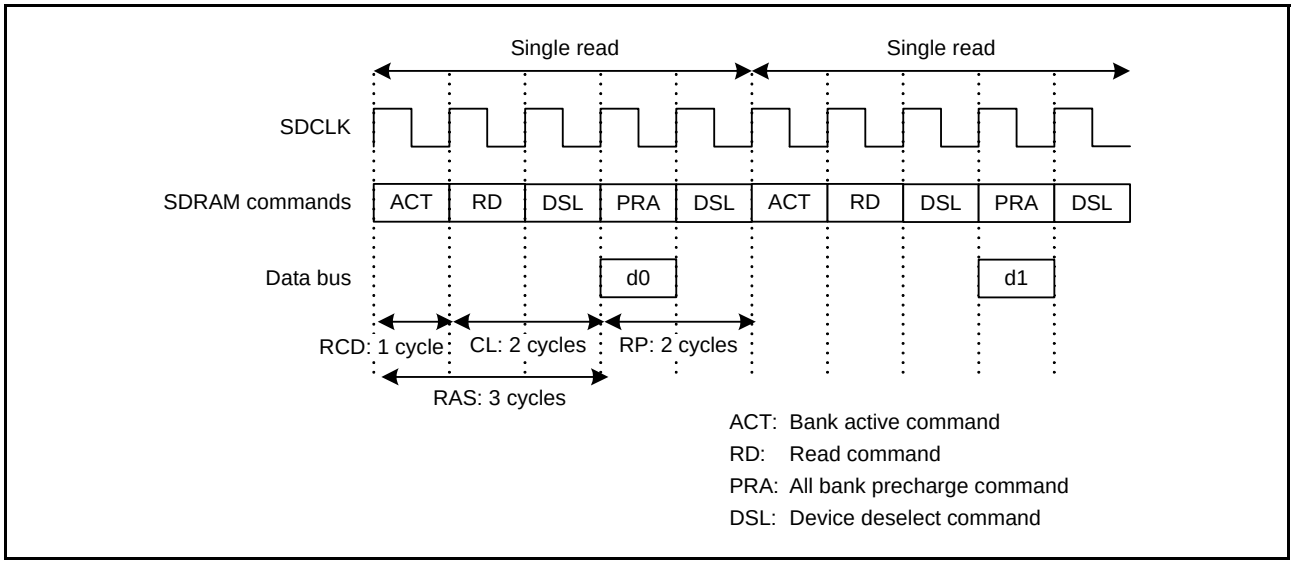


Figure 16.68 Timing Example of Single Read (1)

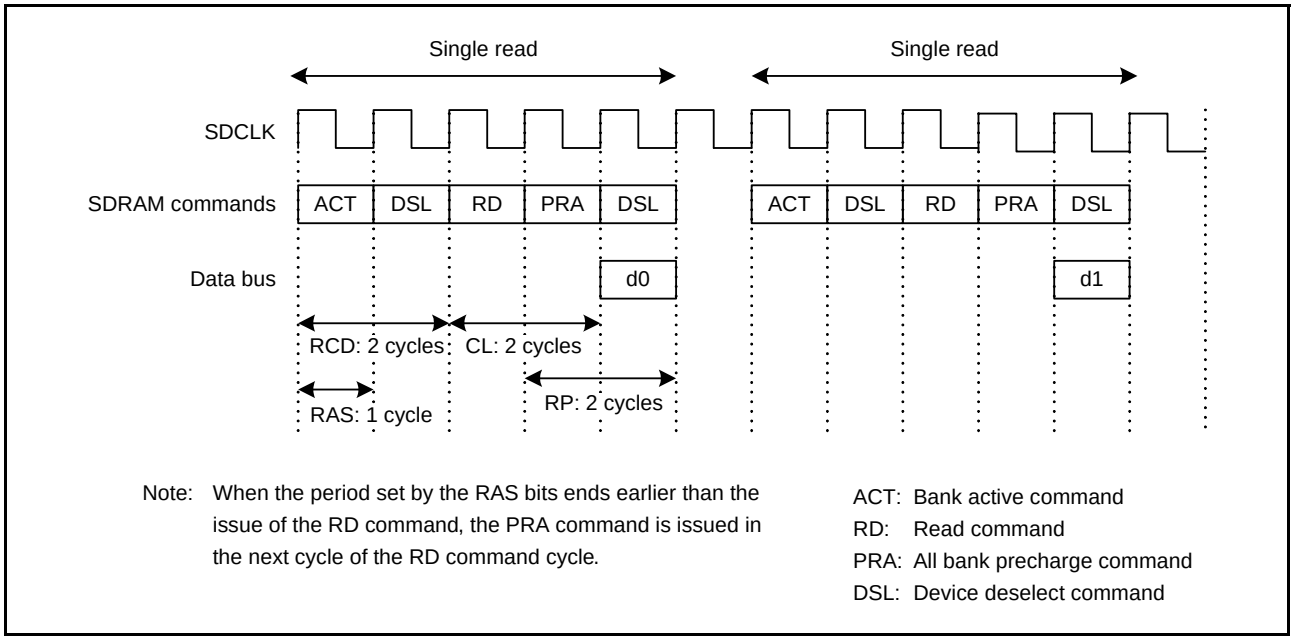


Figure 16.69 Timing Example of Single Read (2)

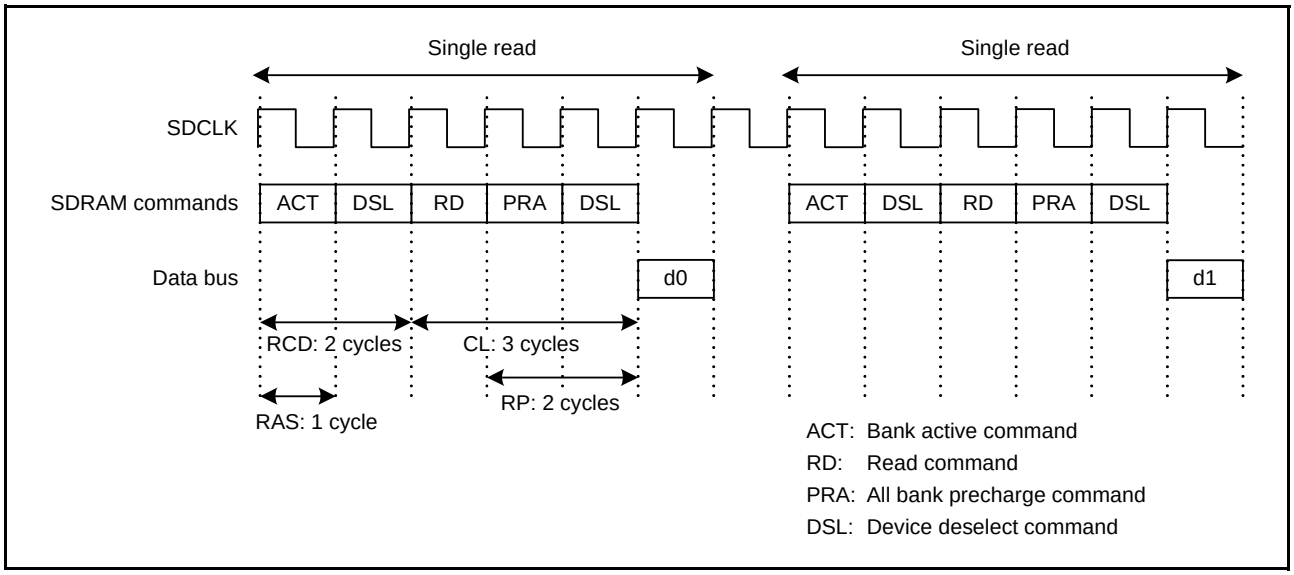


Figure 16.70 Timing Example of Single Read (3)

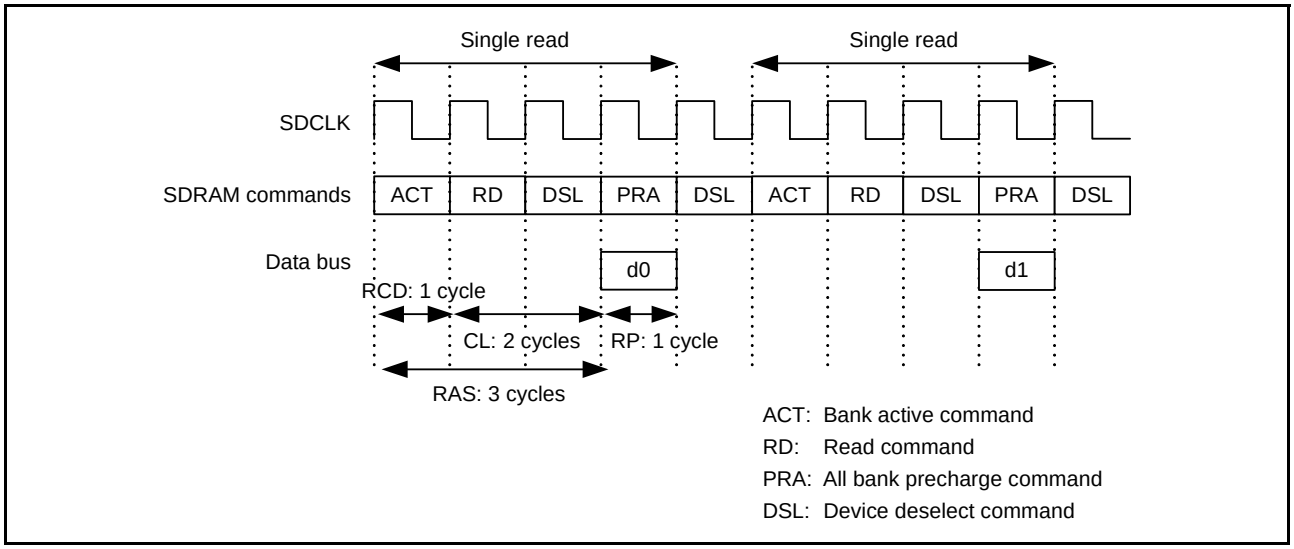


Figure 16.71 Timing Example of Single Read (4)

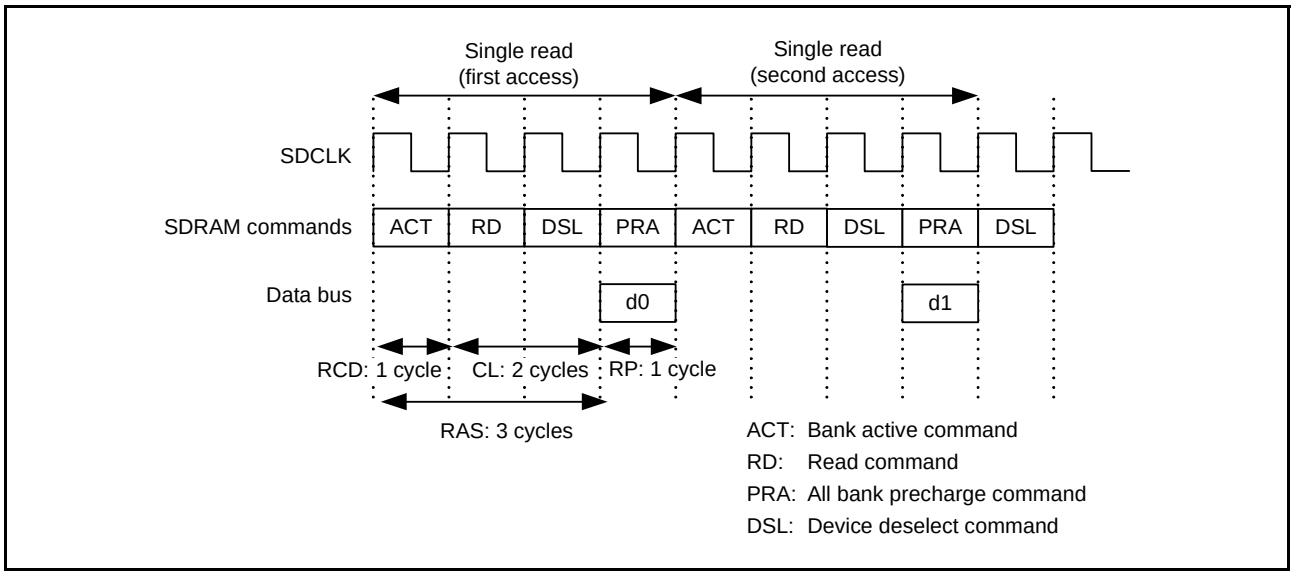


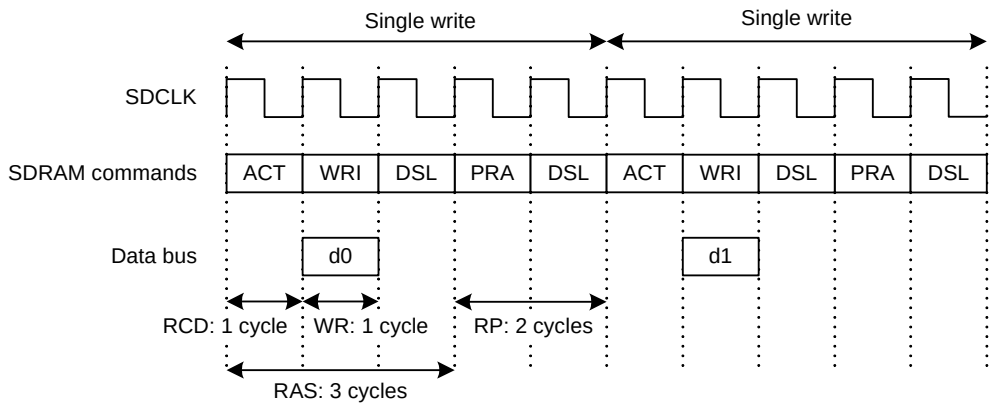
Figure 16.72 Timing Example of Single Read (5) (Two Bus Accesses Occur for One Transfer Request)

(2) Single Write Timing Examples

Figure 16.73 to Figure 16.77 show the relationship between the single write timing and the SDTR register settings. Table 16.17 shows the correspondence between the figures and the SDTR register settings.
During write access, the next bus access is enabled two cycles after an all-bank-precharge command (PRA) is issued at the earliest. However, if two or more accesses occur for one transfer request, the next bus access is enabled one cycle after the PRA is issued at the earliest, as shown in Figure 16.77.

Table 16.17 Correspondence between Target Figures and SDTR Register Settings (Single Write Timing)

Figure No.	RAS[2:0] Settings	Number of Cycles	RCD[1:0] Settings	Number of Cycles	RP[2:0] Settings	Number of Cycles	WR Settings	Number of Cycles
Figure 16.73	010	3	00	1	001	2	0	1
Figure 16.74	000	1	01	2	001	2	0	1
Figure 16.75	000	1	01	2	001	2	1	2
Figure 16.76, Figure 16.77	010	3	00	0	000	2	0	1



Note: When the period specified by the RAS bits is longer than the period specified by WR bits starting from the issue of the WRI command, the period specified by the RAS bits is applied.

ACT: Bank active command
WRI: Write command
PRA: All bank precharge command
DSL: Device deselect command

Figure 16.73 Timing Example of Single Write (1)

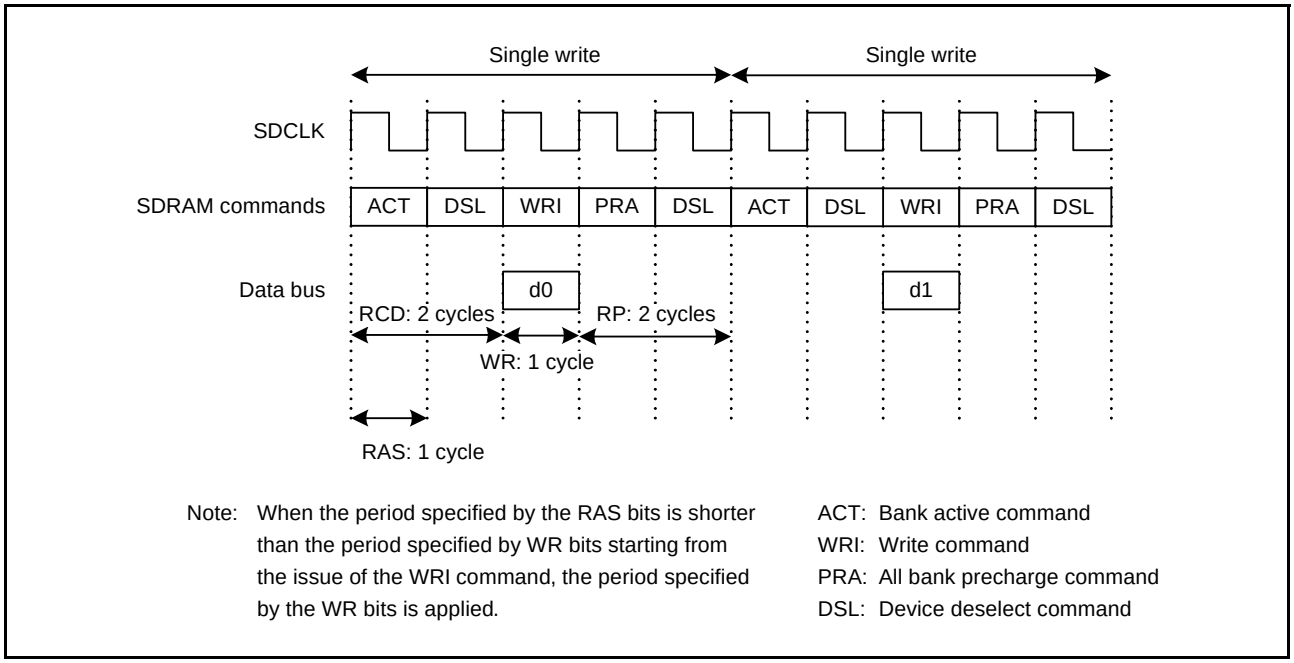


Figure 16.74 Timing Example of Single Write (2)

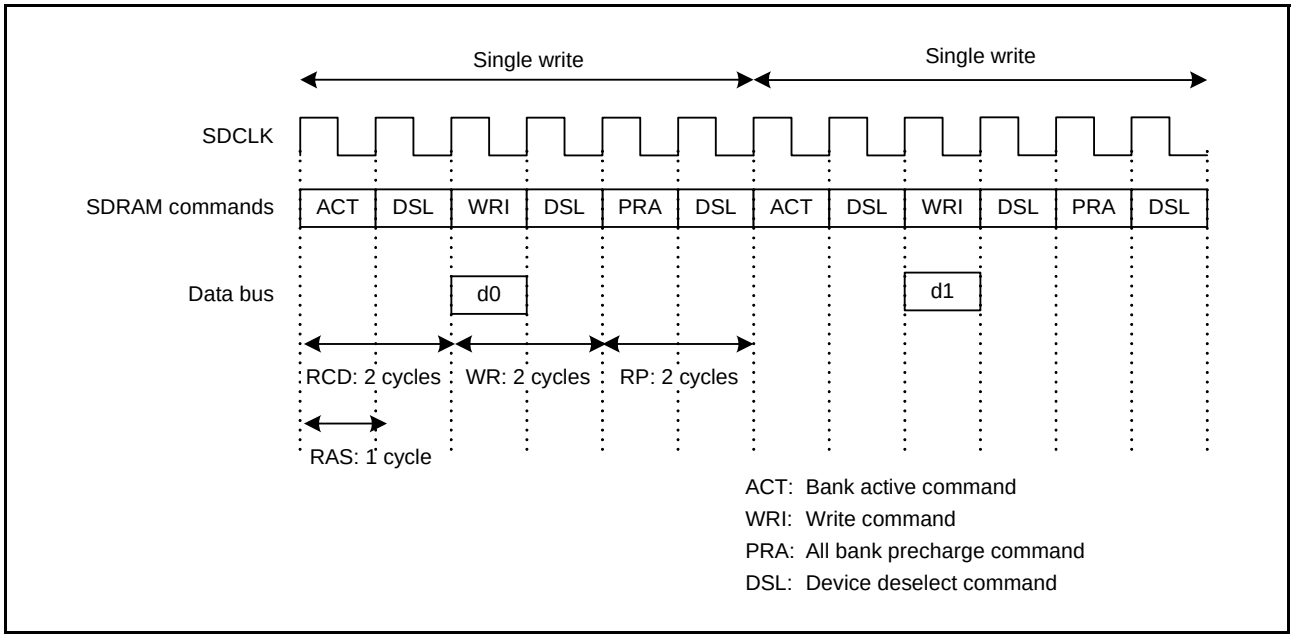
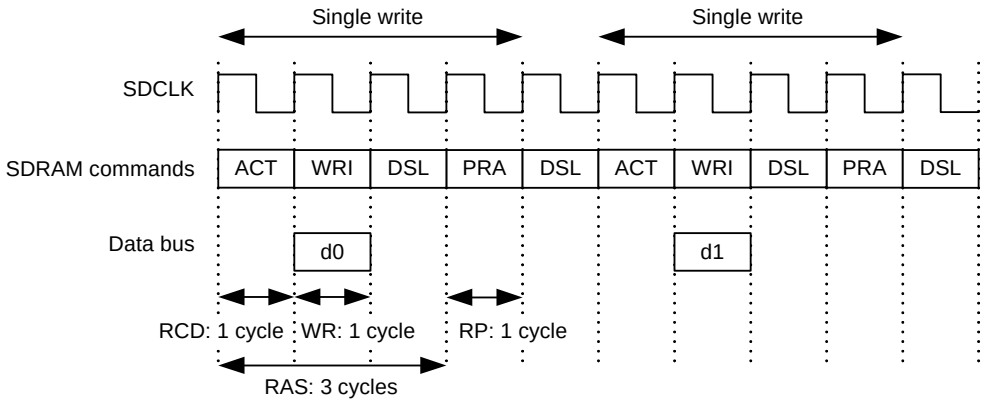


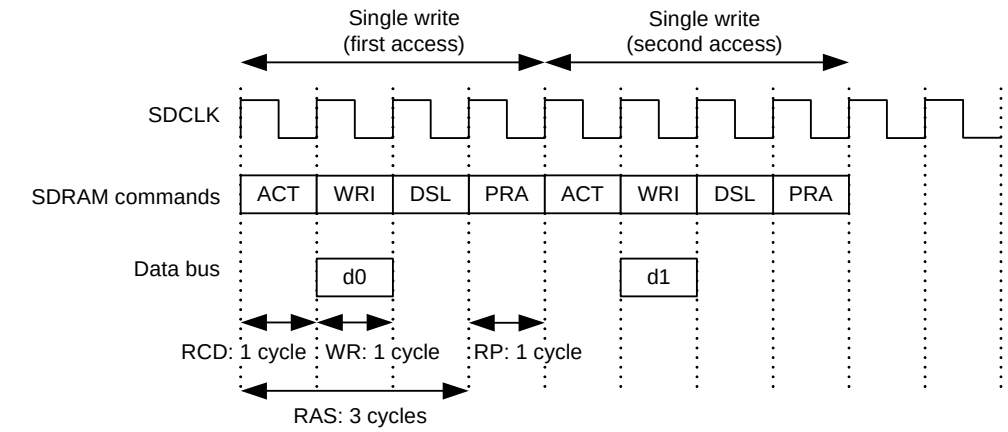
Figure 16.75 Timing Example of Single Write (3)



Note: When the period set by the RAS bits outlasts the WR period, which begins at issuance of the WRI command, the RAS setting is used.

ACT: Bank active command
WRI: Write command
PRA: All bank precharge command
DSL: Device deselect command

Figure 16.76 Timing Example of Single Write (4)



Note: When the period set by the RAS bits is outlasts the WR period, which begins at issuance of the WRI command, the RAS setting is used.

ACT: Bank active command
WRI: Write command
PRA: All bank precharge command
DSL: Device deselect command

Figure 16.77 Timing Example of Single Write (5) (Two Bus Accesses Occur for One Transfer Request)

(3) Consecutive Read Timing Examples

Figure 16.78 to Figure 16.80 show the relationship between the consecutive read timing for four data and the SDTR register settings. Table 16.18 shows the correspondence between the figures and the SDTR register settings.

Table 16.18 Correspondence between Target Figures and SDTR Register Settings (Consecutive Read Timing)

Figure No.	RAS[2:0] Settings	Number of Cycle	RCD[1:0] Settings	Number of Cycle	RP[2:0] Settings	Number of Cycle	CL[2:0] Settings	Number of Cycle
Figure 16.78	010	3	00	1	001	2	010	2
Figure 16.79	000	1	01	2	001	2	010	2
Figure 16.80	000	1	01	2	001	2	011	3

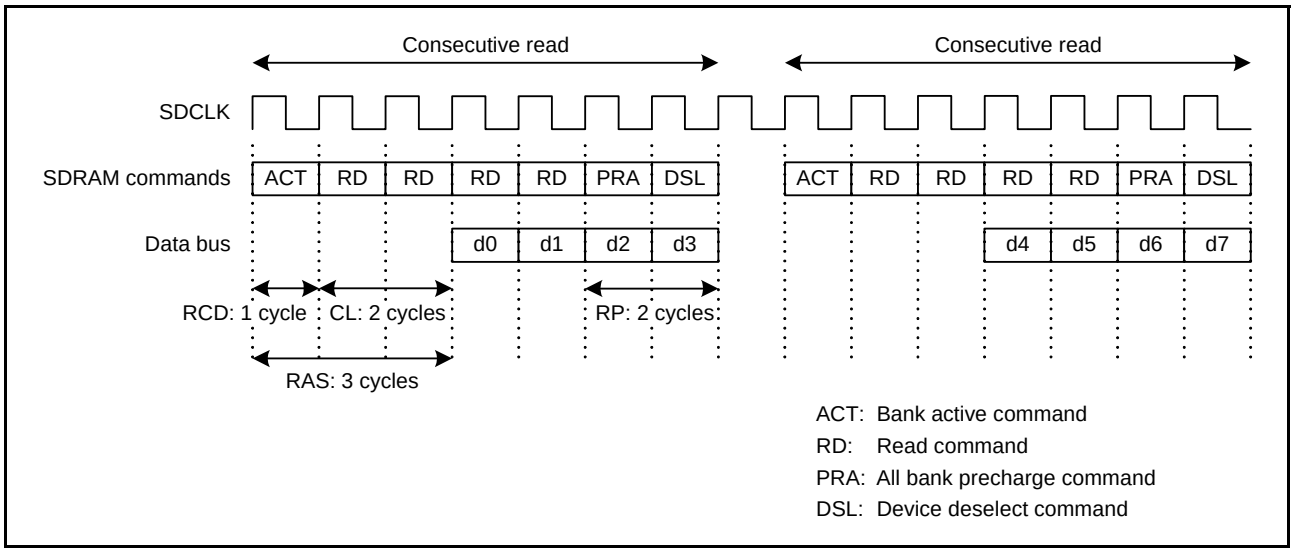


Figure 16.78 Timing Example of Consecutive Read (1)

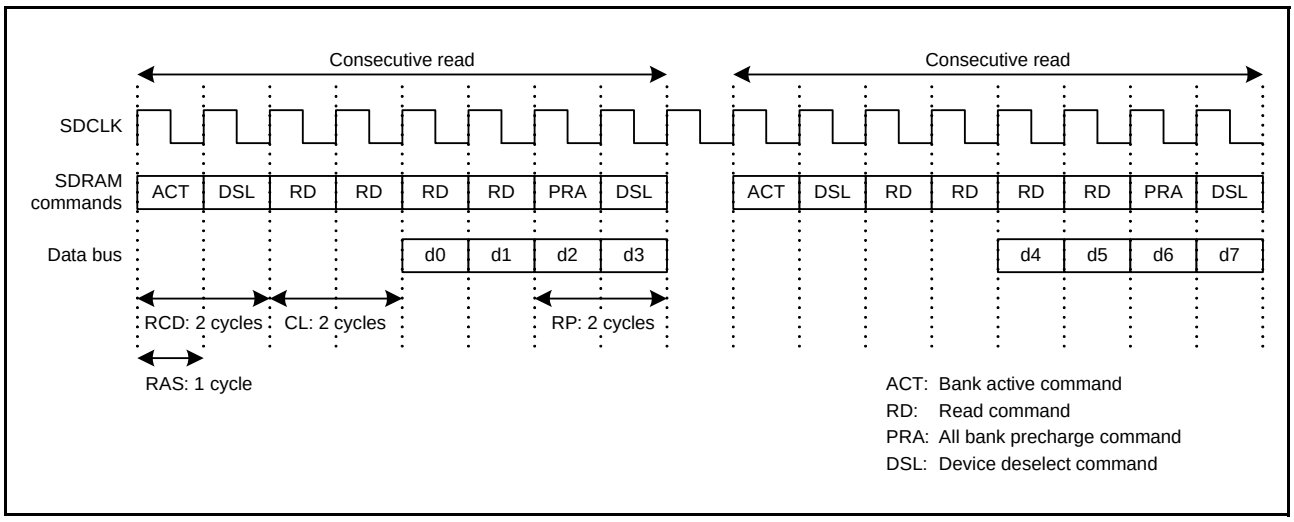


Figure 16.79 Timing Example of Consecutive Read (2)

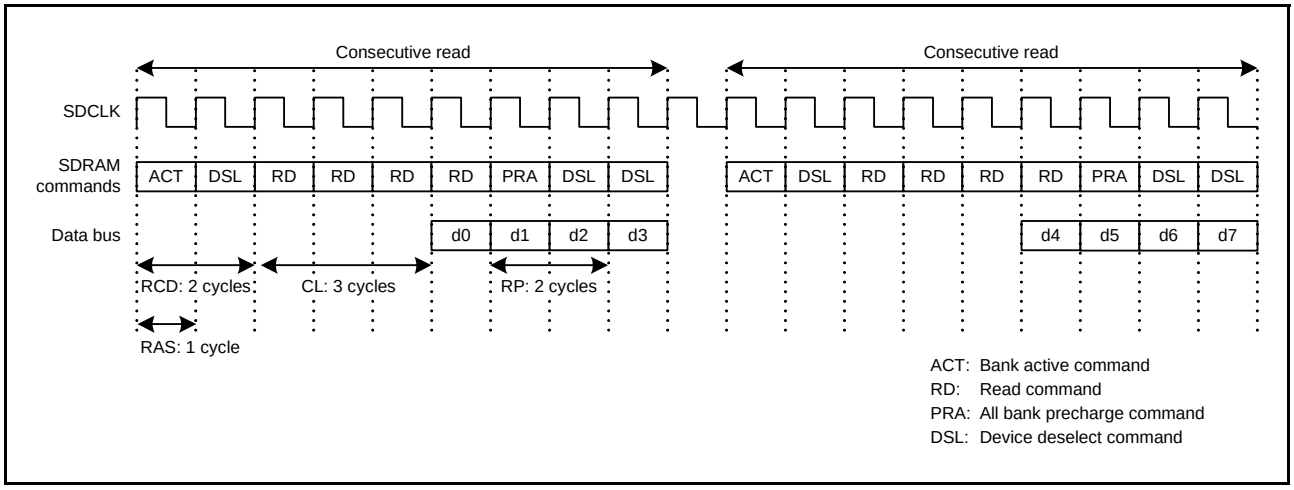


Figure 16.80 Timing Example of Consecutive Read (3)

(4) Consecutive Write Timing Examples

Figure 16.81 to Figure 16.83 show the relationship between the consecutive write timing for four data and the SDTR register settings. Table 16.19 shows the correspondence between the figures and the SDTR register settings.

Table 16.19 Correspondence between Target Figures and SDTR Register Settings (Consecutive Write Timing)

Figure No.	RAS[2:0] Settings	Number of Cycles	RCD[1:0] Settings	Number of Cycles	RP[2:0] Settings	Number of Cycles	WR Settings	Number of Cycles
Figure 16.81	010	3	00	1	001	2	0	1
Figure 16.82	000	1	01	2	001	2	0	1
Figure 16.83	000	1	01	2	001	2	1	2

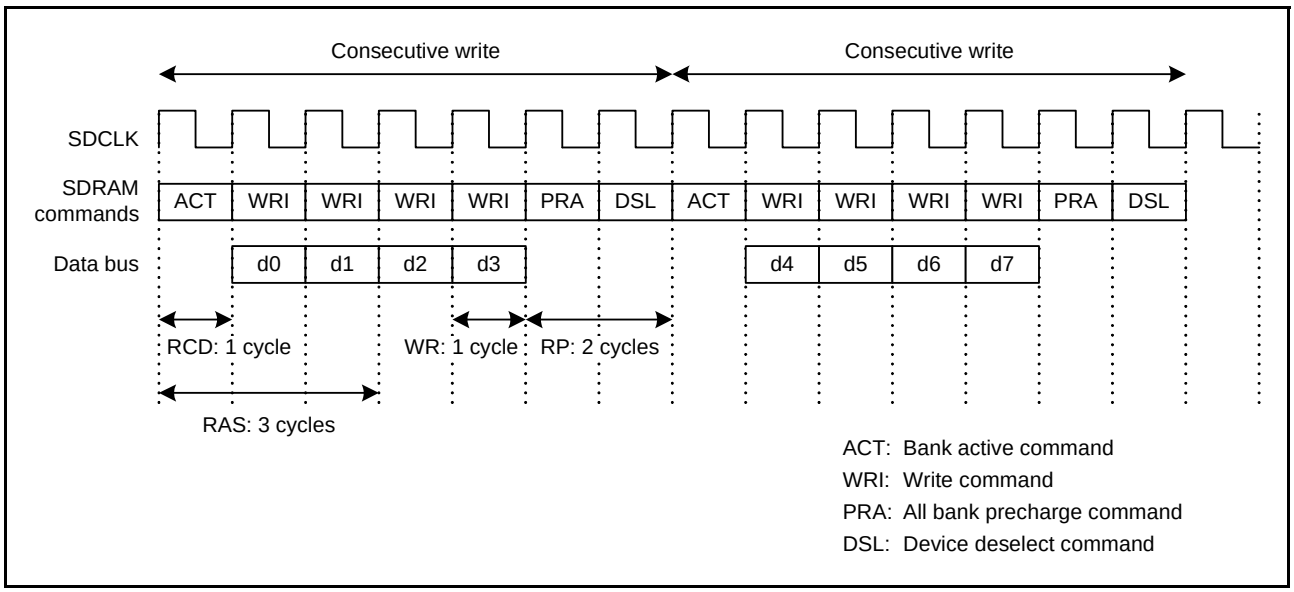


Figure 16.81 Timing Example of Consecutive Write (1)

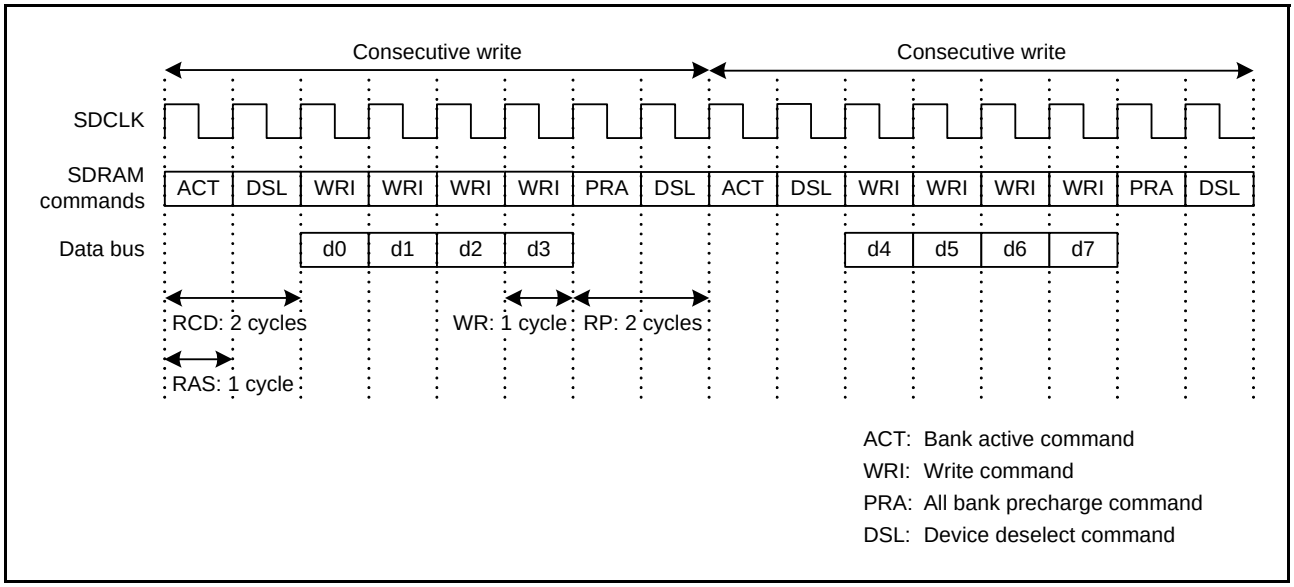


Figure 16.82 Timing Example of Consecutive Write (2)

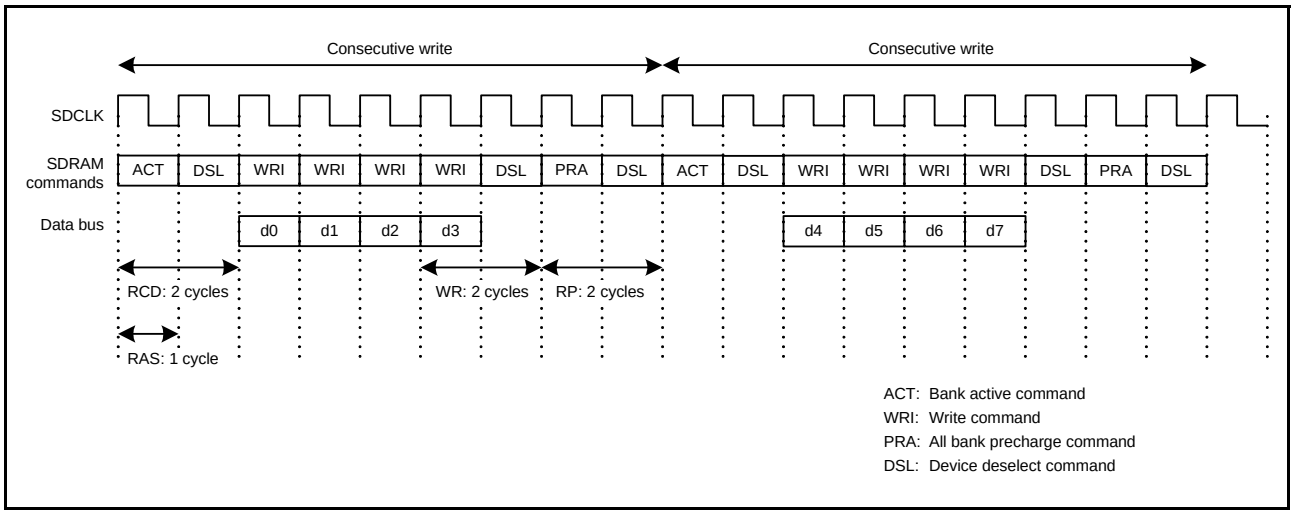


Figure 16.83 Timing Example of Consecutive Write (3)

16.6.13 Address Multiplexing

In the SDRAM space, row and column addresses are multiplexed. The size of the shift in a row address should be specified for address multiplexing by the address multiplex select bits (SDADR.MXC[1:0]) in the SDRAM address register (SDADR). Moreover, in the SDRAM space, the address precharge select command (precharge-sel) is output to the upper bits of column addresses. Table 16.20 shows the relationship between the SDADR.MXC[1:0] settings and the shift amount.

Table 16.20 Address Multiplexing

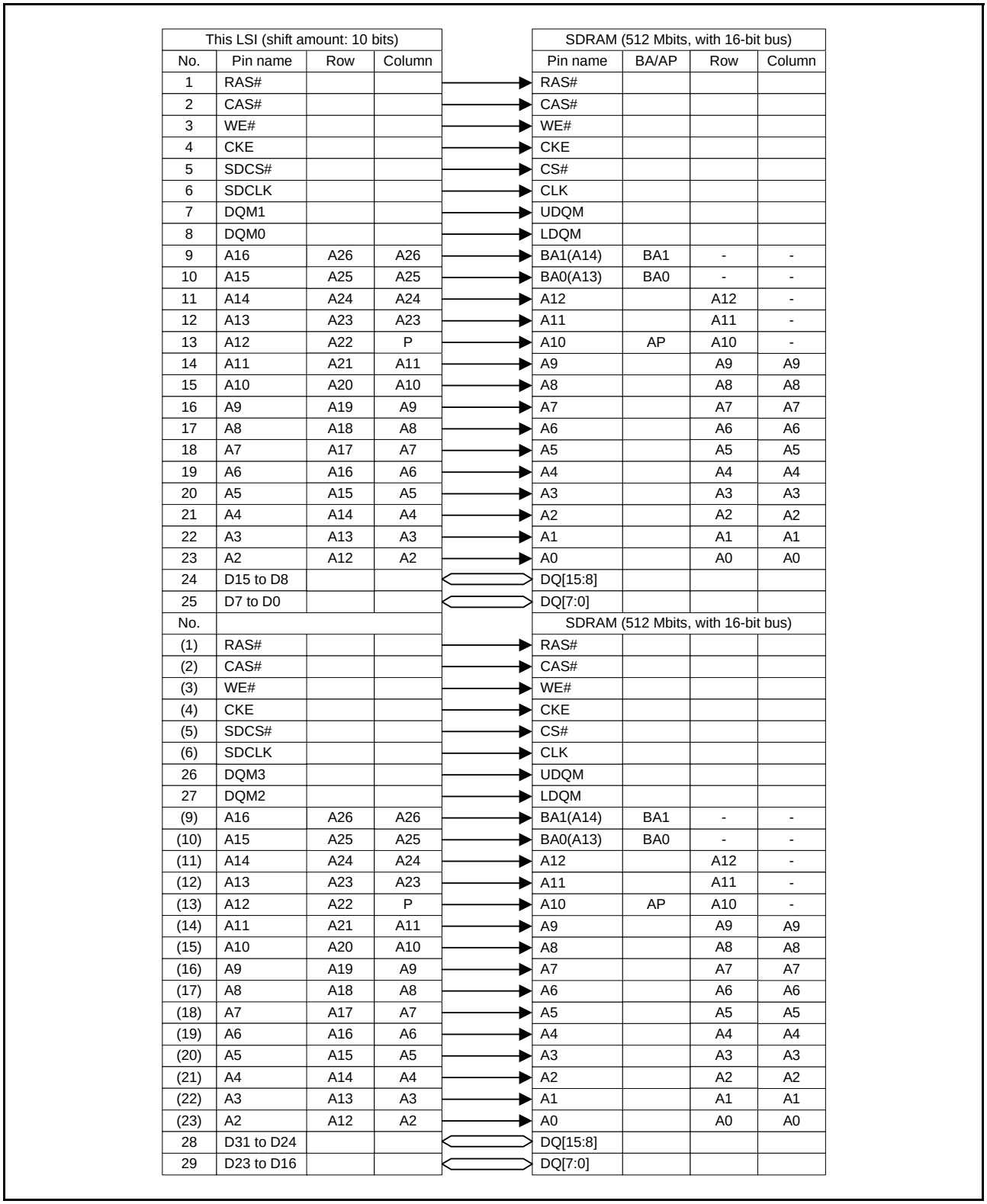
MXC [1:0]	Sift Amount	Data Bus Width	Address	Address Pins External to the Microcomputer																		
				A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
00	8 bits	8 bits	Row	A26	A25	A24	A23	A22	A21	A20	A19	A18*	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8
			Column	A26	A25	A24	A23	A22	A21	A20	A19	P	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		16 bits	Row	A26	A25	A24	A23	A22	A21	A20	A19*	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8
			Column	A26	A25	A24	A23	A22	A21	A20	P	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		32 bits	Row	A26	A25	A24	A23	A22	A21	A20*	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8
			Column	A26	A25	A24	A23	A22	A21	P	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
01	9 bits	8 bits	Row	—	A26	A25	A24	A23	A22	A21	A20	A20*	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9
			Column	—	A26	A25	A24	A23	A22	A21	A20	P	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		16 bits	Row	—	A26	A25	A24	A23	A22	A21	A20*	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9
			Column	—	A26	A25	A24	A23	A22	A21	P	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		32 bits	Row	—	A26	A25	A24	A23	A22	A21*	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9
			Column	—	A26	A25	A24	A23	A22	P	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
10	10 bits	8 bits	Row	—	—	A26	A25	A24	A23	A22	A21	A20*	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
			Column	—	—	A26	A25	A24	A23	A22	A21	P	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		16 bits	Row	—	—	A26	A25	A24	A23	A22	A21*	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
			Column	—	—	A26	A25	A24	A23	A22	P	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		32 bits	Row	—	—	A26	A25	A24	A23	A22*	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
			Column	—	—	A26	A25	A24	A23	P	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
11	11 bits	8 bits	Row	—	—	—	A26	A25	A24	A23	A22	A21*	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
			Column	—	—	—	A26	A25	A24	A23	A10	P	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		16 bits	Row	—	—	—	A26	A25	A24	A23	A22*	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
			Column	—	—	—	A26	A25	A24	A11	P	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
		32 bits	Row	—	—	—	A26	A25	A24	A23*	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
			Column				A26	A25	A12	P	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

Note: P: Precharge setting command (Precharge-sel) is output.
*: When the PALL command is issued, Precharge-sel = 1 (High) is output. When the Active command is issued, the corresponding address is output.
—: Don't care

16.6.14 Examples for Connecting with SDRAMs

16.6.14.1 32-Bit Bus Space

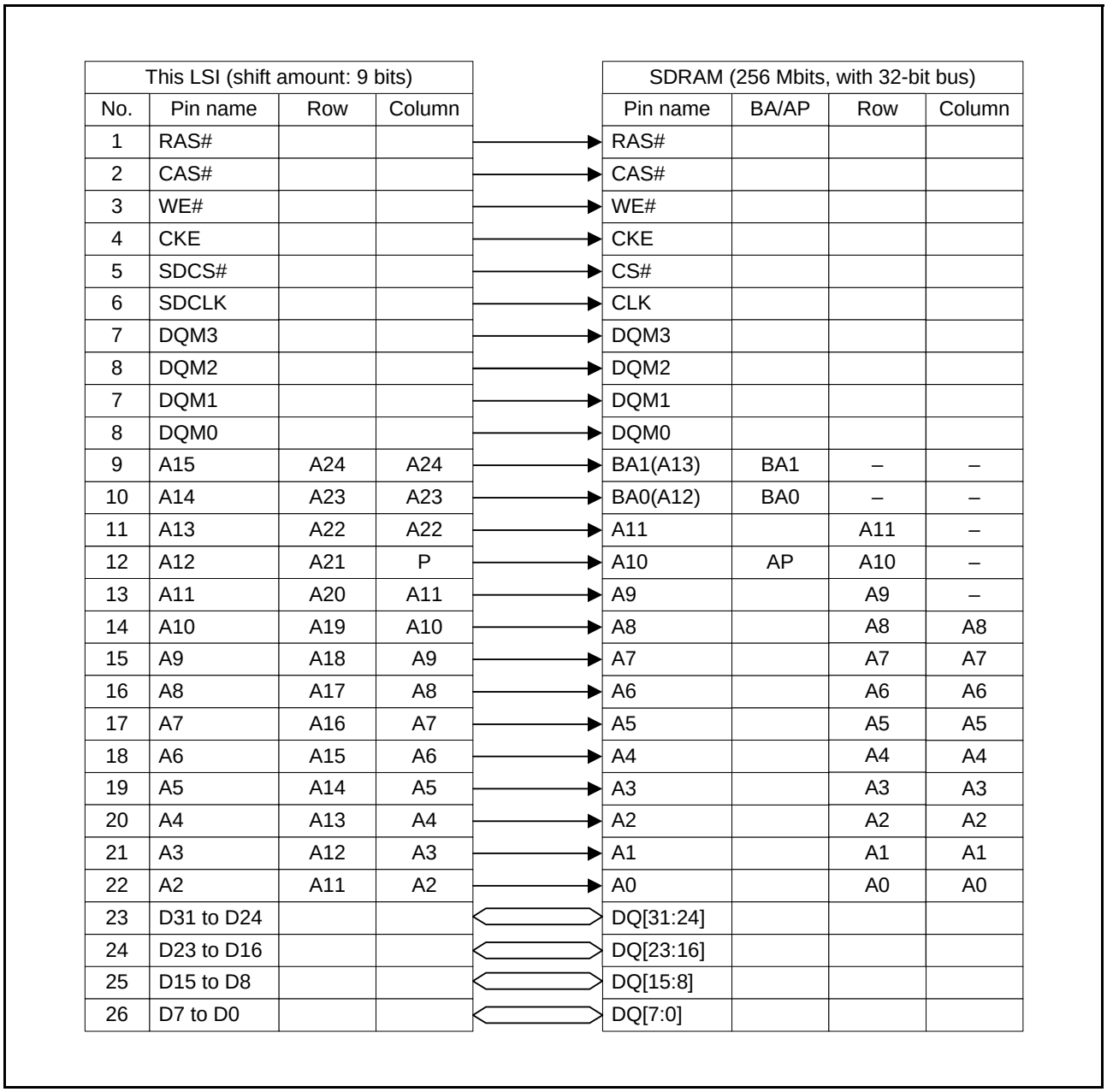
Figure 16.84 shows an example for connecting to two 512-Mbit SDRAMs with 13-bit row address, 10-bit column address and 16-bit bus.



This LSI (shift amount: 10 bits)				SDRAM (512 Mbits, with 16-bit bus)			
No.	Pin name	Row	Column	Pin name	BA/AP	Row	Column
1	RAS#			RAS#			
2	CAS#			CAS#			
3	WE#			WE#			
4	CKE			CKE			
5	SDCS#			CS#			
6	SDCLK			CLK			
7	DQM1			UDQM			
8	DQM0			LDQM			
9	A16	A26	A26	BA1(A14)	BA1	-	-
10	A15	A25	A25	BA0(A13)	BA0	-	-
11	A14	A24	A24	A12		A12	-
12	A13	A23	A23	A11		A11	-
13	A12	A22	P	A10	AP	A10	-
14	A11	A21	A11	A9		A9	A9
15	A10	A20	A10	A8		A8	A8
16	A9	A19	A9	A7		A7	A7
17	A8	A18	A8	A6		A6	A6
18	A7	A17	A7	A5		A5	A5
19	A6	A16	A6	A4		A4	A4
20	A5	A15	A5	A3		A3	A3
21	A4	A14	A4	A2		A2	A2
22	A3	A13	A3	A1		A1	A1
23	A2	A12	A2	A0		A0	A0
24	D15 to D8			DQ[15:8]			
25	D7 to D0			DQ[7:0]			
No.				SDRAM (512 Mbits, with 16-bit bus)			
(1)	RAS#			RAS#			
(2)	CAS#			CAS#			
(3)	WE#			WE#			
(4)	CKE			CKE			
(5)	SDCS#			CS#			
(6)	SDCLK			CLK			
26	DQM3			UDQM			
27	DQM2			LDQM			
(9)	A16	A26	A26	BA1(A14)	BA1	-	-
(10)	A15	A25	A25	BA0(A13)	BA0	-	-
(11)	A14	A24	A24	A12		A12	-
(12)	A13	A23	A23	A11		A11	-
(13)	A12	A22	P	A10	AP	A10	-
(14)	A11	A21	A11	A9		A9	A9
(15)	A10	A20	A10	A8		A8	A8
(16)	A9	A19	A9	A7		A7	A7
(17)	A8	A18	A8	A6		A6	A6
(18)	A7	A17	A7	A5		A5	A5
(19)	A6	A16	A6	A4		A4	A4
(20)	A5	A15	A5	A3		A3	A3
(21)	A4	A14	A4	A2		A2	A2
(22)	A3	A13	A3	A1		A1	A1
(23)	A2	A12	A2	A0		A0	A0
28	D31 to D24			DQ[15:8]			
29	D23 to D16			DQ[7:0]			

Figure 16.84 SDRAM Connection Example (512-Mbit × 2, with 16-Bit Bus)

Figure 16.85 shows an example for connecting to a 256-Mbit SDRAM with 12-bit row address, 9-bit column address and 32-bit bus.



This LSI (shift amount: 9 bits)				SDRAM (256 Mbits, with 32-bit bus)			
No.	Pin name	Row	Column	Pin name	BA/AP	Row	Column
1	RAS#			RAS#			
2	CAS#			CAS#			
3	WE#			WE#			
4	CKE			CKE			
5	SDCS#			CS#			
6	SDCLK			CLK			
7	DQM3			DQM3			
8	DQM2			DQM2			
7	DQM1			DQM1			
8	DQM0			DQM0			
9	A15	A24	A24	BA1(A13)	BA1	–	–
10	A14	A23	A23	BA0(A12)	BA0	–	–
11	A13	A22	A22	A11		A11	–
12	A12	A21	P	A10	AP	A10	–
13	A11	A20	A11	A9		A9	–
14	A10	A19	A10	A8		A8	A8
15	A9	A18	A9	A7		A7	A7
16	A8	A17	A8	A6		A6	A6
17	A7	A16	A7	A5		A5	A5
18	A6	A15	A6	A4		A4	A4
19	A5	A14	A5	A3		A3	A3
20	A4	A13	A4	A2		A2	A2
21	A3	A12	A3	A1		A1	A1
22	A2	A11	A2	A0		A0	A0
23	D31 to D24			DQ[31:24]			
24	D23 to D16			DQ[23:16]			
25	D15 to D8			DQ[15:8]			
26	D7 to D0			DQ[7:0]			

Figure 16.85 SDRAM Connection Example (256-Mbit × 1, with 32-Bit Bus)

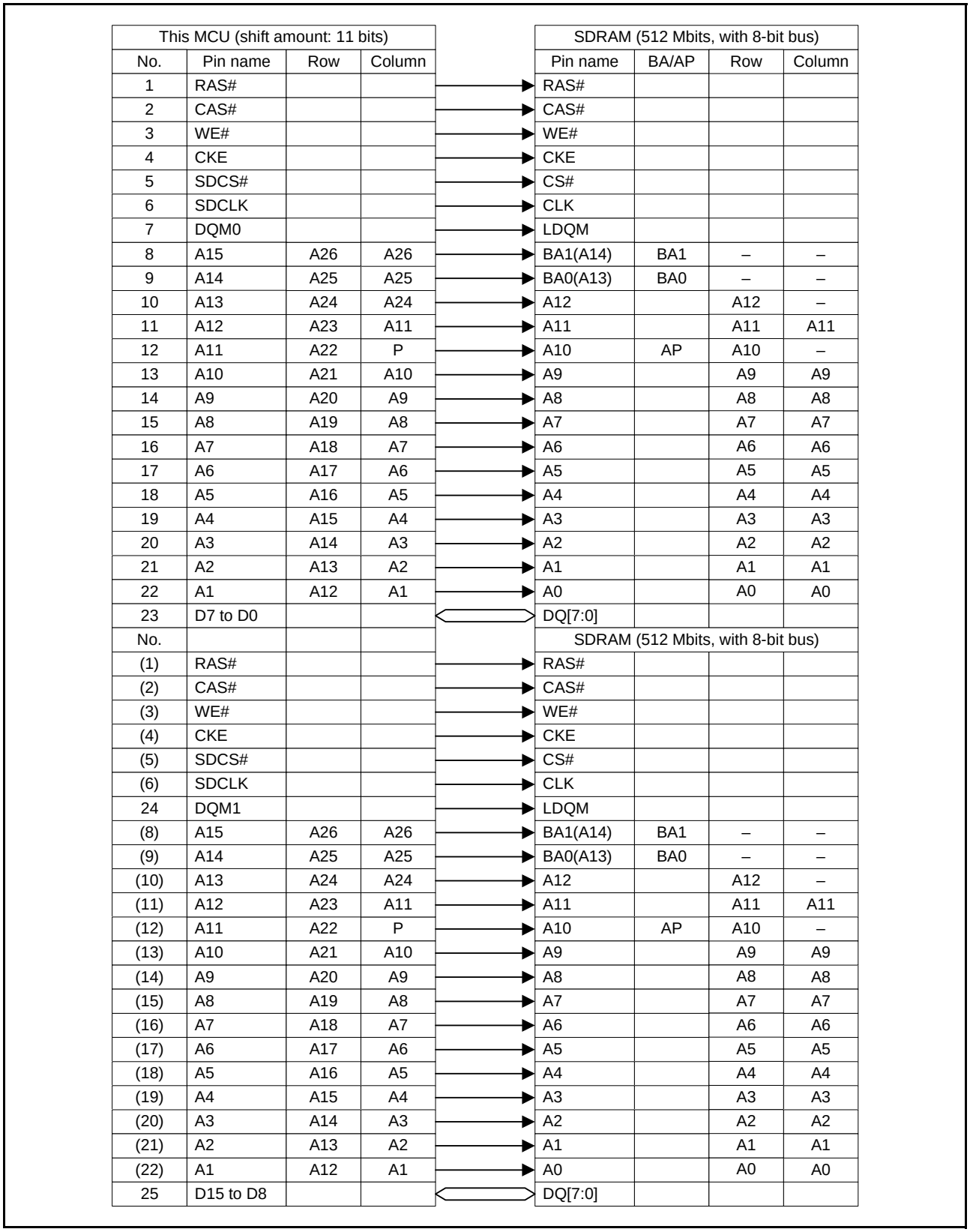
Figure 16.86 shows an example for connecting to two 128-Mbit SDRAMs with 12-bit row address, 9-bit column address and 16-bit bus.

This LSI (shift amount: 9 bits)				SDRAM (128 Mbits, with 16-bit bus)			
No.	Pin name	Row	Column	Pin name	BA/AP	Row	Column
1	RAS#			RAS#			
2	CAS#			CAS#			
3	WE#			WE#			
4	CKE			CKE			
5	SDCS#			CS#			
6	SDCLK			CLK			
7	DQM1			UDQM			
8	DQM0			LDQM			
9	A15	A24	A24	BA1(A13)	BA1	–	–
10	A14	A23	A23	BA0(A12)	BA0	–	–
11	A13	A22	A22	A11		A11	–
12	A12	A21	P	A10	AP	A10	–
13	A11	A20	A11	A9		A9	–
14	A10	A19	A10	A8		A8	A8
15	A9	A18	A9	A7		A7	A7
16	A8	A17	A8	A6		A6	A6
17	A7	A16	A7	A5		A5	A5
18	A6	A15	A6	A4		A4	A4
19	A5	A14	A5	A3		A3	A3
20	A4	A13	A4	A2		A2	A2
21	A3	A12	A3	A1		A1	A1
22	A2	A11	A2	A0		A0	A0
23	D15 to D8			DQ[15:8]			
24	D7 to D0			DQ[7:0]			
No.				SDRAM (128 Mbits, with 16-bit bus)			
(1)	RAS#			RAS#			
(2)	CAS#			CAS#			
(3)	WE#			WE#			
(4)	CKE			CKE			
(5)	SDCS#			CS#			
(6)	SDCLK			CLK			
25	DQM3			UDQM			
26	DQM2			LDQM			
(9)	A15	A24	A24	BA1(A13)	BA1	-	-
(10)	A14	A23	A23	BA0(A12)	BA0	-	-
(11)	A13	A22	A22	A11		A11	-
(12)	A12	A21	P	A10	AP	A10	-
(13)	A11	A20	A11	A9		A9	-
(14)	A10	A19	A10	A8		A8	A8
(15)	A9	A18	A9	A7		A7	A7
(16)	A8	A17	A8	A6		A6	A6
(17)	A7	A16	A7	A5		A5	A5
(18)	A6	A15	A6	A4		A4	A4
(19)	A5	A14	A5	A3		A3	A3
(20)	A4	A13	A4	A2		A2	A2
(21)	A3	A12	A3	A1		A1	A1
(22)	A2	A11	A2	A0		A0	A0
27	D31 to D24			DQ[15:8]			
28	D23 to D16			DQ[7:0]			

Figure 16.86 SDRAM Connection Example (128-Mbit × 2, with 16-Bit Bus)

16.6.14.2 16-Bit Bus Space

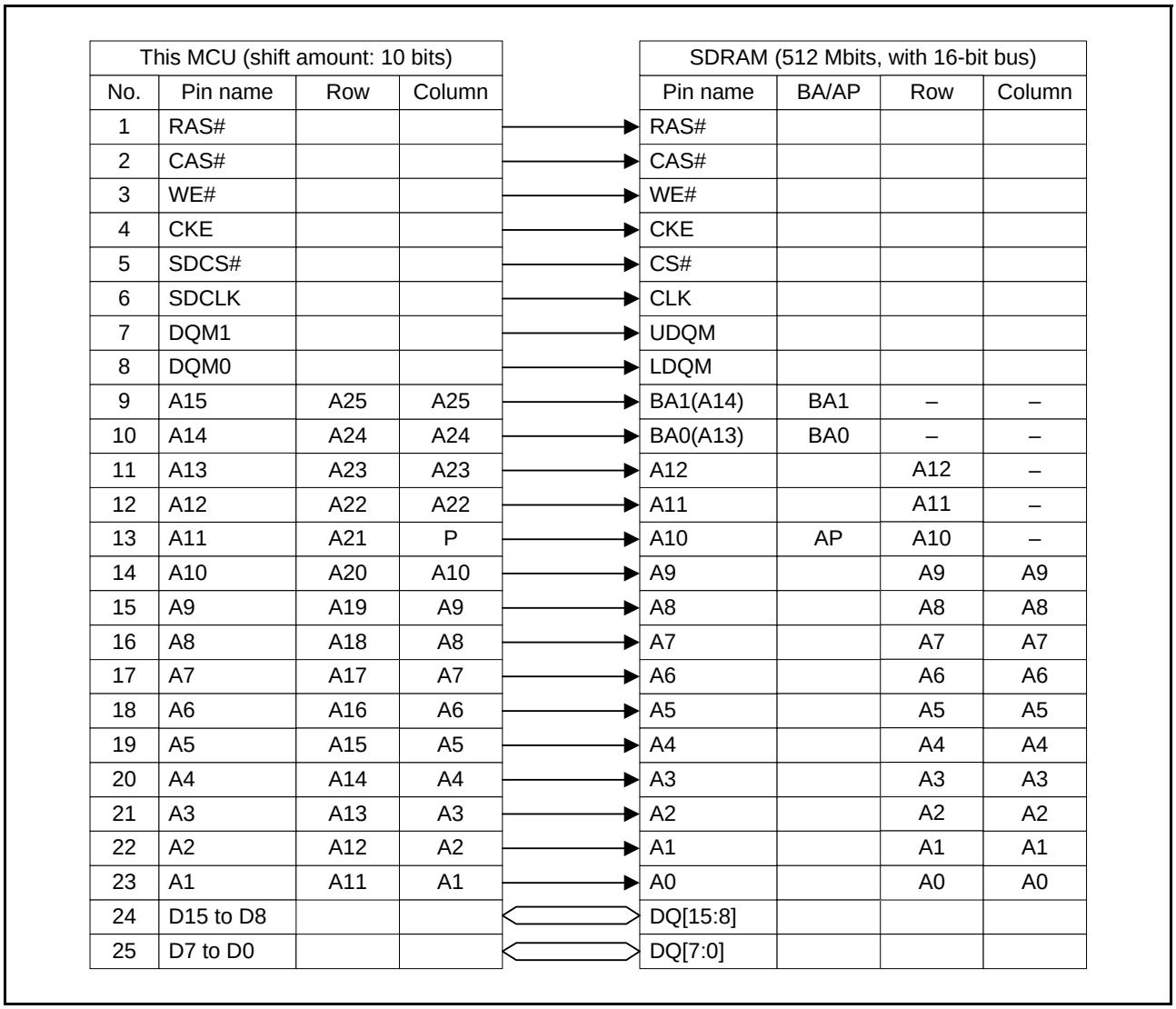
Figure 16.87 shows an example for connecting to two 512-Mbit SDRAMs with 13-bit row address, 11-bit column address and 8-bit bus.



This MCU (shift amount: 11 bits)					SDRAM (512 Mbits, with 8-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS#			→	RAS#			
2	CAS#			→	CAS#			
3	WE#			→	WE#			
4	CKE			→	CKE			
5	SDCS#			→	CS#			
6	SDCLK			→	CLK			
7	DQM0			→	LDQM			
8	A15	A26	A26	→	BA1(A14)	BA1	–	–
9	A14	A25	A25	→	BA0(A13)	BA0	–	–
10	A13	A24	A24	→	A12		A12	–
11	A12	A23	A11	→	A11		A11	A11
12	A11	A22	P	→	A10	AP	A10	–
13	A10	A21	A10	→	A9		A9	A9
14	A9	A20	A9	→	A8		A8	A8
15	A8	A19	A8	→	A7		A7	A7
16	A7	A18	A7	→	A6		A6	A6
17	A6	A17	A6	→	A5		A5	A5
18	A5	A16	A5	→	A4		A4	A4
19	A4	A15	A4	→	A3		A3	A3
20	A3	A14	A3	→	A2		A2	A2
21	A2	A13	A2	→	A1		A1	A1
22	A1	A12	A1	→	A0		A0	A0
23	D7 to D0			⟷	DQ[7:0]			
No.					SDRAM (512 Mbits, with 8-bit bus)			
(1)	RAS#			→	RAS#			
(2)	CAS#			→	CAS#			
(3)	WE#			→	WE#			
(4)	CKE			→	CKE			
(5)	SDCS#			→	CS#			
(6)	SDCLK			→	CLK			
24	DQM1			→	LDQM			
(8)	A15	A26	A26	→	BA1(A14)	BA1	–	–
(9)	A14	A25	A25	→	BA0(A13)	BA0	–	–
(10)	A13	A24	A24	→	A12		A12	–
(11)	A12	A23	A11	→	A11		A11	A11
(12)	A11	A22	P	→	A10	AP	A10	–
(13)	A10	A21	A10	→	A9		A9	A9
(14)	A9	A20	A9	→	A8		A8	A8
(15)	A8	A19	A8	→	A7		A7	A7
(16)	A7	A18	A7	→	A6		A6	A6
(17)	A6	A17	A6	→	A5		A5	A5
(18)	A5	A16	A5	→	A4		A4	A4
(19)	A4	A15	A4	→	A3		A3	A3
(20)	A3	A14	A3	→	A2		A2	A2
(21)	A2	A13	A2	→	A1		A1	A1
(22)	A1	A12	A1	→	A0		A0	A0
25	D15 to D8			⟷	DQ[7:0]			

Figure 16.87 SDRAM Connection Example (512-Mbit × 2, with 8-Bit Bus)

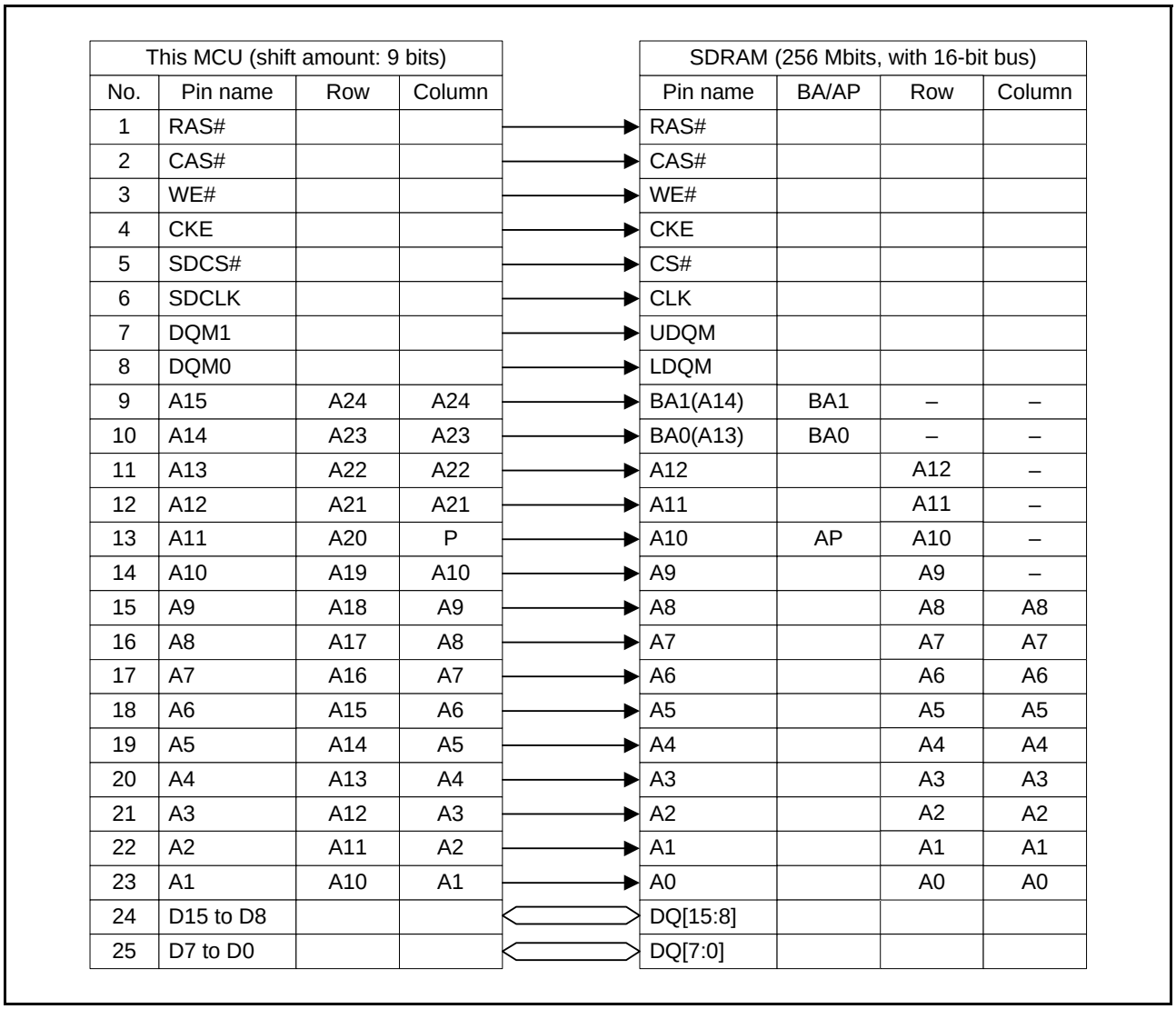
Figure 16.88 shows an example for connecting to a 512-Mbit SDRAM with 13-bit row address, 10-bit column address and 16-bit bus.



This MCU (shift amount: 10 bits)				SDRAM (512 Mbits, with 16-bit bus)			
No.	Pin name	Row	Column	Pin name	BA/AP	Row	Column
1	RAS#			RAS#			
2	CAS#			CAS#			
3	WE#			WE#			
4	CKE			CKE			
5	SDCS#			CS#			
6	SDCLK			CLK			
7	DQM1			UDQM			
8	DQM0			LDQM			
9	A15	A25	A25	BA1(A14)	BA1	–	–
10	A14	A24	A24	BA0(A13)	BA0	–	–
11	A13	A23	A23	A12		A12	–
12	A12	A22	A22	A11		A11	–
13	A11	A21	P	A10	AP	A10	–
14	A10	A20	A10	A9		A9	A9
15	A9	A19	A9	A8		A8	A8
16	A8	A18	A8	A7		A7	A7
17	A7	A17	A7	A6		A6	A6
18	A6	A16	A6	A5		A5	A5
19	A5	A15	A5	A4		A4	A4
20	A4	A14	A4	A3		A3	A3
21	A3	A13	A3	A2		A2	A2
22	A2	A12	A2	A1		A1	A1
23	A1	A11	A1	A0		A0	A0
24	D15 to D8			DQ[15:8]			
25	D7 to D0			DQ[7:0]			

Figure 16.88 SDRAM Connection Example (512-Mbit × 1, with 16-Bit Bus)

Figure 16.89 shows an example for connecting to a 256-Mbit SDRAM with 13-bit row address, 9-bit column address and 16-bit bus.



This MCU (shift amount: 9 bits)				SDRAM (256 Mbits, with 16-bit bus)			
No.	Pin name	Row	Column	Pin name	BA/AP	Row	Column
1	RAS#			RAS#			
2	CAS#			CAS#			
3	WE#			WE#			
4	CKE			CKE			
5	SDCS#			CS#			
6	SDCLK			CLK			
7	DQM1			UDQM			
8	DQM0			LDQM			
9	A15	A24	A24	BA1(A14)	BA1	–	–
10	A14	A23	A23	BA0(A13)	BA0	–	–
11	A13	A22	A22	A12		A12	–
12	A12	A21	A21	A11		A11	–
13	A11	A20	P	A10	AP	A10	–
14	A10	A19	A10	A9		A9	–
15	A9	A18	A9	A8		A8	A8
16	A8	A17	A8	A7		A7	A7
17	A7	A16	A7	A6		A6	A6
18	A6	A15	A6	A5		A5	A5
19	A5	A14	A5	A4		A4	A4
20	A4	A13	A4	A3		A3	A3
21	A3	A12	A3	A2		A2	A2
22	A2	A11	A2	A1		A1	A1
23	A1	A10	A1	A0		A0	A0
24	D15 to D8			DQ[15:8]			
25	D7 to D0			DQ[7:0]			

Figure 16.89 SDRAM Connection Example (256-Mbit × 1, with 16-Bit Bus)

16.6.15 Restrictions

(1) Prohibition of Access that Spans Areas of External Address Space

Single access that spans two areas of the external address space is prohibited, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single word or longword access.
Single access that spans two areas of the address space is also prohibited in the EXDMAC block transfer in single address mode or cluster transfer, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single access in the EXDMAC block transfer in single address mode or cluster transfer.

(2) Restrictions on RMPA and String-Manipulation Instructions

- Although the external space has a per-area ending-switching facility (only for data), the allocation of data to be handled by RMPA or string-manipulation instructions to an area where the endian differs from that of the chip is prohibited, and operation is not guaranteed if this restriction is not observed. If data to be handled by RMPA or string-manipulation instructions are allocated to the external space, they must be allocated to areas where the endian setting is the same as that for the chip.
- The allocation of data to be handled by RMPA or string-manipulation instructions to I/O registers is prohibited, and operation is not guaranteed if this restriction is not observed.

(3) Low Power Consumption State

In all-module clock stop mode, software standby mode, and deep software standby mode, auto-refresh operation is not available since the clock supply to SDRAMC is stopped. To retain the data in the SDRAM when the SDRAM is externally connected, use the self-refresh function. For the procedure for transition to and recovery from self-refresh mode, see section 16.6.7, Self-Refresh.

(4) Consecutive-Access Mode

For block transfer or cluster transfer by the EXDMAC in single-address mode, the setting CL = 1 is prohibited, and operation is not guaranteed if this setting is made.

(5) Setting the SDRAM Timing Register

Set the RAS[2:0] bits in the SDRAM timing register (SDTR) to a value less than or equal to the sum of the row column latency (SDTR.RCD[1:0]) and column latency (SDTR.CL[2:0]) settings. Operation is not guaranteed if this condition is not satisfied.

(6) Restriction on Instruction Code

When the endian setting for each area is different from that for the chip, no instruction code can be allocated in the area. The instruction code should be allocated to the external space whose endian setting is the same as that for the chip.

16.7 Bus Error Monitoring Section

The bus error monitoring section monitors the individual areas for bus errors, and when a bus error occurs, the error is indicated to the bus master.

16.7.1 Types of Bus Error

There are two types of bus error: illegal address access and timeout.
Illegal address access is the detection of illegal access to an area, and time-out is the detection of a bus-access operation not being completed within 768 cycles.

16.7.1.1 Illegal Address Access

When the illegal address access detection enable bit (IGAEN) in the bus error monitoring enable register (BEREN) is set to 1, access of the following types leads to illegal address access errors.

- Access to areas of external space for which operation has been disabled (CSnCR.EXENB = 0, SDCCR.EXENB = 0)
- With respect to areas other than those described above, access to illegal address ranges
  The address ranges where access will lead to illegal address access errors are indicated in Table 16.21.

16.7.1.2 Timeout

When the timeout detection enable bit (TOEN) in the bus error monitoring enable register (BEREN) is set to 1, bus access that is not completed within 768 cycles leads to a timeout error.

- CS areas (CS0 to CS7): Bus access is not completed (the WAIT# signal is not negated) within 768 external bus clock (BCLK) cycles from the start of the access.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 BCLK cycles. If multiple external bus accesses are generated with a single request from the bus master during the transfer, the bus accesses cannot be stopped by a timeout. At this time, timeout errors may occur repeatedly.
- Internal peripheral buses (2 and 3): Bus access is not completed within 768 peripheral module clock (PCLKB) cycles from the start of the access. In this MCU, a timeout error does not occur.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 PCLKB cycles.
- Internal peripheral buses (4): Bus access is not completed within 768 peripheral module clock (PCLKA) cycles from the start of the access.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 PCLKA cycles. If either of the MSTPB15 and MSTPB14 bits is cleared, set the BEREN.TOEN bit to 1.
- Internal peripheral bus (6): Bus access is not completed within 768 FlashIF clock (FCLK) cycles from the start of the access.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 FCLK cycles. In this MCU, a timeout error does not occur.

16.7.2 Operations When a Bus Error Occurs

When a bus error occurs, the error is indicated to the CPU. Operation is not guaranteed when a bus error occurs.

- Bus error indication to the CPU
  An interrupt is generated. The IERn register in the ICU can specify whether to generate an interrupt in the case of a bus error.

16.7.3 Conditions Leading to Bus Errors

Table 16.21 lists the types of bus errors for each area in the respective address space.
If an illegal address access error or timeout is detected when no bus error has occurred (bus error status register n (BERSRn; n = 1 or 2) is cleared), the detected error is reflected on the BERSRn. Once a bus error occurs, no subsequent bus errors are reflected on the register unless the register is cleared.
If bus errors are simultaneously caused by two or more bus masters, error information of only one bus master is reflected. Once a bus error occurs, the status is retained until BERSRn is cleared.

Table 16.21 Types of Bus Errors

Address	Type of Area		Type of Error			
			Illegal Address Access		Timeout	
	Code Flash Memory		Code Flash Memory		Code Flash Memory	
	Enabled	Disabled	Enabled	Disabled	Enabled	Disabled
0000 0000h to 0007 FFFFh	Memory bus 1		—		—	
0008 0000h to 0008 7FFFh	Internal peripheral bus 1		—		—	
0008 8000h to 0009 FFFFh	Internal peripheral bus 2		Δ		—	
000A 0000h to 000B FFFFh	Internal peripheral bus 3		Δ		—	
000C 0000h to 000D FFFFh	Internal peripheral bus 4		Δ		○	
000E 0000h to 000F FFFFh	Reserved area		—		—	
0010 0000h to 0011 FFFFh	Internal peripheral bus 6	Reserved area	—	○	—	—
0012 0000h to 007F FFFFh			Δ	○	—	—
0080 0000h to 00FF FFFFh	Memory bus 3		—	—	—	—
0100 0000h to 07FF FFFFh	External bus (CS1 to CS7)		[IA]		[TO]	
0800 0000h to 0FFF FFFFh	External bus (SDRAM area)		[IA]		—	—
1000 0000h to 7FFF FFFFh	Reserved area		○		—	—
8000 0000h to FEFF FFFFh	Memory bus 2	Reserved area	—	○	—	—
FF00 0000h to FF7F FFFFh		External bus (CS0)	—	[IA]	—	[TO]
FF80 0000h to FFFF FFFFh			—		—	

—: A bus error does not result.
Δ: A bus error may or may not result.
○: A bus error results.
[IA]: Access to this area leads to detection of a bus error if operation for this area is disabled (CSnCR.EXENB = 0; n = 0 to 7, SDCCR.EXENB = 0).
[TO]: Bus access not being completed within 768 cycles leads to detection of a bus error.
Note: The capacity of the RAM, data flash memory, and code flash memory differs depending on the product. For details, see section 61, RAM, section 63, Flash Memory.

16.8 Interrupt

16.8.1 Interrupt Source

An illegal address access error or detection of a timeout leads to a bus error signal for the interrupt controller.

Table 16.22 Interrupt Source

Name	Interrupt Source	DTC Trigger	DMAC Trigger
BUSERR	Illegal address access error or timeout	Not possible	Not possible

16.9 Usage Notes

16.9.1 Setting Drive Capacity Control Registers for I/O Ports

When BCLK = 30 MHz or higher, set the Pmn drive capacity control bit (m = 2, 5, 9, A to E, G; n = 0 to 7) in the drive capacity control register (DSCR) to high-drive output for the I/O ports used for buses.

17. Memory-Protection Unit (MPU)

17.1 Overview

The RXv2 CPU incorporates a memory-protection unit that checks the addresses of CPU access to the overall address space (0000 0000h to FFFF FFFFh).

Access-control information can be set for up to eight regions, and permission for access to each region is in accord with this information. The default response to the detection of access to a region where permission has not been set is the generation of a memory-protection error.

The supported access-control information for the individual regions consists of permission to read, permission to write, and permission to execute. This access-control information is effective when the processor mode of the CPU is user mode. Memory protection is not applied when the CPU is in supervisor mode.

Table 17.1 lists the specifications of the memory-protection unit, and Figure 17.1 shows a block diagram of the memory-protection unit.

Table 17.1 Specifications of Memory Protection

Specifictaions	Description
Region to be covered by memory protection and processor mode	0000 0000h to FFFF FFFFh (in user mode) No memory protection in supervisor mode
Number of regions	8
Page size (smallest unit of protection)	16 bytes
Specifying addresses of individual regions	Setting the page numbers where regions start and end
Setting to make memory protection effective or ineffective in individual regions	A V (valid) bit in each region-n end page number register (REPAGEn) makes the settings effective or ineffective for the corresponding region (n = 0 to 7).
Access-control information settings for individual regions	Instruction execution: Permission to execute Operand access: Permission to read, permission to write
Start of memory-protection operations	After the memory-protection unit has been enabled, access monitoring starting up with the transition to user mode.
Memory-protection error processing	Generation of access exceptions
Addresses where memory-protection errors are generated	Address in instruction execution: The PC value is preserved on the stack. Address in operand access: The address is stored in the data memory-protection error address register (MPDEA).
Determining the reasons for memory-protection errors	The memory-protection error status register (MPESTS) holds indicators of the reason.
Background region setting	Access-control information can be set for the background region (the whole address space).
Processing where regions overlap	The access-control information for access to an overlap between regions is the logical OR of the attributes for the given regions, and permission is given priority.

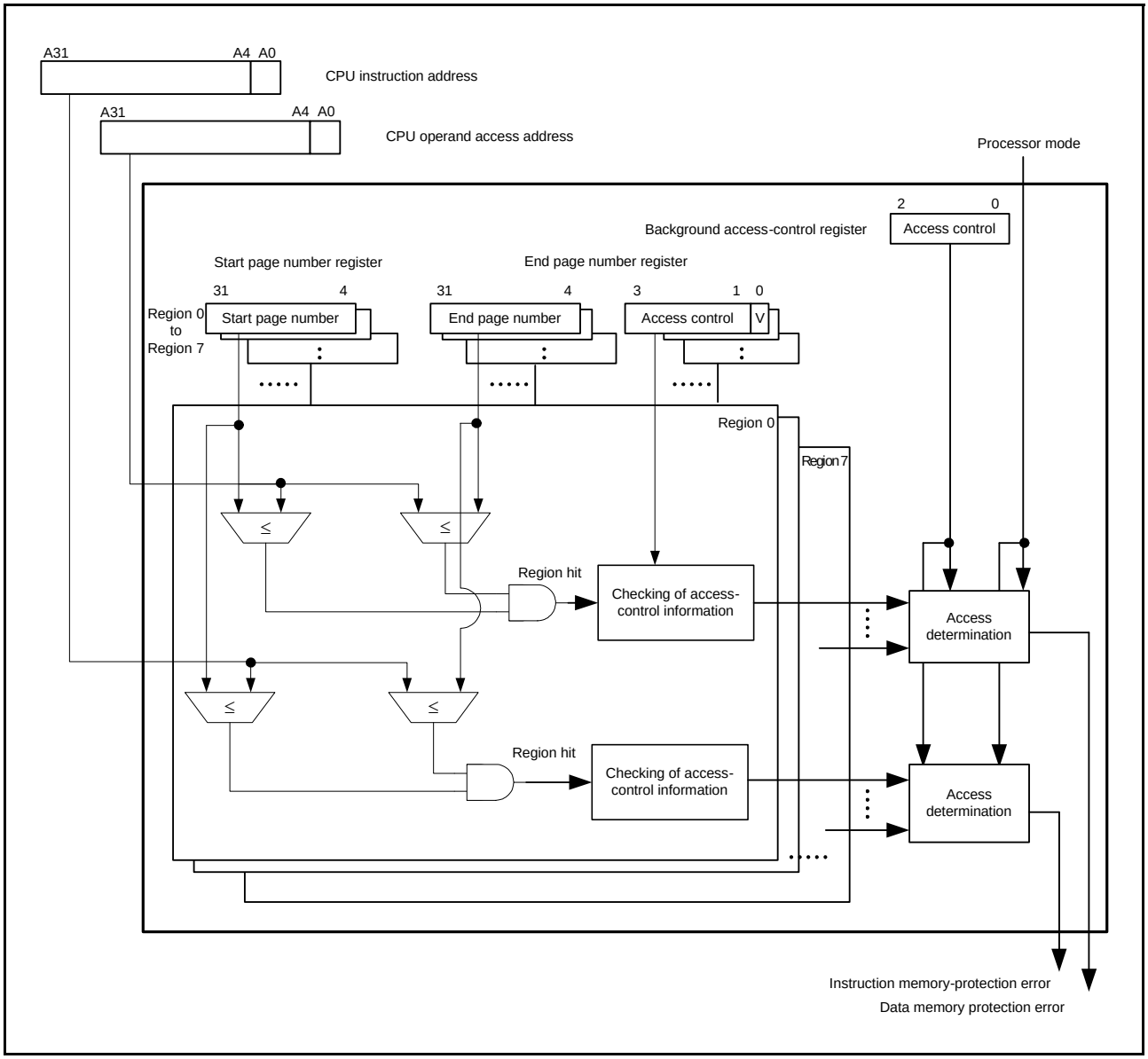


Figure 17.1 Block Diagram of the Memory-Protection Unit

17.1.1 Types of Access Control

There are three types of access control information: permission for instruction execution, permission to read operands, and permission to write operands. Violations of these types of access control are only detected when programs are running in user mode. Violations are not detected when programs are running in supervisor mode.

17.1.2 Regions for Access Control

Up to eight regions for access control are definable. Settings of the range of memory for each access-control region are made in the corresponding region-n start page number register (RSPAGEn) and region-n end page number register (REPAGEn), where n = 0 to 7.
The minimum unit for control of access is the “page”, by which the address space is divided into 16-byte units. The 28 higher-order bits ([31:4]) of the address [31:0] bits correspond to the page number.
The REPAGEn register specifies the access-control information for each area and whether the area is enabled or not.

17.1.3 Background Region

“Background region” refers to the whole address space (0000 0000h to FFFF FFFFh). Access-control information for the background region is set in the background-region access-control register (MPBAC). In contrast to the access-control information for the eight individual regions, protection information for the background region is effective as long as memory protection is enabled (the MPEN bit in the MPEN register is 1).

17.1.4 Overlap between Regions

In cases of overlap between multiple regions, the access-control information becomes the logical OR of the access-control bits for the overlapping regions (including the background region), with permission given priority.

17.1.5 Instructions and Data that Span Regions

Operations in response to the detection of memory-protection errors when instructions or data span regions for which different access-control settings have been made are undefined. Ensure that instructions and data do not span regions for which different access-control settings have been made.

17.2 Register Descriptions

17.2.1 Region-n Start Page Number Register (RSPAGEn) (n = 0 to 7)

Address(es): RSPAGE0 0008 6400h, RSPAGE1 0008 6408h, RSPAGE2 0008 6410h, RSPAGE3 0008 6418h, RSPAGE4 0008 6420h, RSPAGE5 0008 6428h, RSPAGE6 0008 6430h, RSPAGE7 0008 6438h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RSPN[27:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RSPN[27:0]												—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Function	R/W
b3 to b0	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b31 to b4	RSPN[27:0]	Region-Start Page Number	Page number where the region starts, for use in region determination	R/W

RSPN[27:0] Bits (Region-Start Page Number)

These bits specify the page number where the region starts.

17.2.2 Region-n End Page Number Register (REPAGEn) (n = 0 to 7)

Address(es): REPAGE0 0008 6404h, REPAGE1 0008 640Ch, REPAGE2 0008 6414h, REPAGE3 0008 641Ch, REPAGE4 0008 6424h, REPAGE5 0008 642Ch, REPAGE6 0008 6434h, REPAGE7 0008 643Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	REPN[27:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REPN[27:0]												UAC[2:0]			V
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Function	R/W
b0	V	Valid Bit	0: Region setting invalid 1: Region setting valid	R/W
b3 to b1	UAC[2:0]	Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R/W
b31 to b4	REPN[27:0]	Region-End Page Number	Page number where the region ends, for use in region determination	R/W

V Bit (Valid Bit)

This bit enables or disables the settings for the corresponding region.
This bit is set to 0 when the region invalidation operation register (MPOPI) invalidates all access-controlled areas.

UAC[2:0] Bits (Access Control Bits in User Mode)

These bits specify the access control in user mode.

REPN[27:0] Bits (Region-End Page Number)

These bits specify the page number where the region ends.
Specify a value that is greater than or equal to the page number where the corresponding region starts. The page specified by the region-end page number is part of the target region for memory protection.

17.2.3 Memory-Protection Enable Register (MPEN)

Address(es): 0008 6500h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
b0	MPEN	Memory-Protection Enable	1: The memory protection is enabled. 0: The memory protection is disabled.	R/W
b31 to b1	—	Reserved	The read value is 0. The write value should always be 0	R/W

MPEN Bit (Memory-Protection Enable)

This bit enables or disables the memory protection.

After 1 has been written to this bit, address checking for memory protection by the CPU starts on the execution of a branch instruction (RTE or RTFI) that shifts operation to the user mode.

17.2.4 Background Access Control Register (MPBAC)

Address(es): 0008 6504h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	UBAC[2:0]			—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
b0	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b3 to b1	UBAC[2:0]	Background Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R/W
b31 to b4	—	Reserved	The read value is 0. The write value should always be 0.	R/W

UBAC[2:0] Bits (Background Access Control Bits in User Mode)

These bits specify the background access control in user mode.

17.2.5 Memory-Protection Error Status-Clearing Register (MPECLR)

Address(es): 0008 6508h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
b0	CLR	Error Status-Clearing	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: The DRW, DMPER and IMPER bits in MPESTS are set to 0.	R/W
b31 to b1	—	Reserved	The read value is 0. The write value should always be 0.	R/W

CLR Bit (Error Status-Clearing)

This bit clears the data read/write bit (DRW), the data memory-protection error generation bit (DMPER), and the instruction memory-protection error generation bit (IMPER) in the memory-protection error status register (MPESTS) to 0.

17.2.6 Memory-Protection Error Status Register (MPESTS)

Address(es): 0008 650Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	DRW	DMPER	IMPER
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
b0	IMPER	Instruction Memory-Protection Error Generation	0: No instruction memory-protection error was generated. 1: Instruction memory-protection error was generated.	R
b1	DMPER	Data Memory-Protection Error Generation	0: No data memory-protection error was generated. 1: Data memory-protection error was generated.	R
b2	DRW	Data Read/Write	0: Data were read. 1: Data were written.	R
b31 to b3	—	Reserved	The read value is 0. The write value should always be 0.	R/W

IMPER Bit (Instruction Memory-Protection Error Generation)

This bit indicates the state of memory-protection error generation by instruction execution.

Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

DMPER Bit (Data Memory-Protection Error Generation)

This bit indicates the state of memory-protection error generation by operand access.

Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

DRW Bit (Data Read/Write)

For a memory-protection error produced by operand access, this bit indicates the read/write attribute of the access operation. This bit is only valid when the DMPER bit is 1.

Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

17.2.7 Data Memory-Protection Error Address Register (MPDEA)

Address(es): 0008 6514h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DEA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Function	R/W
b31 to b0	DEA[31:0]	Data Memory-Protection Error Address	Data memory-protection error address	R

DEA[31:0] Bits (Data Memory-Protection Error Address)

These bits retain the address for which operand access generated a memory-protection error.

17.2.8 Region Search Address Register (MPSA)

Address(es): 0008 6520h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Function	R/W
b31 to b0	SA[31:0]	Region Search Address	Address for region searching	R/W

SA[31:0] Bits (Region Search Address)

These bits specify the address for use in comparison with region-start addresses in the region-n start page number registers (RSPAGEn) and region-end addresses in the region-n end page number registers (REPAGEn).

17.2.9 Region Search Operation Register (MPOPS)

Address(es): 0008 6524h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
b0	S	Region Search Operation	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: A region-search operation proceeds.	R/W
b15 to b1	—	Reserved	The read value is 0. The write value should always be 0.	R/W

S Bit (Region Search Operation)

Setting this bit to 1 makes the memory-protection unit perform a region-search operation. The address specified in the region search address register (MPSA) is compared with the address information for individual regions to search for a hitting region.

The result of searching is stored in the data-hit region bits (HITD[7:0]) of the data-hit region register (MHITD). Moreover, the logical OR of the respective access control bits for hitting regions is stored in the data-hit region access control bits (UHACD[2:0]) in user mode.

17.2.10 Region Invalidation Operation Register (MPOPI)

Address(es): 0008 6526h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
b0	INV	Region Invalidate Start	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: All access-controlled areas are invalidated.	R/W
b15 to b1	—	Reserved	The read value is 0. The write value should always be 0.	R/W

INV Bit (Region Invalidate Start)

Setting this bit to 1 clears the valid (V) bits in all of the region-n end page number registers (REPAGEn) to 0. After a V bit is set to 0, all settings other than background access-control settings are invalid.

17.2.11 Instruction-Hit Region Register (MHITI)

Address(es): 0008 6528h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	HITI[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	UHACI[2:0]			—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
b0	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b3 to b1	UHACI[2:0]	Instruction-Hit Region Access Control Bits in User Mode	b3 0: Reading prohibited 1: Read permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution is permitted.	R
b15 to b4	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b23 to b16	HITI[7:0]	Instruction-Hit Region	When the instruction memory-protection error generation bit (MPESTS.IMPER) = 1, [b23:b16] = 0000 0000b indicates that attempted access to the background region led to an instruction memory-protection error. Other than above b23 0: Instruction memory-protection error was not generated in region 7. 1: Instruction memory-protection error was generated in region 7. b22 0: Instruction memory-protection error was not generated in region 6. 1: Instruction memory-protection error was generated in region 6. b21 0: Instruction memory-protection error was not generated in region 5. 1: Instruction memory-protection error was generated in region 5. b20 0: Instruction memory-protection error was not generated in region 4. 1: Instruction memory-protection error was generated in region 4. b19 0: Instruction memory-protection error was not generated in region 3. 1: Instruction memory-protection error was generated in region 3. b18 0: Instruction memory-protection error was not generated in region 2. 1: Instruction memory-protection error was generated in region 2. b17 0: Instruction memory-protection error was not generated in region 1. 1: Instruction memory-protection error was generated in region 1. b16 0: Instruction memory-protection error was not generated in region 0. 1: Instruction memory-protection error was generated in region 0.	R
b31 to b24	—	Reserved	The read value is 0. The write value should always be 0.	R/W

UHACI[2:0] Bits (Instruction-Hit Region Access Control Bits in User Mode)

These bits hold the user-mode access control bits (REPAGEn.UAC[2:0]) for the region where the instruction memory-protection error was generated.

If the error was generated in an overlap between regions, the value stored here is the logical OR of the user-mode access control bits for the corresponding regions (including the background region).

HITI[7:0] Bits (Instruction-Hit Region)

These bits indicate the region where an instruction memory-protection error was generated. These bits are set to 0000 0000b in response to the generation of an instruction memory-protection error in the background region.

17.2.12 Data-Hit Region Register (MHITD)

Address(es): 0008 652Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	HITD[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	UHACD[2:0]			—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
b0	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b3 to b1	UHACD[2:0]	Data-Hit Region Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R
b15 to b4	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b23 to b16	HITD[7:0]	Data-Hit Region	When the data memory-protection error generation bit (MPESTS.DMPER) = 1, [b23:b16] = 0000 0000b indicates that attempted access to the background region led to a data memory-protection error. Other than above b23 0: Neither a data memory-protection error nor a search hit was generated in region 7. 1: A data memory-protection error or search hit was generated in region 7. b22 0: Neither a data memory-protection error nor a search hit was generated in region 6. 1: A data memory-protection error or search hit was generated in region 6. b21 0: Neither a data memory-protection error nor a search hit was generated in region 5. 1: A data memory-protection error or search hit was generated in region 5. b20 0: Neither a data memory-protection error nor a search hit was generated in region 4. 1: A data memory-protection error or search hit was generated in region 4. b19 0: Neither a data memory-protection error nor a search hit was generated in region 3. 1: A data memory-protection error or search hit was generated in region 3. b18 0: Neither a data memory-protection error nor a search hit was generated in region 2. 1: A data memory-protection error or search hit was generated in region 2. b17 0: Neither a data memory-protection error nor a search hit was generated in region 1. 1: A data memory-protection error or search hit was generated in region 1. b16 0: Neither a data memory-protection error nor a search hit was generated in region 0. 1: A data memory-protection error or search hit was generated in region 0.	R
b31 to b24	—	Reserved	The read value is 0. The write value should always be 0.	R/W

UHACD[2:0] Bits (Data-Hit Region Access Control Bits in User Mode)

These bits hold the user-mode access control bits (REPAGEn.UAC[2:0]) that have been set for the region where a data memory-protection error was generated or the region that produced a hit in region searching.

When an error is generated in an overlap between regions or a hit was generated in region searching, the value stored here is the logical OR of the user-mode access control bits for the corresponding regions (including the background region).

HITD[7:0] Bits (Data-Hit Region)

These bits indicate the region where a data memory-protection error was generated or the region that produced a hit in a region search. These bits are set to 0000 0000b for a data memory-protection error generated in the background region.

Note: When access to a register of memory protection unit in user mode generates a data memory-protection error, the value in this register is set to 0000 0000h.

17.3 Functions

17.3.1 Memory Protection

Memory protection means monitoring, in accord with the access-control information that has been set for the individual access-control regions and the background region, whether or not access by programs running in user mode violates the access-control settings. The memory-protection unit notifies the CPU of access-control violations (or memory-protection errors) when they are detected, causing the CPU to start access-exception processing.
Memory protection is enabled by setting the memory-protection enable (MPEN) bit in the memory-protection enable (MPEN) register to 1.
An instruction memory-protection error is generated on detection of an instruction-execution violation and a data memory-protection error is generated on detection of an operand-access reading or writing violation. Operand access that leads to a data memory-protection error is not actually executed.

17.3.2 Region Search

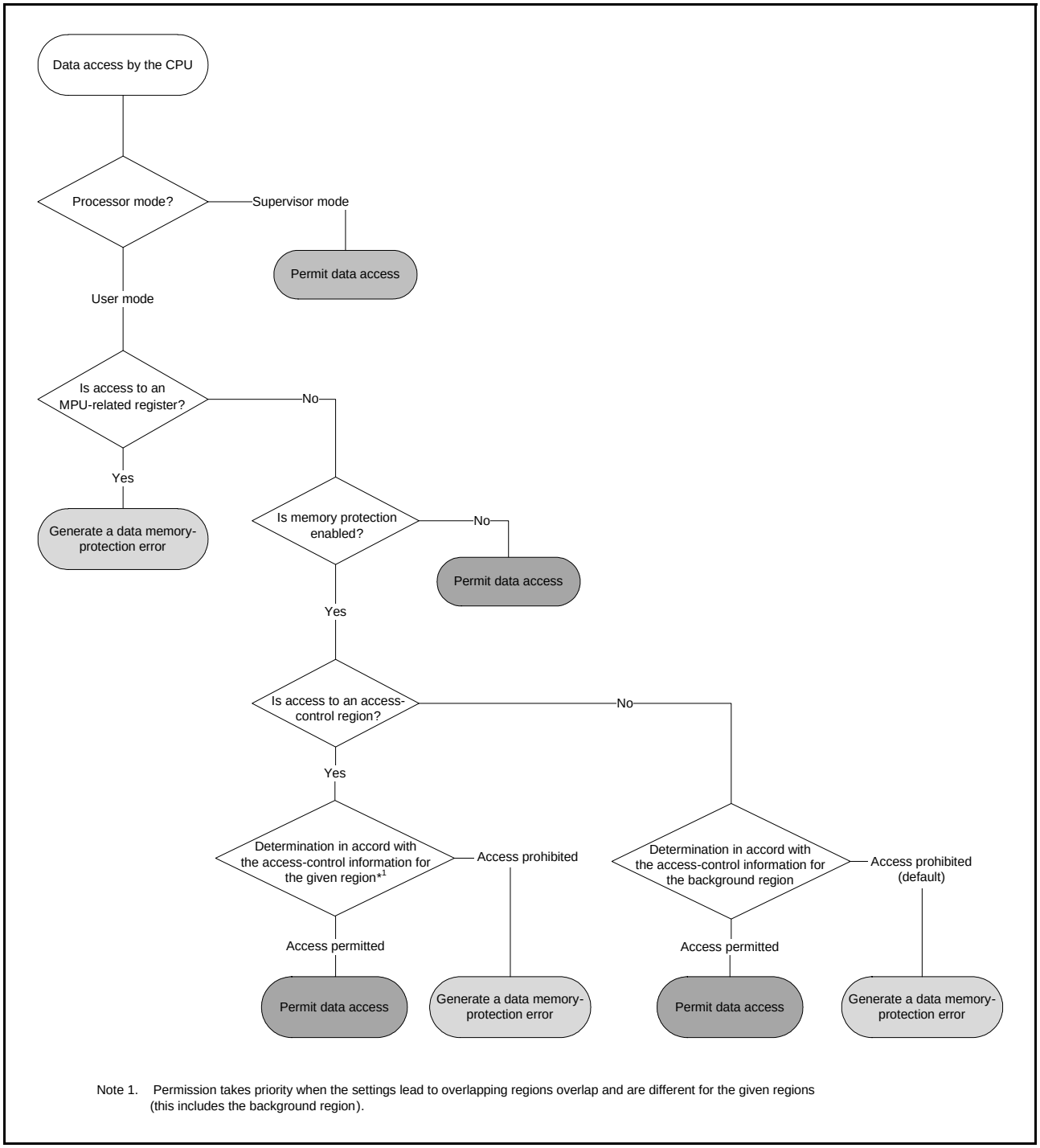
Region search means enquiry as to which of the eight specified access regions was “hit” and how the access-control information (permission to execute, to read, and to write) is set.
When the region search operation (S) bit in the region-search operation (MPOPS) register is set to 1, the address specified in the region search address (MPSA) register is compared with the addresses for the individual regions. After a region search is executed, the data-hit region register (MHITD) indicates the logical OR of the access-control information for the region which was “hit” and for the other regions.

17.3.3 Protection of Registers Related to the Memory-Protection Unit

Registers related to the memory-protection unit are not accessible through means of access other than operand access by the CPU (i.e. by instruction fetching or DMA). Attempted access to registers related to the memory-protection unit in user mode through operand access by the CPU leads to a data memory-protection error regardless of whether or not memory protection is in effect at the given location.

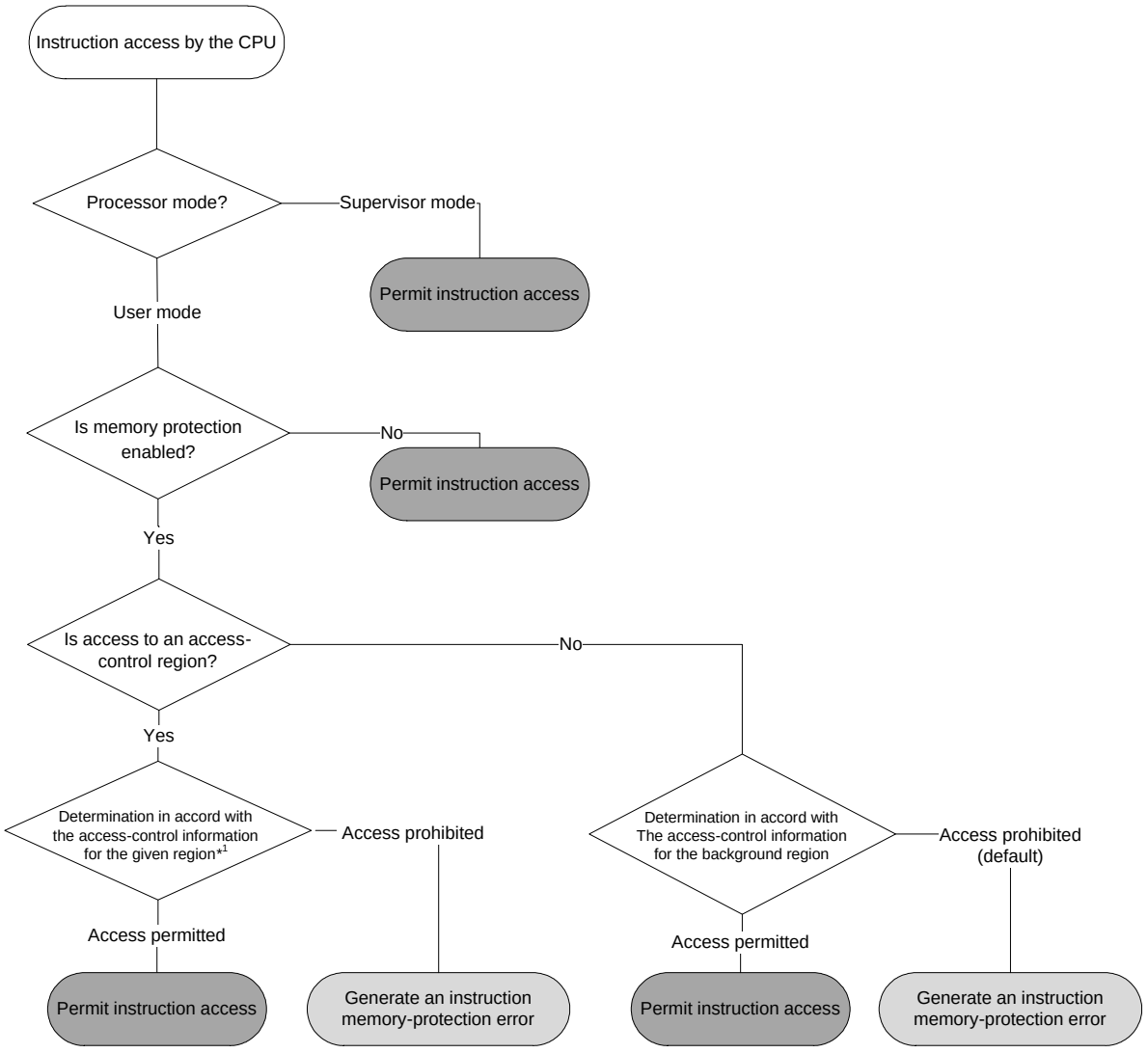
17.3.4 Flow for Determination of Access by the Memory-Protection Function

Figure 17.2 shows the flow of determination in the case of data access and Figure 17.3 shows the flow of determination in the case of instruction access.



Note 1. Permission takes priority when the settings lead to overlapping regions overlap and are different for the given regions (this includes the background region).

Figure 17.2 Flow of Determination for Data Access



Note 1. Permission takes priority when the settings lead to overlapping regions overlap and are not the same for the given regions (this includes the background region).

Figure 17.3 Flow of Determination for Instruction Access

17.4 Procedures for Using Memory Protection

17.4.1 Setting Access-Control Information

Access-control information for the various regions is set in supervisor mode.
Settings for up to eight access-control regions are made in the region-n start page number registers (RSPAGEn) and region-n end page number registers (REPAGEn), where n = 0 to 7.
Settings for the background access-control region are made in the background access-control register (MPBAC).

17.4.2 Enabling Memory Protection

Setting the memory-protection enable (MPEN) bit in the memory-protection enable (MPEN) register to 1 while operation is in supervisor mode enables memory protection.

17.4.3 Transition to User Mode

After updating the registers related to the memory-protection unit, read any of these registers and check that the settings have been made before the transition to user mode.
Either of the methods below can be used for the transition from supervisor mode to user mode.

- Set the processor mode setting (PM) bit in the copy of the processor status word (PSW) saved in the stack area to 1 (the setting for user mode) and then execute an RTE instruction.
- Set the PM bit in the backup processor status word (BPSW) to 1 and then execute an RTFI instruction.

Note: Using an MVTC or POPC instruction to write to the PSW.PM bit is invalid. Use an RTE or RTFI instruction to update the value of the PSW.PM bit.

The memory-protection unit starts checking instruction-execution access and operand access by the CPU on the transition to user mode.

17.4.4 Processing in Response to Memory-Protection Errors

The CPU starts access-exception processing on detection of a violation of protection set up by the access-control information (i.e. a memory-protection error). For details on CPU operations in access-exception processing, refer to section 14, Exception Handling.
To determine whether an instruction memory-protection error or data memory-protection error has been generated, check the values of the instruction memory-protection error generation (IMPER) and data memory-protection error generation (DMPER) bits in the memory-protection error status register (MPESTS) from within the exception-processing routine.
After confirming the type of error, clear the memory-protection error status register (MPESTS) by writing 1 to the status clearing (CLR) bit in the memory-protection error status clearing register (MPECLR).

(1) When a data memory-protection error is generated

Access-exception processing by the CPU saves the address of the instruction that led to the memory-protection error on the stack. Furthermore, the address of the operand for which access led to a memory-protection error is stored in the data memory-protection error address register (MPDEA) and the region information for the region where the memory-protection error was generated is stored in the data-hit region register (MHITD).

- Violations of access control in access to valid regions 0 to 7

The data-hit region bits (MHITD.HITD[7:0]) with the same region number as the region where the error occurred is set to 1. In user mode, the logical “or” of the region access-control information for the location where the error occurred is set in the data-hit region access-control bits (MHITD.UHACD[2:0]).

- Violations of access control for the background region, besides access to outside valid regions 0 to 7

The data-hit region bits (MHITD.HITD[7:0]) are set to 0000 0000b. In user mode, the access-control information for the background region is set in the data-hit region access-control bits (MHITD.UHACD[2:0]).

Referring to this information can pinpoint the sources of errors.

(2) When an instruction memory-protection error is generated

Access-exception processing by the CPU saves the address of the instruction that led to the memory-protection error on the stack. Furthermore, the region information for the region where the memory-protection error was generated is stored in the instruction-hit region register (MHITI).

- Violations of access control in access to valid regions 0 to 7

The instruction-hit region bit (MHITD.HITI[7:0]) with the same region number as the region where the error occurred is set to 1. In user mode, the logical “or” of the region access-control information for the location where the error occurred is set in the instruction-hit region access-control bits (MHITI.UHACI[2:0]).

- Violations of access control for the background region, besides access to outside valid regions 0 to 7

The instruction-hit region bits (MHITI.HITI[7:0]) are set to 0000 0000b. In user mode, the access-control information for the background region is set in the instruction-hit region access-control bits (MHITI.UHACI[2:0]).

Referring to this information can pinpoint the sources of errors.

18. DMA Controller (DMACAa)

This MCU incorporates an 8-channel direct memory access controller (DMAC).
The DMAC is a module to transfer data without the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address.

18.1 Overview

Table 18.1 lists the specifications of the DMAC, and Figure 18.1 shows a block diagram of the DMAC.

Table 18.1 Specifications of DMAC

Item		Description
Number of channels		8 (DMACm (m = 0 to 7))
Transfer space		512 Mbytes (0000 0000h to 0FFF FFFFh and F000 0000h to FFFF FFFFh excluding reserved areas)
Maximum transfer volume		64 Mbytes (Maximum number of transfers in block transfer mode: 1,024 data × 65,536 blocks)
DMA request source		• Request source selectable for each channel Software trigger Interrupt requests from peripheral modules or trigger input to external interrupt input pins*1
Channel priority		Channel 0 > Channel 1 > Channel 2 > Channel 3... > Channel 7 (Channel 0: Highest)
Transfer data	Single data	Bit length: 8, 16, 32 bits
	Block size	Number of data: 1 to 1,024
Transfer mode	Normal transfer mode	• One data transfer by one DMA transfer request • Free running mode (setting in which total number of data transfers is not specified) settable
	Repeat transfer mode	• One data transfer by one DMA transfer request • Program returns to the transfer start address on completion of the repeat size of data transfer specified for the transfer source or destination. • Maximum settable repeat size: 1,024
	Block transfer mode	• One block data transfer by one DMA transfer request • Maximum settable block size: 1,024 data
Selective functions	Extended repeat area function	• Function in which data can be transferred by repeating the address values in the specified range with the upper bit values in the transfer address register fixed • Area of 2 bytes to 128 Mbytes separately settable as extended repeat area for transfer source and destination
Interrupt request	Transfer end interrupt	Generated when the specified number of transfers is completed in normal transfer mode Generated when the specified repeat count of transfers is completed in repeat transfer mode Generated when the specified block count of transfers is completed in block transfer mode
	Transfer escape end interrupt	Generated when the repeat size of data transfer is completed or the extended repeat area overflows.
Event link function		An event link request is generated after each data transfer (for block transfer, after each block is transferred).
Power consumption reduction function		Module-stop state can be set.

Note 1. For details on DMA request sources, see Table 15.5, Interrupt Vector Table in section 15, Interrupt Controller (ICUA).

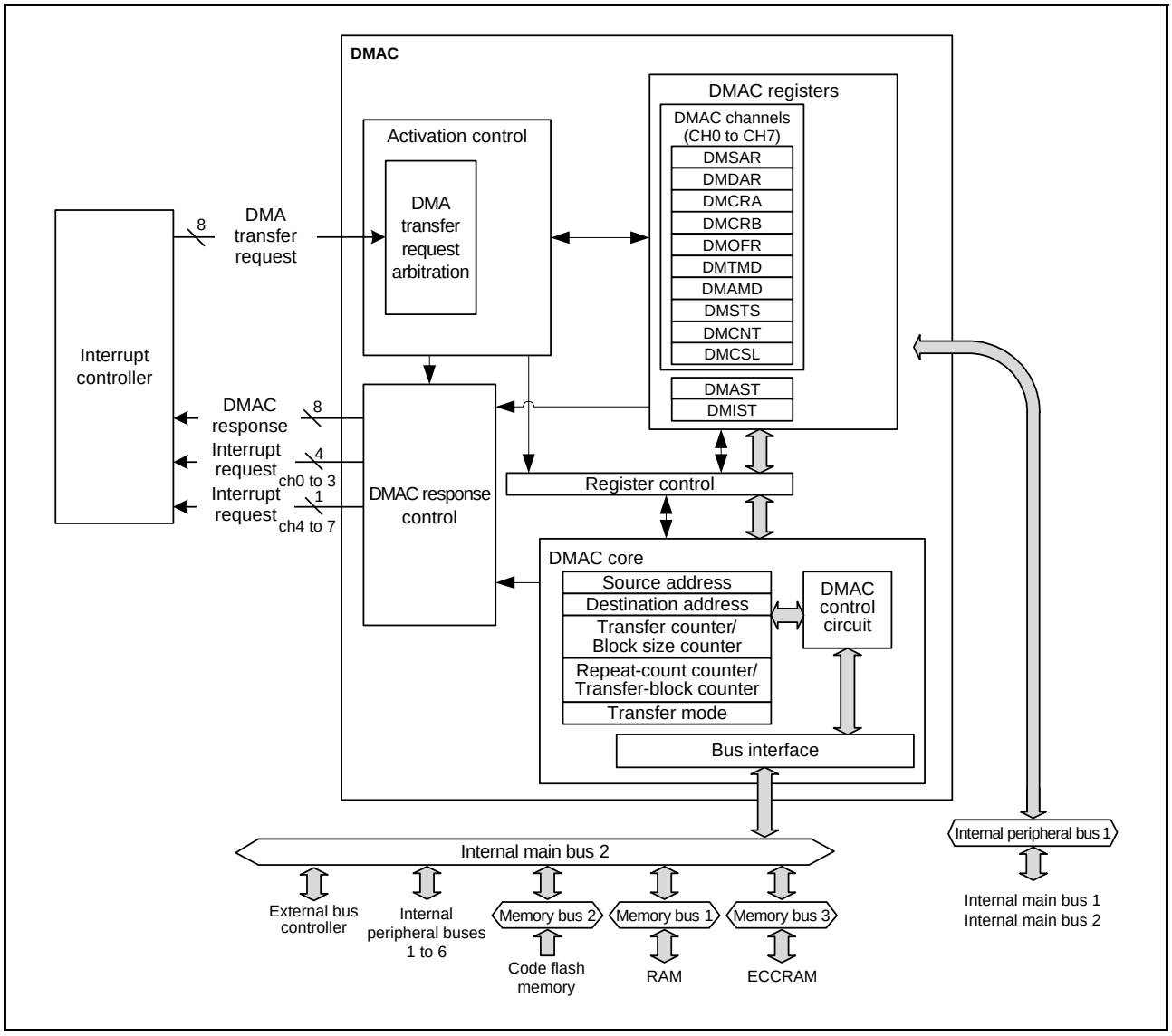
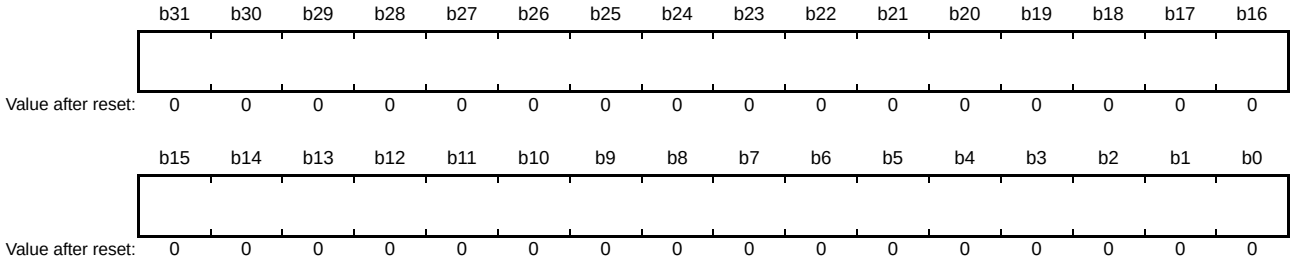


Figure 18.1 Block Diagram of DMAC

18.2 Register Descriptions

18.2.1 DMA Source Address Register (DMSAR)

Address(es): DMAC0.DMSAR 0008 2000h, DMAC1.DMSAR 0008 2040h, DMAC2.DMSAR 0008 2080h, DMAC3.DMSAR 0008 20C0h, DMAC4.DMSAR 0008 2100h, DMAC5.DMSAR 0008 2140h, DMAC6.DMSAR 0008 2180h, DMAC7.DMSAR 0008 21C0h



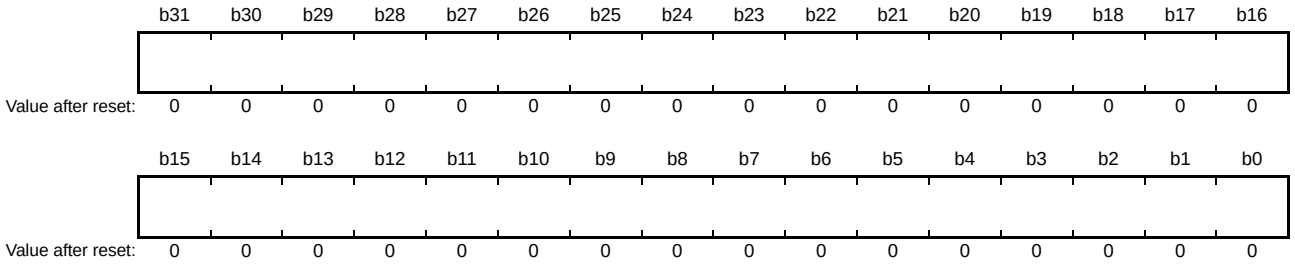
Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer source start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

Set DMSAR while DMAC stops (the DMST bit in DMAST = 0) or DMA transfer is disabled (the DTE bit in DMCNT = 0).

Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading DMSAR returns the extended value.

18.2.2 DMA Destination Address Register (DMDAR)

Address(es): DMAC0.DMDAR 0008 2004h, DMAC1.DMDAR 0008 2044h, DMAC2.DMDAR 0008 2084h, DMAC3.DMDAR 0008 20C4h, DMAC4.DMDAR 0008 2104h, DMAC5.DMDAR 0008 2144h, DMAC6.DMDAR 0008 2184h, DMAC7.DMDAR 0008 21C4h



Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer destination start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

Set DMDAR while DMAC stops (the DMST bit in DMAST = 0) or DMA transfer is disabled (the DTE bit in DMCNT = 0).

Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading DMDAR returns the extended value.

18.2.3 DMA Transfer Count Register (DMCRA)

Address(es): DMAC0.DMCRA 0008 2008h, DMAC1.DMCRA 0008 2048h, DMAC2.DMCRA 0008 2088h, DMAC3.DMCRA 0008 20C8h, DMAC4.DMCRA 0008 2108h, DMAC5.DMCRA 0008 2148h, DMAC6.DMCRA 0008 2188h, DMAC7.DMCRA 0008 21C8h

- Normal transfer mode

DMCRAH

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMCRAL

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- Repeat transfer mode, block transfer mode

DMCRAH

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMCRAL

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	0	0	0	0	0	0										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	Bit Name	Description	R/W
DMCRAL	Lower bits of transfer count	Specifies the number of transfer operations	R/W
DMCRAH	Upper bits of transfer count		R/W

Note: Set the same value for DMCRAH and DMCRAL in repeat transfer mode and block transfer mode.

(1) Normal Transfer Mode (MD[1:0] Bits in DMACm.DMTMD = 00b)

DMCRAL functions as a 16-bit transfer counter.

The number of transfer operations is one when the setting is 0001h, and 65,535 when it is FFFFh. The value is decremented by one each time data is transferred.

When the setting is 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode).

DMCRAH is not used in normal transfer mode. Write 0000h to DMCRAH.

(2) Repeat Transfer Mode (MD[1:0] Bits in DMACm.DMTMD = 01b)

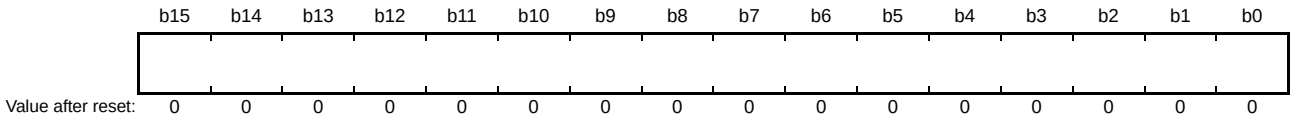
DMCRAH specifies the repeat size and DMCRAL functions as a 10-bit transfer counter.
The number of transfer operations is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In repeat transfer mode, a value in the range of 000h to 3FFh (1 to 1024) can be set for DMCRAH and DMCRAL.
Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.
The value in DMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in DMCRAH is loaded into DMCRAL.

(3) Block Transfer Mode (MD[1:0] Bits in DMACm.DMTMD = 10b)

DMCRAH specifies the block size and DMCRAL functions as a 10-bit block size counter.
The block size is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In block transfer mode, a value in the range of 000h to 3FFh can be set for DMCRAH and DMCRAL.
Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.
The value in DMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in DMCRAH is loaded into DMCRAL.

18.2.4 DMA Block Transfer Count Register (DMCRB)

Address(es): DMAC0.DMCRB 0008 200Ch, DMAC1.DMCRB 0008 204Ch, DMAC2.DMCRB 0008 208Ch, DMAC3.DMCRB 0008 20CCh, DMAC4.DMCRB 0008 210Ch, DMAC5.DMCRB 0008 214Ch, DMAC6.DMCRB 0008 218Ch, DMAC7.DMCRB 0008 21CCh



Bit	Description	Setting Range	R/W
b15 to b0	Specifies the block count or repeat count of transfers.	0001h to FFFFh (1 to 65,535) 0000h (65,536)	R/W

This register specifies the transfer block count in block transfer mode and the repeat count in repeat transfer mode.
In repeat transfer mode, the value is decremented by one when the final data of one repeat size is transferred.
In block transfer mode, the value is decremented by one when the final data of one block size is transferred.
In normal transfer mode, DMCRB is not used. The setting is invalid.

18.2.5 DMA Transfer Mode Register (DMTMD)

Address(es): DMAC0.DMTMD 0008 2010h, DMAC1.DMTMD 0008 2050h, DMAC2.DMTMD 0008 2090h, DMAC3.DMTMD 0008 20D0h, DMAC4.DMTMD 0008 2110h, DMAC5.DMTMD 0008 2150h, DMAC6.DMTMD 0008 2190h, DMAC7.DMTMD 0008 21D0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MD[1:0]		DTS[1:0]		—	—	SZ[1:0]		—	—	—	—	—	—	DCTG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DCTG[1:0]	Transfer Request Source Select	b1 b0 0 0: Software 0 1: Interrupts*1 from peripheral modules or external interrupt input pins 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	SZ[1:0]	Transfer Data Size Select	b9 b8 0 0: 8 bits 0 1: 16 bits 1 0: 32 bits 1 1: Setting prohibited	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13, b12	DTS[1:0]	Repeat Area Select	b13 b12 0 0: The destination is specified as the repeat area or block area. 0 1: The source is specified as the repeat area or block area. 1 0: The repeat area or block area is not specified. 1 1: Setting prohibited	R/W
b15, b14	MD[1:0]	Transfer Mode Select	b15 b14 0 0: Normal transfer 0 1: Repeat transfer 1 0: Block transfer 1 1: Setting prohibited	R/W

Note 1. DMA request source is selected using the DMRSRm registers of the ICU. For details on DMA request sources, refer to Table 15.5, Interrupt Vector Table in section 15, Interrupt Controller (ICUA).

DTS[1:0] Bits (Repeat Area Select)

DTS[1:0] select either the source or destination as the repeat area in repeat or block transfer mode. In normal transfer mode, setting these bits is invalid.

18.2.6 DMA Interrupt Setting Register (DMINT)

Address(es): DMAC0.DMINT 0008 2013h, DMAC1.DMINT 0008 2053h, DMAC2.DMINT 0008 2093h, DMAC3.DMINT 0008 20D3h, DMAC4.DMINT 0008 2113h, DMAC5.DMINT 0008 2153h, DMAC6.DMINT 0008 2193h, DMAC7.DMINT 0008 21D3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the destination address 1: Enables an interrupt request for an extended repeat area overflow on the destination address	R/W
b1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the source address 1: Enables an interrupt request for an extended repeat area overflow on the source address	R/W
b2	RPTIE	Repeat Size End Interrupt Enable	0: Disables the repeat size end interrupt request. 1: Enables the repeat size end interrupt request.	R/W
b3	ESIE	Transfer Escape End Interrupt Enable	0: Disables the transfer escape end interrupt request. 1: Enables the transfer escape end interrupt request.	R/W
b4	DTIE	Transfer End Interrupt Enable	0: Disables the transfer end interrupt request. 1: Enables the transfer end interrupt request.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DARIE Bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the destination address occurs while this bit is set to 1, the DTE bit in DMCNT is set to 0. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that an interrupt by an extended repeat area overflow on the destination address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DTE bit in DMACm.DMCNT of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the destination address, this bit is ignored.

SARIE Bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the source address occurs while this bit is set to 1, the DTE bit in DMCNT is set to 0. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that an interrupt by an extended repeat area overflow on the source address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DTE bit in DMACm.DMCNT of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the source address, this bit is ignored.

RPTIE Bit (Repeat Size End Interrupt Enable)

When this bit is set to 1 in repeat transfer mode, the DTE bit in DMCNT is set to 0 after completion of a 1-repeat size data transfer. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in DMTMD are 10b (= repeat area or block area is not specified).

When this bit is set to 1 in block transfer mode, the DTE bit in DMCNT is set to 0 after completion of a 1-block data transfer in the same way as repeat transfer mode. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in DMTMD are 10b (= repeat area or block area is not specified).

ESIE Bit (Transfer Escape End Interrupt Enable)

This bit enables or disables the transfer escape end interrupt requests (repeat size end interrupt request and extended repeat area overflow interrupt request) that are generated during DMA transfer.

The transfer escape end interrupt is generated when the ESIF flag in DMSTS is set to 1 with this bit set to 1. The transfer escape end interrupt is cleared by setting this bit or the ESIF flag in DMSTS to 0.

DTIE Bit (Transfer End Interrupt Enable)

This bit enables or disables the transfer end interrupt request to be generated on completion of a specified number of data transfers.

The transfer end interrupt is generated when the DTIF bit in DMSTS is set to 1 with this bit set to 1. The transfer end interrupt is cleared by setting this bit or the DTIF bit in DMSTS to 0.

18.2.7 DMA Address Mode Register (DMAMD)

Address(es): DMAC0.DMAMD 0008 2014h, DMAC1.DMAMD 0008 2054h, DMAC2.DMAMD 0008 2094h, DMAC3.DMAMD 0008 20D4h, DMAC4.DMAMD 0008 2114h, DMAC5.DMAMD 0008 2154h, DMAC6.DMAMD 0008 2194h, DMAC7.DMAMD 0008 21D4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SM[1:0]		—	SARA[4:0]					DM[1:0]		—	DARA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DARA[4:0]	Destination Address Extended Repeat Area	Specifies the extended repeat area on the destination address. For details on the settings, see Table 18.2.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	DM[1:0]	Destination Address Update Mode	b7 b6 0 0: Destination address is fixed. 0 1: Offset addition*1 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
b12 to b8	SARA[4:0]	Source Address Extended Repeat Area	Specifies the extended repeat area on the source address. For details on the settings, see Table 18.2.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15, b14	SM[1:0]	Source Address Update Mode	b15 b14 0 0: Source address is fixed. 0 1: Offset addition*1 1 0: Source address is incremented. 1 1: Source address is decremented.	R/W

Note 1. Offset addition can be specified only for DMAC0.

DARA[4:0] Bits (Destination Address Extended Repeat Area)

These bits specify the extended repeat area on the destination address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer or block transfer is selected, and when DMACm.DMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat area or block area), write 00000b in the DARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DARIE bit in DMINT set to 1. Table 18.2 lists the settings and the corresponding extended repeat areas.

DM[1:0] Bits (Destination Address Update Mode)

These bits select the mode of updating the destination address.

When increment is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the destination address is incremented by 1, 2, and 4, respectively.

When decrement is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the destination address is decremented by 1, 2, and 4, respectively.

When offset addition is selected, the offset specified by the DMAC0.DMOFR register is added to the address.

Offset addition can be specified only for DMAC0.

SARA[4:0] Bits (Source Address Extended Repeat Area)

These bits specify the extended repeat area on the source address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer source, do not specify the extended repeat area on the source address. When repeat transfer or block transfer is selected, and when DMACm.DMTMD.DTS[1:0] = 01b (the transfer source is specified as the repeat area or block area), write 00000b in the SARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the SARIE bit in DMINT set to 1. Table 18.2 lists the settings and the corresponding extended repeat areas.

SM[1:0] (Source Address Update Mode)

These bits select the mode of updating the source address.

When increment is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the source address is incremented by 1, 2, and 4, respectively.

When decrement is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the source address is decremented by 1, 2, and 4, respectively.

When offset addition is selected, the offset specified by the DMAC0.DMOFR register is added to the address. Offset addition can be specified only for DMAC0.

Table 18.2 SARA[4:0] or DARA[4:0] Settings and Corresponding Repeat Areas

SARA[4:0] or DARA[4:0]	Extended Repeat Area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 Kbyte specified as extended repeat area by the lower 10 bits of the address
01011b	2 Kbytes specified as extended repeat area by the lower 11 bits of the address
01100b	4 Kbytes specified as extended repeat area by the lower 12 bits of the address
01101b	8 Kbytes specified as extended repeat area by the lower 13 bits of the address
01110b	16 Kbytes specified as extended repeat area by the lower 14 bits of the address
01111b	32 Kbytes specified as extended repeat area by the lower 15 bits of the address
10000b	64 Kbytes specified as extended repeat area by the lower 16 bits of the address
10001b	128 Kbytes specified as extended repeat area by the lower 17 bits of the address
10010b	256 Kbytes specified as extended repeat area by the lower 18 bits of the address
10011b	512 Kbytes specified as extended repeat area by the lower 19 bits of the address
10100b	1 Mbyte specified as extended repeat area by the lower 20 bits of the address
10101b	2 Mbytes specified as extended repeat area by the lower 21 bits of the address
10110b	4 Mbytes specified as extended repeat area by the lower 22 bits of the address
10111b	8 Mbytes specified as extended repeat area by the lower 23 bits of the address
11000b	16 Mbytes specified as extended repeat area by the lower 24 bits of the address
11001b	32 Mbytes specified as extended repeat area by the lower 25 bits of the address
11010b	64 Mbytes specified as extended repeat area by the lower 26 bits of the address
11011b	128 Mbytes specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	Setting prohibited.

18.2.8 DMA Offset Register (DMOFR)

Address(es): DMAC0.DMOFR 0008 2018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination.	0000 0000h to 00FF FFFFh (0 bytes to (16 M – 1) bytes) FF00 0000h to FFFF FFFFh (–16 Mbytes to –1 byte)	R/W

Write to this register while the DMAC operation is stopped or DMA transfer is disabled (not during data transfer).
Setting bits 31 to 25 is invalid; a value of bit 24 is extended to bits 31 to 25. Reading DMOFR returns the extended value.

18.2.9 DMA Transfer Enable Register (DMCNT)

Address(es): DMAC0.DMCNT 0008 201Ch, DMAC1.DMCNT 0008 205Ch, DMAC2.DMCNT 0008 209Ch, DMAC3.DMCNT 0008 20DCh, DMAC4.DMCNT 0008 211Ch, DMAC5.DMCNT 0008 215Ch, DMAC6.DMCNT 0008 219Ch, DMAC7.DMCNT 0008 21DCh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DTE	DMA Transfer Enable	0: Disables DMA transfer. 1: Enables DMA transfer.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DTE Bit (DMA Transfer Enable)

When the DMST bit in DMAST is set to 1 (DMAC module start) and this bit is set to 1 (DMA transfer is enabled), DMA transfer can be started for the corresponding channel.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When 0 is written to this bit.
- When the specified total volume of data transfer is completed.
- When DMA transfer is stopped by the repeat size end interrupt.
- When DMA transfer is stopped by the extended repeat area overflow interrupt.

18.2.10 DMA Software Start Register (DMREQ)

Address(es): DMAC0.DMREQ 0008 201Dh, DMAC1.DMREQ 0008 205Dh, DMAC2.DMREQ 0008 209Dh, DMAC3.DMREQ 0008 20DDh, DMAC4.DMREQ 0008 211Dh, DMAC5.DMREQ 0008 215Dh, DMAC6.DMREQ 0008 219Dh, DMAC7.DMREQ 0008 21DDh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CLRS	—	—	—	SWREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SWREQ	DMA Software Start	0: DMA transfer is not requested. 1: DMA transfer is requested.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CLRS	DMA Software Start Bit Auto Clear Select	0: SWREQ bit is cleared after DMA transfer is started by software. 1: SWREQ bit is not cleared after DMA transfer is started by software.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SWREQ Bit (DMA Software Start)

When 1 is written to this bit, a DMA transfer request is generated. After DMA transfer is started in response to the request, this bit is set to 0 if the CLRS bit is set to 0. This bit is not set to 0 while the CLRS bit is set to 1. In this case, a DMA transfer request can be issued again after completion of a transfer.

Note that, however, setting this bit is valid and DMA transfer by software is enabled only when the DCTG[1:0] bits in DMTMD are set to 00b (DMA request source is software).

Setting this bit is invalid when the DCTG[1:0] bits in DMTMD are set to a value other than 00b.

To start DMA transfer by software with the CLRS bit being 0, ensure that the SWREQ bit is 0, and then write 1 to the SWREQ bit.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When a DMA transfer request by software is accepted and DMA transfer is started while the CLRS bit is set to 0 (the SWREQ bit is cleared after DMA transfer is started by software).
- When 0 is written to this bit.

CLRS Bit (DMA Software Start Bit Auto Clear Select)

This bit specifies whether to clear the SWREQ bit to 0 after DMA transfer is started in response to the DMA transfer request generated by setting the SWREQ bit to 1. With this bit set to 0, the SWREQ bit is set to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not set to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

18.2.11 DMA Status Register (DMSTS)

Address(es): DMAC0.DMSTS 0008 201Eh, DMAC1.DMSTS 0008 205Eh, DMAC2.DMSTS 0008 209Eh, DMAC3.DMSTS 0008 20DEh, DMAC4.DMSTS 0008 211Eh, DMAC5.DMSTS 0008 215Eh, DMAC6.DMSTS 0008 219Eh, DMAC7.DMSTS 0008 21DEh

	b7	b6	b5	b4	b3	b2	b1	b0
	ACT	—	—	DTIF	—	—	—	ESIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESIF	Transfer Escape End Interrupt Flag	0: A transfer escape end interrupt has not been generated. 1: A transfer escape end interrupt has been generated.	R/W*1
b3 to b1	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b4	DTIF	Transfer End Interrupt Flag	0: A transfer end interrupt has not been generated. 1: A transfer end interrupt has been generated.	R/W*1
b6, b5	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	ACT	DMA Active Flag	0: DMAC operation is suspended. 1: DMAC is operating.	R

Note 1. Only 0 can be written to clear the flag.

ESIF Flag (Transfer Escape End Interrupt Flag)

This flag indicates that the transfer escape end interrupt has been generated.

[Setting conditions]

- When 1-repeat size data transfer is completed in repeat transfer mode with the RPTIE bit in DMINT set to 1.
- When 1-block data transfer is completed in block transfer mode with the RPTIE bit in DMINT set to 1.
- When an extended repeat area overflow on the source address occurs while the SARIE bit in DMINT is set to 1 and the SARA[4:0] bits in DMAMD are set to a value other than 00000b (extended repeat area is specified on the transfer source address)
- When an extended repeat area overflow on the destination address occurs while the DARIE bit in DMINT is set to 1 and the DARA[4:0] bits in DMAMD are set to a value other than 00000b (extended repeat area is specified on the transfer destination address)

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the DTE bit in DMCNT.

DTIF Flag (Transfer End Interrupt Flag)

This flag indicates that the transfer end interrupt has been generated.
[Setting conditions]

- When the specified number of unit-transfers are completed in normal transfer mode (the value of DMCRAL becoming 0 on completion of transfer)
- When the specified number of repeat transfer operations are completed in repeat transfer mode (the value of DMCRB becoming 0 on completion of transfer))
- When the specified number of blocks have been transferred in block transfer mode (the value of DMCRB becoming 0 on completion of transfer)

[Clearing conditions]

- When 0 is written to this bit
- When 1 is written to the DTE bit in DMCNT

ACT Flag (DMA Active Flag)

This flag indicates whether the DMAC is in the idle or active state.
[Setting condition]

- When the DMAC starts data transfer operation

[Clearing condition]

- When data transfer in response to one transfer request is completed

18.2.12 DMA Request Source Flag Control Register (DMCSL)

Address(es): DMAC0.DMCSL 0008 201Fh, DMAC1.DMCSL 0008 205Fh, DMAC2.DMCSL 0008 209Fh, DMAC3.DMCSL 0008 20DFh, DMAC4.DMCSL 0008 211Fh, DMAC5.DMCSL 0008 215Fh, DMAC6.DMCSL 0008 219Fh, DMAC7.DMCSL 0008 21DFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DISEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DISEL	Interrupt Select	0: At the beginning of transfer, clear the interrupt status flag of the request source to 0. 1: At the end of transfer, the interrupt status flag of the request source issues an interrupt to the CPU.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DISEL Bit (Interrupt Select)

This bit selects whether the interrupt status flag of the request source is set to 0 or issues an interrupt request to the CPU, at the beginning of DMA transfer.
When DMTMD.DCTG[1:0] = 00b (trigger by software), the setting of the DISEL bit does not affect the operation.

18.2.13 DMAC Module Start Register (DMAST)

Address(es): 0008 2200h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DMST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DMST	DMAC Module Start	0: DMAC module stop 1: DMAC module start	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DMST Bit (DMAC Module Start)

When this bit is set to 1, DMAC is ready to accept transfer requests for all channels.

When 1 is written to the DMACm.DMCNT.DTE bit (DMA transfer is enabled) of multiple channels and then this bit is set to 1 (DMAC module start), the corresponding multiple channels can be placed in the transfer request acceptable state at the same time.

When the DMST bit is set to 0 during DMA transfer, DMA transfer is suspended after completion of the current data transfer corresponding to a single transfer request. DMA transfer can be resumed by setting the DMST bit to 1 again.

[Setting condition]

- When 1 is written to this bit

[Clearing condition]

- When 0 is written to this bit

18.2.14 DMAC74 Interrupt Status Monitor Register (DMIST)

Address(es): 0008 2204h

	b7	b6	b5	b4	b3	b2	b1	b0
	DMIS7	DMIS6	DMIS5	DMIS4	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b4	DMIS4	DMAC4 Interrupt Status Flag	0: DMAC4 interrupt is not requested. 1: DMAC4 interrupt is requested.	R
b5	DMIS5	DMAC5 Interrupt Status Flag	0: DMAC5 interrupt is not requested. 1: DMAC5 interrupt is requested.	R
b6	DMIS6	DMAC6 Interrupt Status Flag	0: DMAC6 interrupt is not requested. 1: DMAC6 interrupt is requested.	R
b7	DMIS7	DMAC7 Interrupt Status Flag	0: DMAC7 interrupt is not requested. 1: DMAC7 interrupt is requested.	R

DMISm Flag (DMACm Interrupt Status Flag) (m = 4 to 7)

This bit monitors the DMACm interrupt request. Writing to this bit will be ignored.

While the DMACm.DMINT.DTIE bit is 1 and the DMACm.DMSTS.DTIF bit is 1, or the DMACm.DMINT.ESIE bit is 1 and the DMACm.DMSTS.ESIF bit is 1, the DMIST.DMISm bit is set to 1.

18.3 Operation

18.3.1 Transfer Mode

(1) Normal Transfer Mode

In normal transfer mode, one data is transferred by one transfer request. A maximum of 65535 can be set as the number of transfer operations using the DMCRAL of DMACm. When these bits are set to 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode). Setting DMCRB of DMACm is invalid in normal transfer mode. Except in free running mode, a transfer end interrupt request can be generated after completion of the specified number of transfer operations.
Table 18.3 summarizes the register update operation in normal transfer mode, and Figure 18.2 shows the operation in normal transfer mode.

Table 18.3 Register Update Operation in Normal Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request
DMACm.DMSAR	Transfer source address	Increment/decrement/fixed/offset addition*1
DMACm.DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition*1
DMACm.DMCRAL	Transfer counter	Decremented by one/not updated (in free running mode)
DMACm.DMCRAH	—	Not updated (Not used in normal transfer mode)
DMACm.DMCRB	—	Not updated (Not used in normal transfer mode)

Note 1. Offset addition can be specified only for DMAC0.

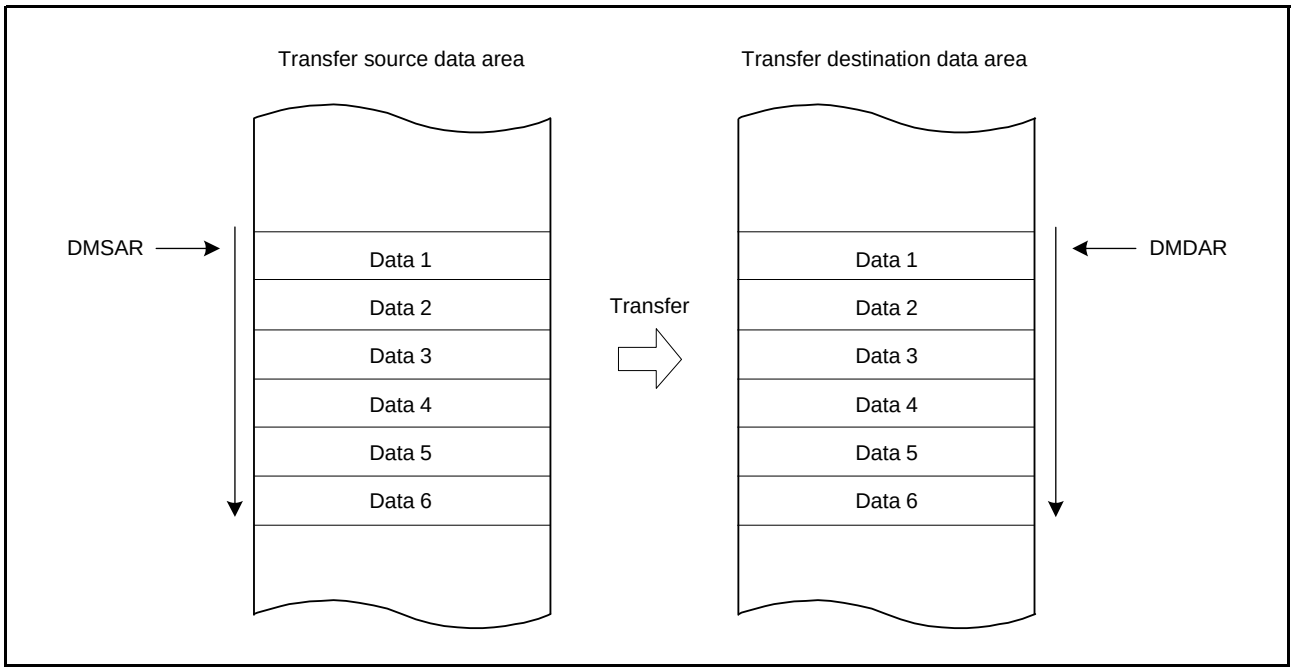


Figure 18.2 Operation in Normal Transfer Mode

(2) Repeat Transfer Mode

In repeat transfer mode, one data is transferred by one transfer request.
A maximum of 1K data can be set as a total repeat transfer size using DMCRA of the DMACm.
A maximum of 64K can be set as the number of repeat transfer operations using DMCRB of the DMACm; therefore, a maximum of 64M data (1K data × 64K counts of repeat transfer operations) can be set as a total data transfer size.
Either the transfer source or transfer destination can be specified as a repeat area. When transfer of the repeat size data is completed, the address of the specified repeat area (DMSAR or DMDAR of the DMACm) returns to the transfer start address. When data of the specified repeat size has all been transferred in repeat transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm in the repeat size end interrupt handling.
A transfer end interrupt request can be generated after completion of the specified number of repeat transfer operations.
Table 18.4 summarizes the register update operation in repeat transfer mode, and Figure 18.3 shows the operation in repeat transfer mode.

Table 18.4 Register Update Operation in Repeat Transfer Mode

		Update Operation after Completion of a Transfer by One Transfer Request	
Register	**Function**	**When DMACm.DMCRAL is not 1**	**When DMACm.DMCRAL is 1 (Transfer of the Last Data in Repeat Size)**
DMACm.DMSAR	Transfer source address	Increment/decrement/fixed/offset addition*1	• DMACm.DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 01b Initial value of DMACm.DMSAR • DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
DMACm.DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition*1	• DMACm.DMTMD.DTS[1:0] = 00b Initial value of DMACm.DMDAR • DMACm.DMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
DMACm.DMCRAH	Repeat size	Not updated	Not updated
DMACm.DMCRAL	Transfer counter	Decremented by one	DMACm.DMCRAH
DMACm.DMCRB	Repeat-count counter	Not updated	Decremented by one

Note 1. Offset addition can be specified only for DMAC0.

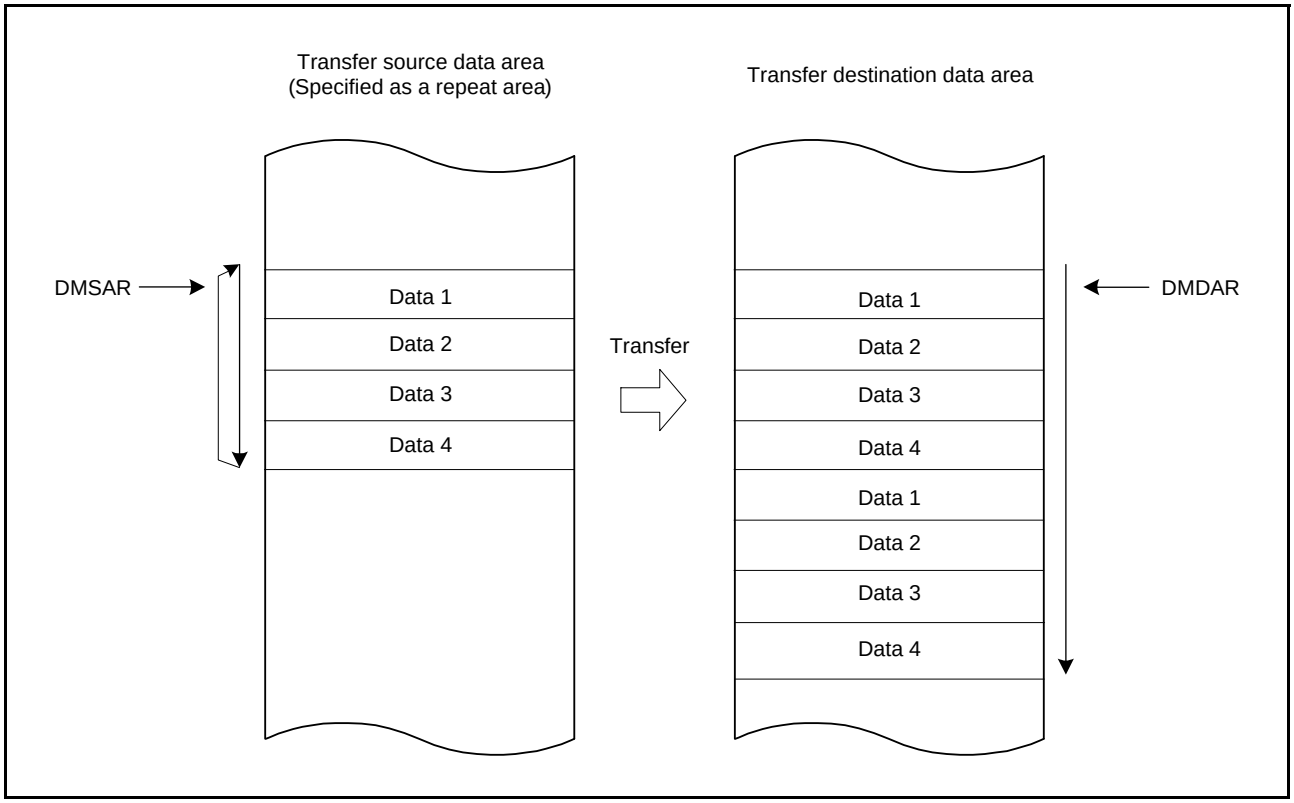


Figure 18.3 Operation in Repeat Transfer Mode

(3) Block Transfer Mode

In block transfer mode, a single block data is transferred by one transfer request.
A maximum of 1K data can be set as a total block transfer size using DMCRA of the DMACm.
A maximum of 64K can be set as the number of block transfer operations using DMCRB of the DMACm; therefore, a maximum of 64M data (1K data × 64K counts of block transfer operations) can be set as a total data transfer size.
Either the transfer source or transfer destination can be specified as a block area. When transfer of a single block data is completed, the address of the specified block area (DMSAR or DMDAR of the DMACm) returns to the transfer start address. When a single block data has all been transferred in block transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm in the repeat size end interrupt handling.
Transfer end interrupt request can be generated after completion of the specified number of block transfer operations.
Table 18.5 summarizes the register update operation in block transfer mode, and Figure 18.4 shows the operation in block transfer mode.

Table 18.5 Register Update Operation in Block Transfer Mode

Register	Function	Update Operation after Completion of Single-Block Transfer by One Transfer Request
DMACm.DMSAR	Transfer source address	• DMACm.DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 01b Initial value of DMACm.DMSAR • DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
DMACm.DMDAR	Transfer destination address	• DMACm.DMTMD.DTS[1:0] = 00 b Initial value of DMACm.DMDAR • DMACm.DMTMD.DTS[1:0] = 01 b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 10 b Increment/decrement/fixed/offset addition*1
DMACm.DMCRAH	Block size	Not updated
DMACm.DMCRAL	Block size counter	DMACm.DMCRAH
DMACm.DMCRB	Transfer-block counter	Decremented by one

Note 1. Offset addition can be specified only for DMAC0.

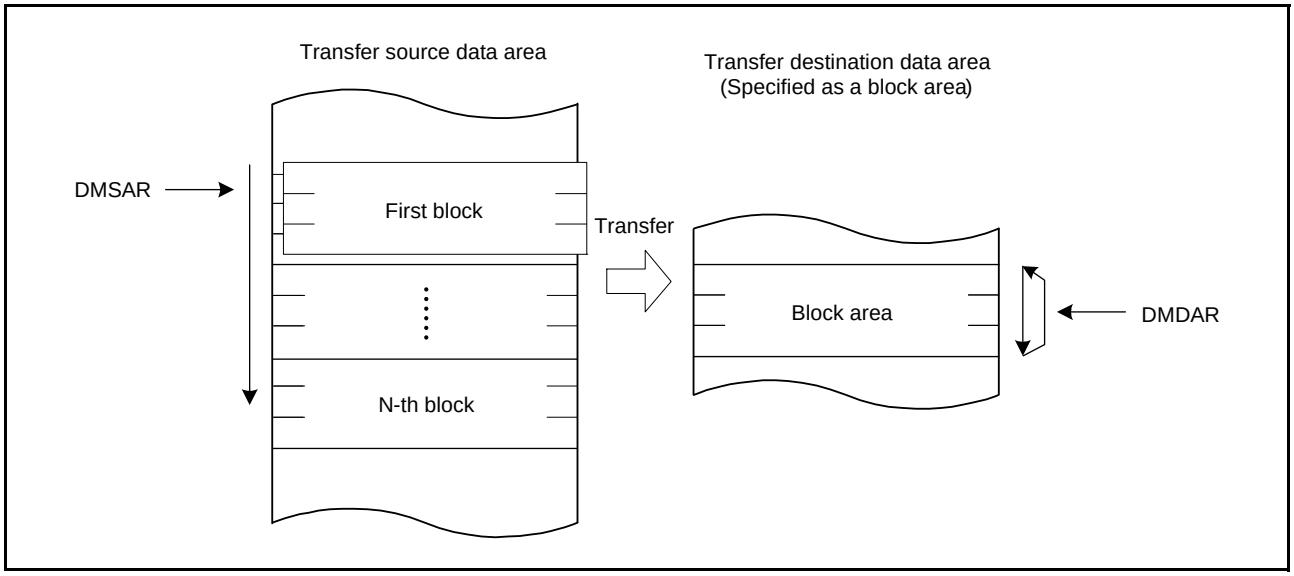


Figure 18.4 Operation in Block Transfer Mode

18.3.2 Extended Repeat Area Function

The DMAC supports a function to specify the extended repeat areas on the transfer source and destination addresses. With the extended repeat areas set, the address registers repeatedly indicate the addresses of the specified extended repeat areas.

The extended repeat areas can be specified separately to the transfer source address register (DMSAR) and transfer destination address register (DMDAR) of DMACm.

The extended repeat area on the source address is specified by the SARA[4:0] bits in DMAMD of DMACm. The extended repeat area on the destination address is specified by the DARA[4:0] bits in DMAMD of DMACm. The size can be specified separately for the source and destination sides.

However, the area (of transfer source or transfer destination) which is specified as the repeat area or block area should not be specified as the extended repeat area.

When the address register value reaches the end address of the extended repeat area and the extended repeat area overflows, DMA transfer is stopped and an interrupt by an extended repeat area overflow can be requested. When an overflow occurs in the extended repeat area on the transfer source while the SARIE bit in DMINT of DMACm is set to 1, the ESIF flag in DMSTS of DMACm is set to 1 and the DTE bit in DMCNT of DMACm is set to 0 to stop DMA transfer. At this time, if the ESIE bit in DMINT of DMACm is set to 1, an interrupt by an extended repeat area overflow is requested. When the DARIE bit in DMINT of DMACm is set to 1, the destination address register becomes a target to apply the function. DMA transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm in the interrupt handling.

Figure 18.5 shows an example of the extended repeat area operation.

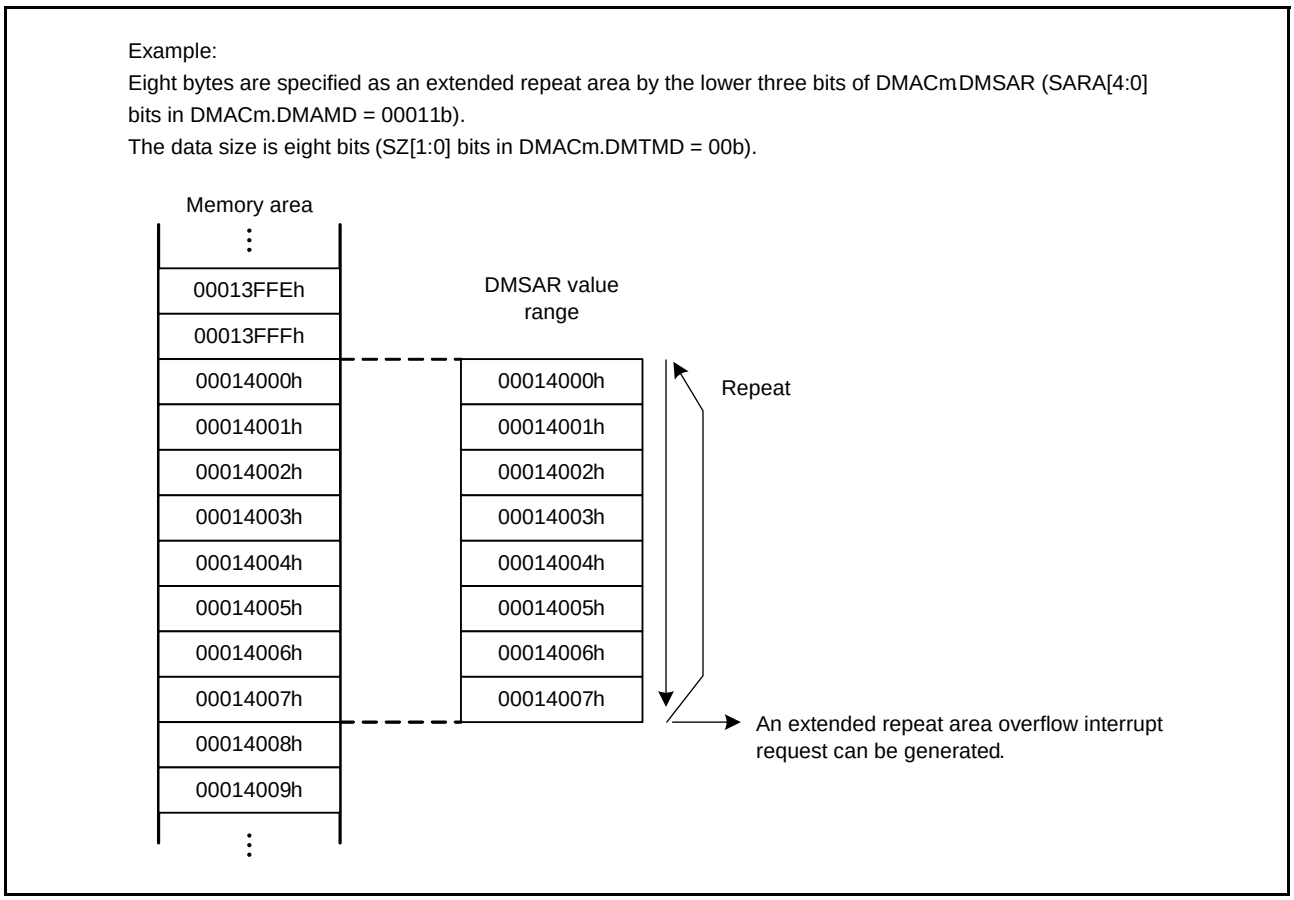


Figure 18.5 Example of Extended Repeat Area Operation

When an interrupt by an extended repeat area overflow is used in block transfer mode, the following should be taken into consideration.
When a transfer is stopped by an interrupt by an extended repeat area overflow, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block, the interrupt by the overflow is suspended until transfer of the block is completed, and the transfer overruns.

Figure 18.6 shows an example when the extended repeat area function is used in block transfer mode.

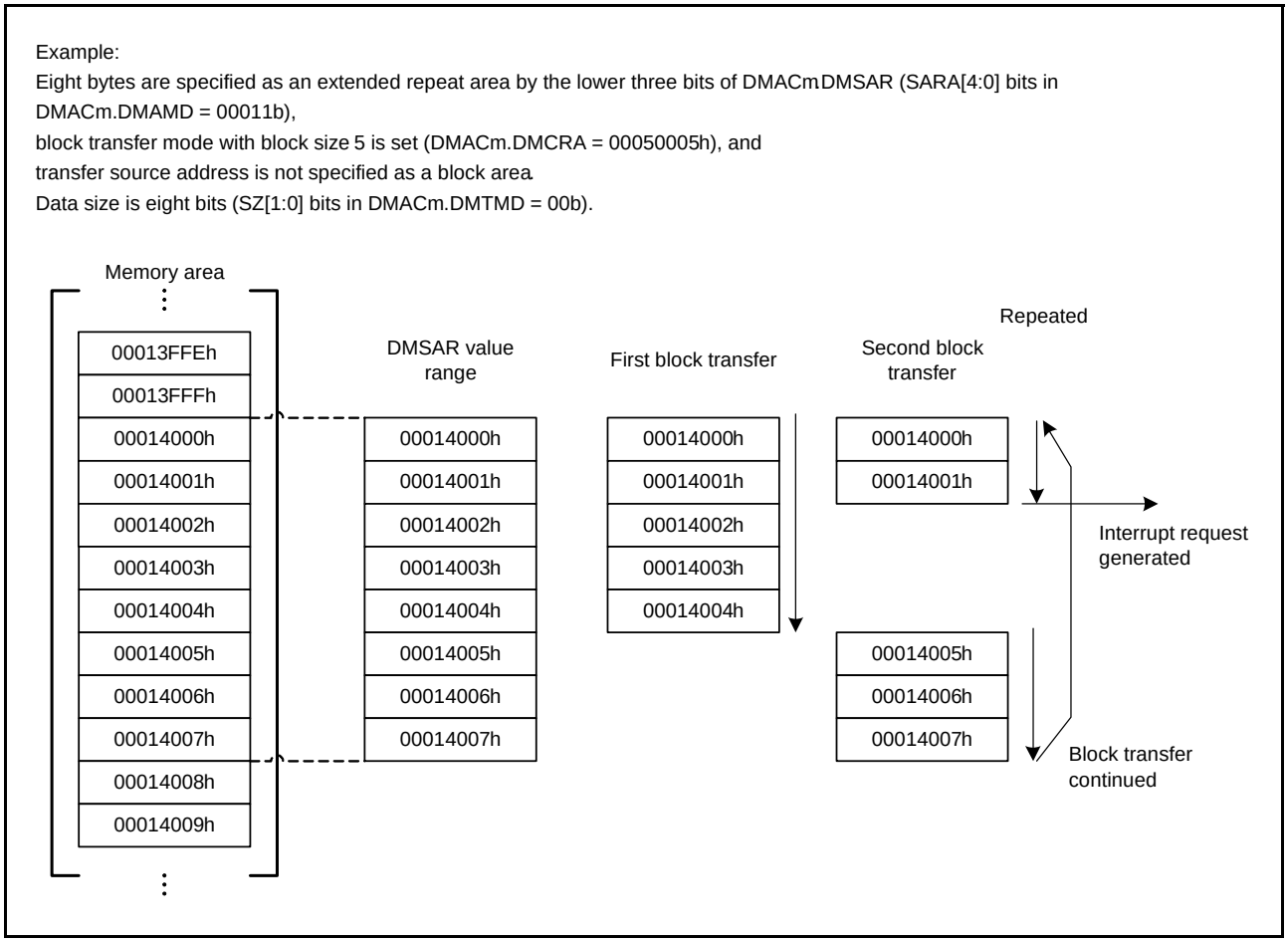


Figure 18.6 Example of Extended Repeat Area Function in Block Transfer Mode

18.3.3 Address Update Function using Offset

The source and destination addresses can be updated by fixing, increment, decrement, or offset addition. When the offset addition is selected, the offset specified by the DMA offset register (DMOFR of DMAC0) is added to the address every time the DMAC performs one data transfer. This function realizes a data transfer where addresses are allocated to separated areas.

Offset subtraction can also be realized by setting a negative value in DMOFR register of DMAC0. In this case, the negative value must be 2's complement.

Address update function using offset can be specified only for the DMAC0 channel.

Table 18.6 shows the address update method in each address update mode.

Table 18.6 Address Update Method in Each Address Update Mode

Address Update Mode	Settings of DMACm.DMAMD.SM[1:0] and DMACm.DMAMD.DM[1:0] for Address Update Modes	Address Update Method (for Different SZ[1:0] Settings in DMTMD of DMACm)		
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b
Address fixed	00b	Fixed		
Offset addition	01b	+DMACm.DMOFR*1		
Increment	10b	+1	+2	+4
Decrement	11b	–1	–2	–4

Note 1. When setting a negative value in the DMA offset register, the value must be 2's complement. The 2's complement is obtained by the following formula.
2's complement of a negative offset value = ~ (offset) + 1 (~: bit inversion)

(1) Basic Transfer Using Offset Addition

Figure 18.7 shows an example of address updating using offset addition.

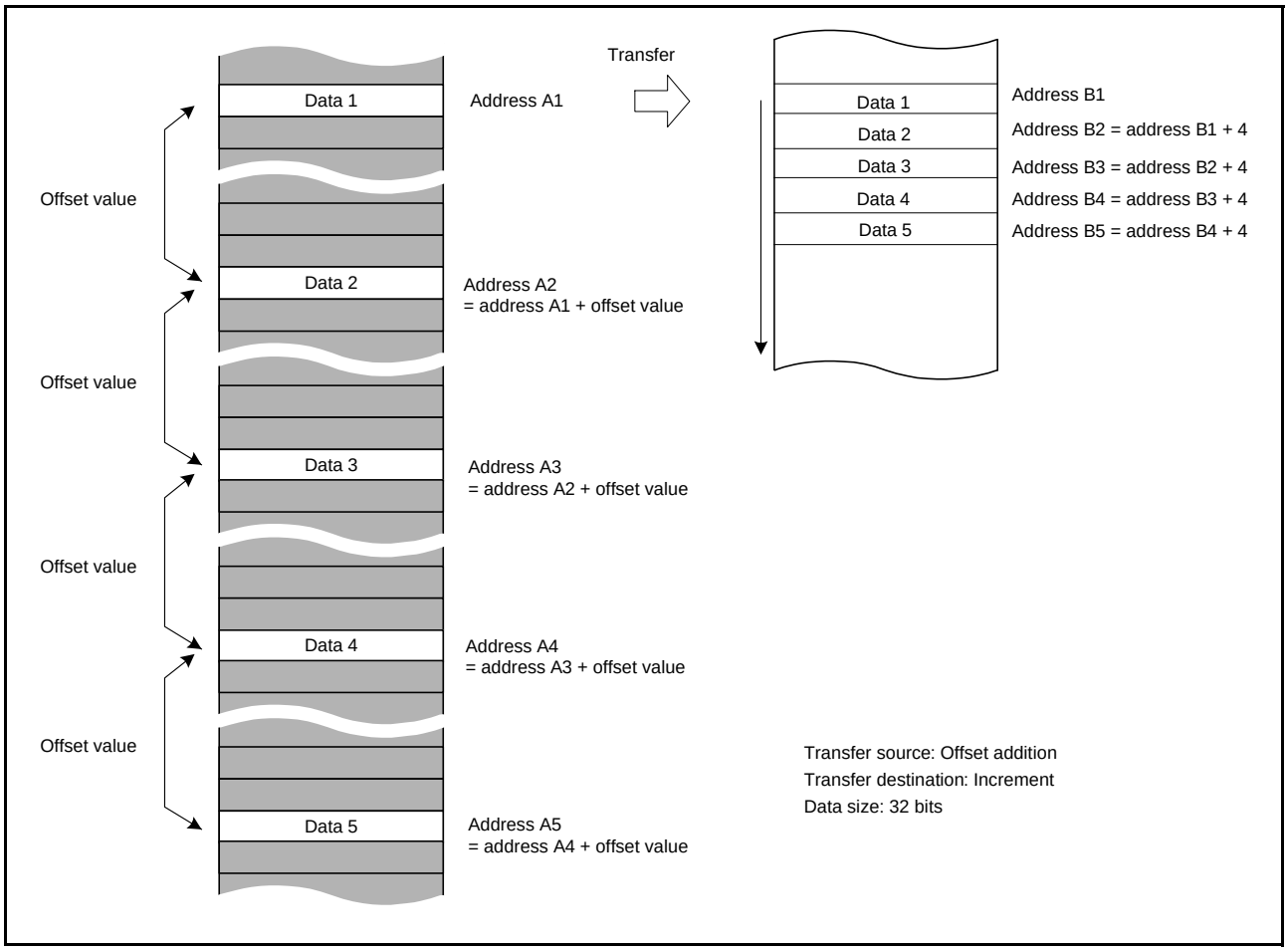


Figure 18.7 Example of Address Updating by Offset Addition

In Figure 18.7, the transfer data is 32 bits long, and offset addition and increment are set as the transfer source address update mode and transfer destination address update mode, respectively. The second and subsequent data is each read from the transfer source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to the continuous locations on the destination.

(2) Example of XY Conversion Using Offset Addition

Figure 18.8 shows the XY conversion using offset addition in repeat transfer mode.
Settings are as follows:

- DMAC0.DMAMD: Transfer source address update mode: Offset addition
- DMAC0.DMAMD: Transfer destination address update mode: Destination address is incremented.
- DMAC0.DMTMD: Transfer data size select: 32 bits
- DMAC0.DMTMD: Transfer mode select: Repeat transfer
- DMAC0.DMTMD: Repeat area select: The source is specified as the repeat area.
- DMAC0.DMOFR: Offset address: 10h
- DMAC0.DMCRA: Repeat size: 4h
- DMAC0.DMINT: The repeat size end interrupt is enabled.

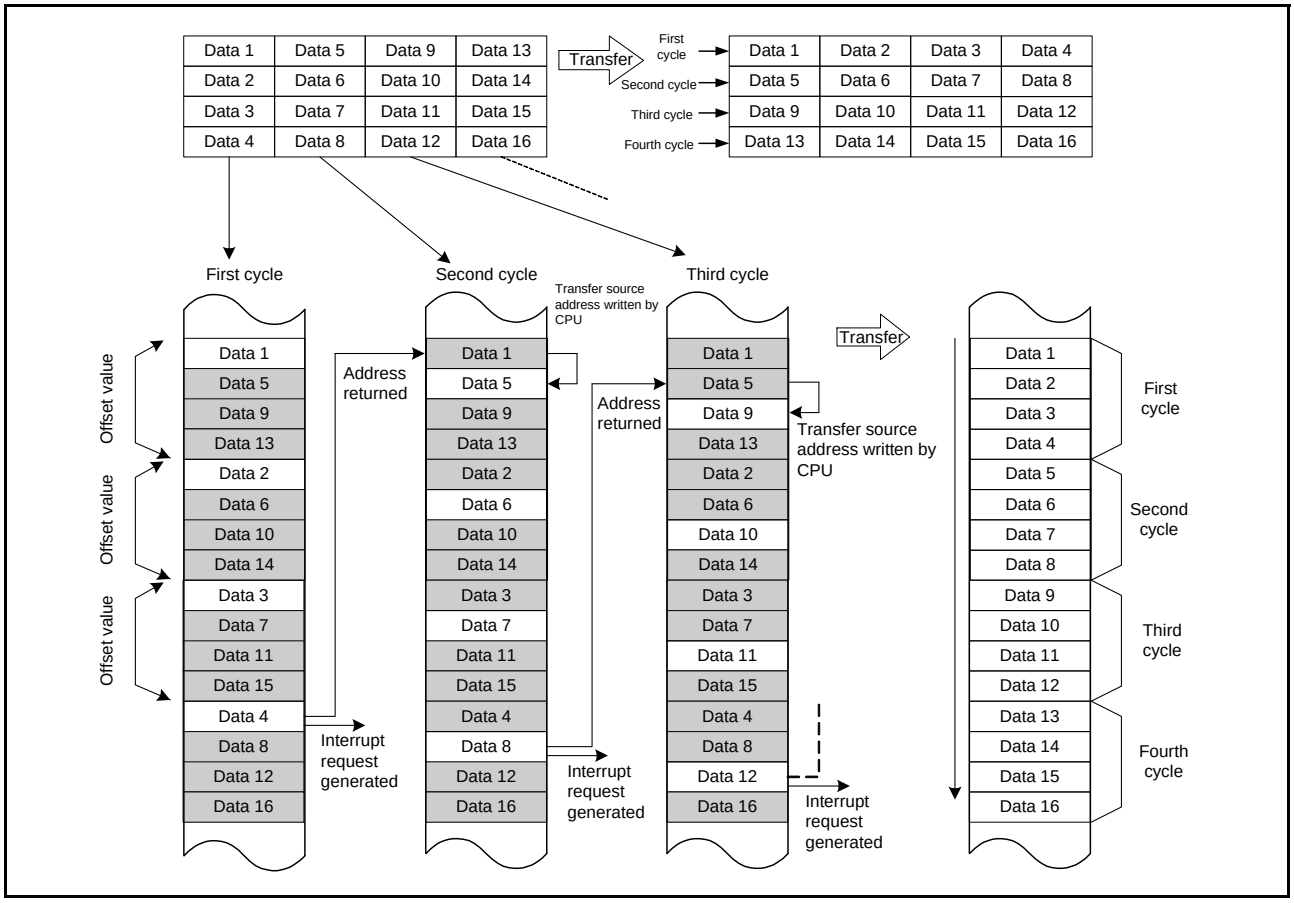


Figure 18.8 XY Conversion Operation Using Offset Addition in Repeat Transfer Mode

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to the destination continuous addresses. When data 4 is transferred, which means that the repeat size of transfers is completed, the transfer source address returns to the transfer start address (address of data 1 on the transfer source) and a repeat size end interrupt is requested. While this interrupt stops the transfer temporarily, perform the following.

- DMAC0.DMSAR: Rewrite the DMA transfer source address to the address of data 5 (with the above example, the data 1 address + 4).
- DMAC0.DMCNT: Set the DTE bit to 1.

The DMA transfer is resumed from the state when the DMA transfer is stopped. After that, the operations described above are repeated until the transfer source data is transposed to the destination area (XY conversion).

Figure 18.9 shows a flowchart of the XY conversion.

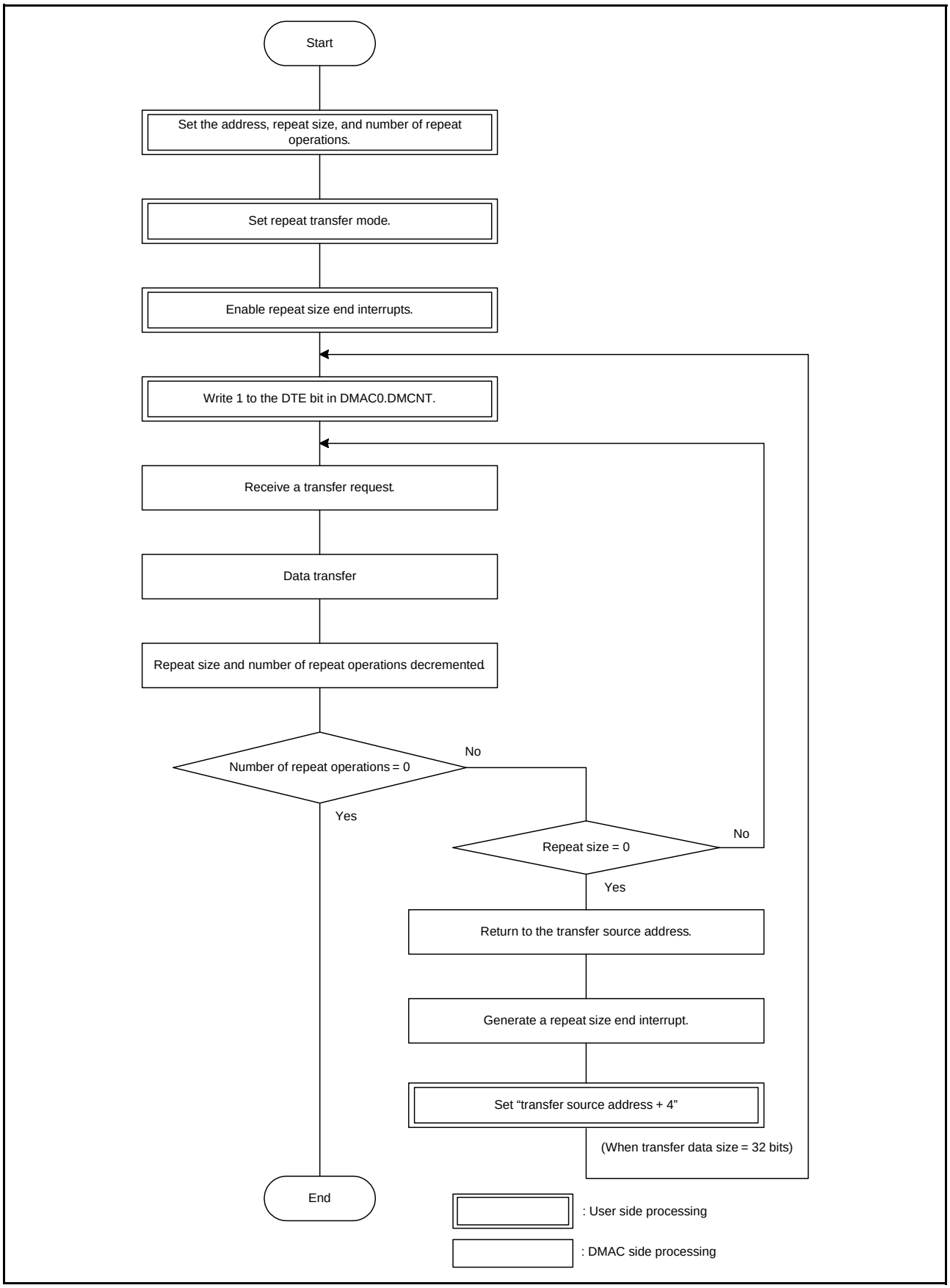


Figure 18.9 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode

18.3.4 Request Sources

Software, the interrupt requests from the peripheral modules, and the external interrupt requests can be specified as the DMA request sources. Setting the DCTG[1:0] bits in DMTMD of DMACm selects the request source.

(1) Trigger by Software

Setting the DCTG[1:0] bits in DMTMD of DMACm to 00b enables the trigger by software.
To start DMA transfer by software, set the DCTG[1:0] bits in DMTMD of DMACm to 00b, and then set the DTE bit in DMCNT of DMACm to 1 (DMA transfer is enabled) and the SWREQ bit in DMREQ of DMACm to 1 (DMA transfer is requested) with the DMST bit in DMAST set to 1 (DMAC module start).
When the DMAC is triggered by software while the CLRS bit in DMREQ of DMACm is 0, the SWREQ bit in DMREQ of DMACm is set to 0 after data transfer is started in response to a DMA transfer request.
When the DMAC is triggered by software while the CLRS bit is 1, the SWREQ bit is not set to 0 after data transfer is started. In this case, a DMA transfer request is issued again after completion of a transfer.

(2) Trigger by Interrupt Requests from On-Chip Peripheral Modules or External Interrupt Requests

Interrupt requests from the on-chip peripheral modules and external interrupt requests can be specified as the DMA request sources. The request source can be selected separately for each channel using the DMRSRm registers (m = 0 to 7) of the ICU.
The DMA transfer is triggered when an interrupt request from the on-chip peripheral module or an external interrupt request is generated while the DCTG[1:0] bits in DMTMD of DMACm are set to 01b (interrupts from the peripheral modules and the external interrupt pins are selected), the DTE bit in DMCNT of DMACm is set to 1 (DMA transfer is enabled), and the DMST bit in DMAST is set to 1 (DMAC module start).

For interrupt requests specified as DMA request sources, refer to Table 15.5, Interrupt Vector Table, in section 15, Interrupt Controller (ICUA).

18.3.5 Operation Timing

Figure 18.10 and Figure 18.11 show DMAC operation timing examples.

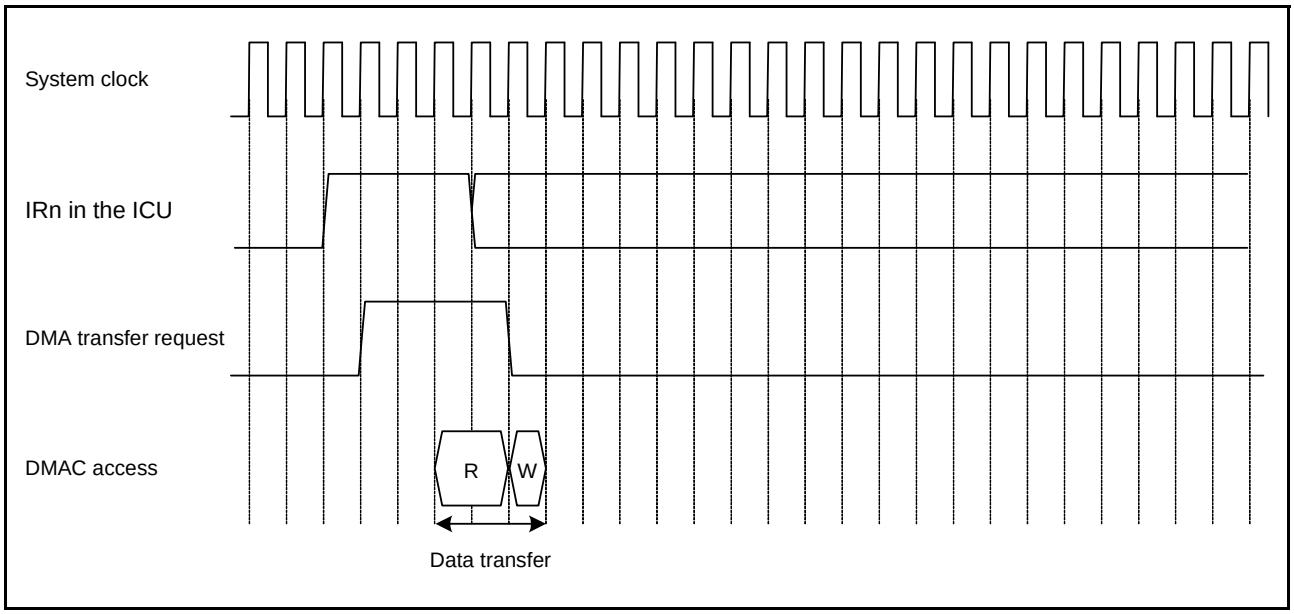


Figure 18.10 DMAC Operation Timing Example (1) (Trigger by Interrupt from Peripheral Module/External Interrupt Input Pin, Normal Transfer Mode, Repeat Transfer Mode)

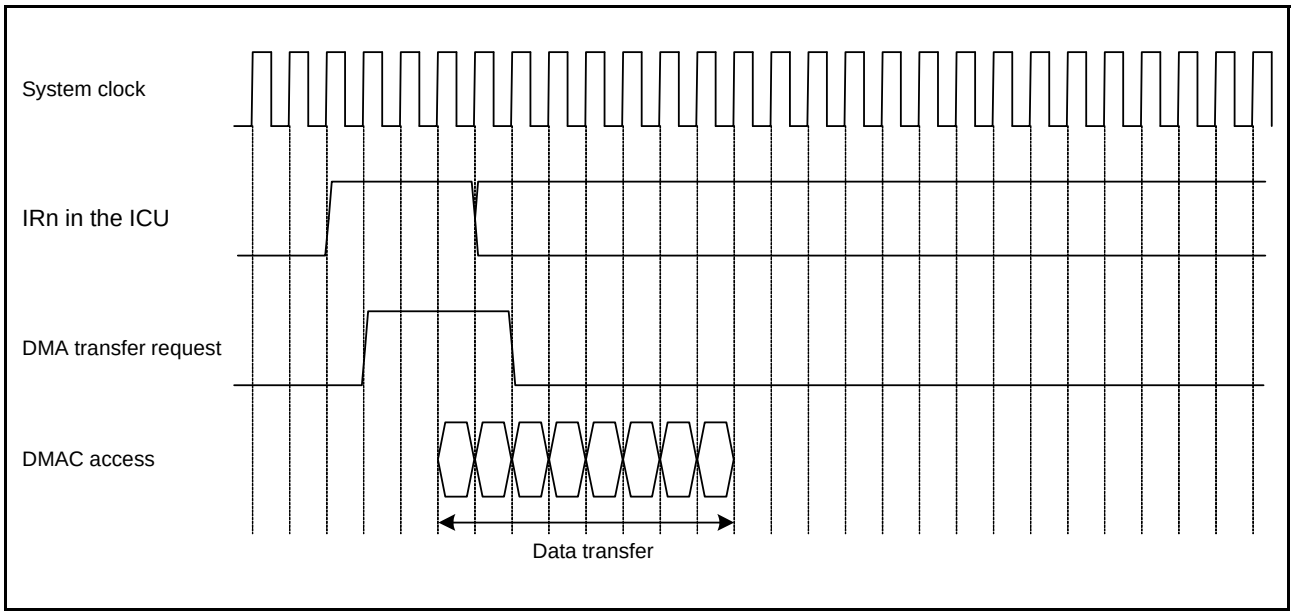


Figure 18.11 DMAC Operation Timing Example (2) (Trigger by Interrupt from Peripheral Module/External Interrupt Input Pin, Block Transfer Mode, Block Size = 4)

18.3.6 DMAC Execution Cycles

Table 18.7 lists execution cycles in one DMAC data transfer operation.

Table 18.7 DMAC Execution Cycles

Transfer Mode	Data Transfer (Read)	Data Transfer (Write)
Normal	Cr+1	Cw
Repeat	Cr+1	Cw
Block*1	P × Cr	P × Cw

Note 1. This is the case when the block size is 2 or more. When the block size is 1, normal transfer cycle is applied.

P: Block size (DMCRAH register setting)
Cr: Data read destination access cycle
Cw: Data write destination access cycle

Cr and Cw depend on the access destination. For the number of cycles for each access destination, see section 61, RAM, section 63, Flash Memory, section 5, I/O Registers, and section 16.2.6, External Bus.
The unit for +1 in “Data Transfer (Read)” column is one system clock cycle (ICLK).
For the operation example, see section 18.3.5, Operation Timing.

18.3.7 Activating the DMAC

Figure 18.12 shows the register setting procedure.

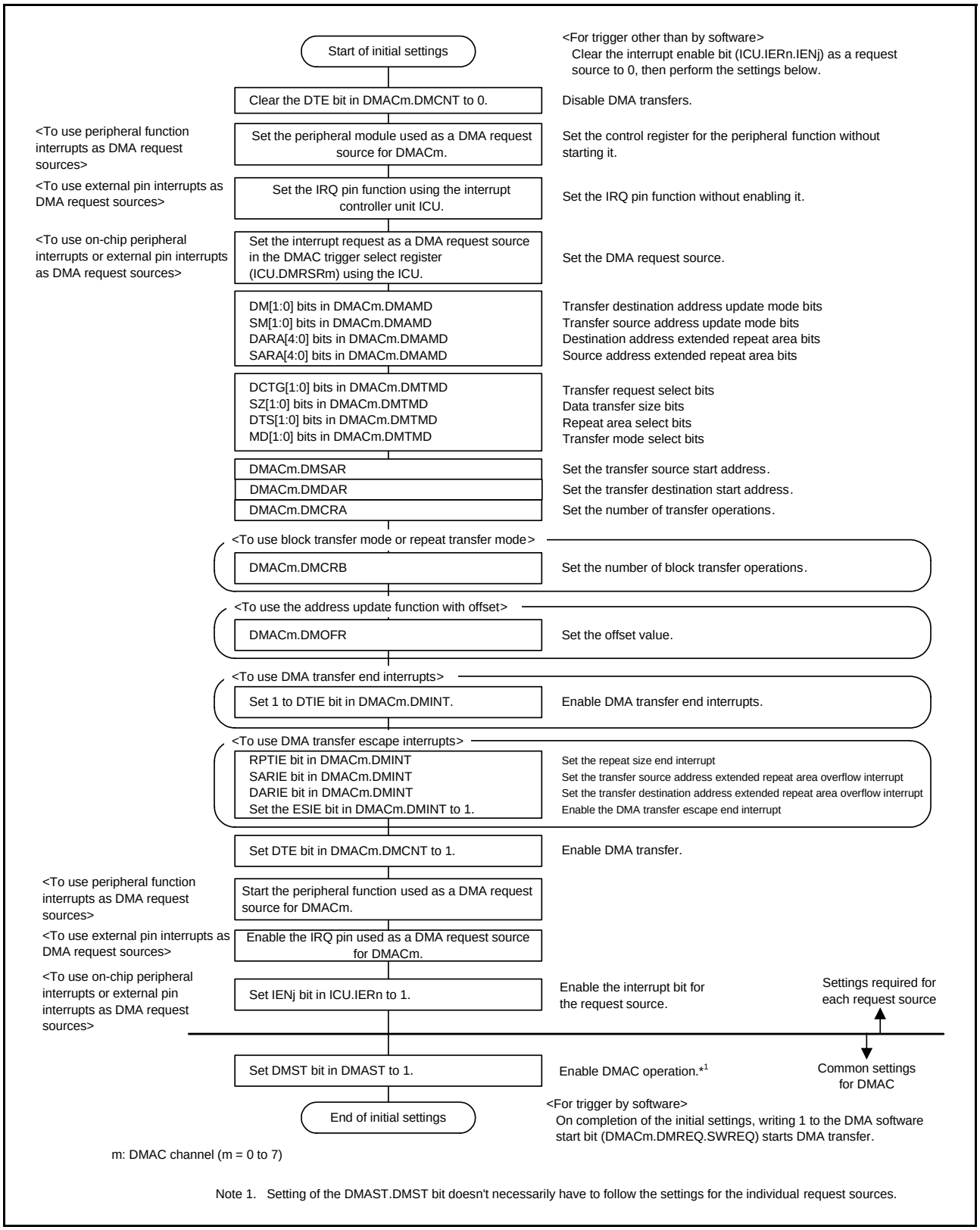


Figure 18.12 Register Setting Procedure

18.3.8 Starting DMA Transfer

Setting the DTE bit in DMCNT of DMACm to 1 (DMA transfer enabled) and setting the DMST bit in DMAST to 1 (DMAC module start) enable DMA transfer of channel m (m = 0 to 7).
Another transfer request cannot be accepted during the transfer of other DMAC channel or DTC. When the proceeding transfer is completed, channel arbitration is performed where a DMA transfer request of the highest priority channel is accepted and DMA transfer of the channel starts. When DMA transfer starts, the ACT bit in DMSTS of DMACm is set to 1 (the DMAC is in the active state).

18.3.9 Registers during DMA Transfer

The DMAC registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers to be updated are DMSAR, DMDAR, DMCRA, DMCRB, DMCNT, and DMSTS of DMACm.

(1) DMA Source Address Register (DMACm.DMSAR)

When data has been transferred in response to one transfer request, the contents of DMSAR are updated to the address to be accessed by the next transfer request.
For details on register update operation in each transfer mode, refer to Table 18.3 to Table 18.5.

(2) DMA Destination Address Register (DMACm.DMDAR)

When data has been transferred in response to one transfer request, the contents of DMDAR are updated to the address to be accessed by the next transfer request.
For details on register update operation in each transfer mode, refer to Table 18.3 to Table 18.5.

(3) DMA Transfer Count Register (DMACm.DMCRA)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.
For details on register update operation in each transfer mode, refer to Table 18.3 to Table 18.5.

(4) DMA Block Transfer Count Register (DMACm.DMCRB)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.
For details on register update operation in each transfer mode, refer to Table 18.3 to Table 18.5.

(5) DMA Transfer Enable Bit (DMACm.DMCNT.DTE)

Although the DMACm.DMCNT.DTE bit enables or disables data transfer by the register write access, it is automatically set to 0 by the DMAC according to the DMA transfer state.
The conditions for clearing this bit by the DMAC are as follows:

- When the specified total volume of data transfer is completed
- When DMA transfer is stopped by the repeat size end interrupt
- When DMA transfer is stopped by the extended repeat area overflow interrupt

Writing to the registers for the channels when the corresponding DMACm.DMCNT.DTE bit is set to 1 is prohibited (except for DMACm.DMCNT). In this case, writing must be performed after the bit is set to 0.

(6) DMA Active Flag (DMACm.DMSTS.ACT)

The ACT bit in DMSTS of DMACm indicates whether the DMACm is in the idle or active state.
This flag is set to 1 when the DMAC starts data transfer, and is set to 0 when data transfer in response to one transfer request is completed.
Even when DMA transfer is stopped by writing 0 to the DTE bit in DMCNT of DMACm during DMA transfer, this flag remains 1 until DMA transfer is completed.

(7) Transfer End Interrupt Flag (DMACm.DMSTS.DTIF)

The DTIF flag in DMSTS of DMACm is set to 1 after DMA transfer of the total transfer size of data is completed.
When both this flag and the DTIE bit in DMINT of DMACm are set to 1, a transfer end interrupt is requested.
This flag is set to 1 when the DMA transfer bus cycle is completed and the ACT flag in DMSTS of DMACm is set to 0 indicating the DMA transfer end.
This flag is automatically set to 0 when the DTE bit in DMCNT of DMACm is set to 1 during the interrupt handling.

(8) Transfer Escape End Interrupt Flag (DMACm.DMSTS.ESIF)

The ESIF flag in DMSTS of DMACm is set to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When this bit and the ESIE bit in DMINT of DMACm are set to 1, a transfer escape end interrupt is requested.
This flag is set to 1 when the bus cycle of the DMA transfer having caused the interrupt request is completed and the ACT flag in DMSTS of DMACm is set to 0 indicating the DMA transfer end.
This flag is automatically set to 0 when the DTE bit in DMCNT of DMACm is set to 1 during an interrupt handling.
Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUA).

18.3.10 Channel Priority

When multiple DMA transfer requests are present, the DMAC determines the priority of channels that have DMA transfer requests.
The channel priority is fixed as channel 0 > channel 1 > channel 2 > channel 3 > channel 4 > channel 5 > channel 6 > channel 7 (channel 0: highest).
When a DMA transfer request is generated during data transfer, channel arbitration is started after the final data has been transferred, and DMA transfer of the higher-priority channel starts.

18.4 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the DTE bit in DMCNT and the ACT flag in DMSTS of DMACm are changed from 1 to 0, indicating that DMA transfer has ended.

18.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations

(1) In Normal Transfer Mode (DMACm.DMTMD.MD[1:0] = 00b)

When the value of DMCRAL of DMACm changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in DMCNT of DMACm is set to 0 and the DTIF bit in DMSTS of DMACm is set to 1 at the same time. If the DTIE bit in DMINT of DMACm is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(2) In Repeat Transfer Mode (DMACm.DMTMD.MD[1:0] = 01b)

When the value of DMCRB of DMACm changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in DMCNT of DMACm is set to 0 and the DTIF bit in DMSTS of DMACm is set to 1 at the same time. If the DTIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC.

(3) In Block Transfer Mode (DMACm.DMTMD.MD[1:0] = 10b)

When the value of DMCRB of DMACm changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in DMCNT of DMACm is set to 0 and the DTIF bit in DMSTS of DMACm is set to 1 at the same time. If the DTIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC.
Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUA).

18.4.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, a repeat size end interrupt is requested when transfer of a 1-repeat size of data is completed while the RPTIE bit in DMINT of DMACm is set to 1. When the interrupt is requested to complete DMA transfer, the DTE bit in DMCNT of DMACm is set to 0 and the ESIF flag in DMSTS of DMACm is set to 1. If the ESIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC. Here, the transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm.
A repeat size end interrupt can be requested also in block transfer mode. In block transfer mode, the interrupt is requested in the same way as in repeat transfer mode when transfer of a 1-block size data is completed.
Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUA).

18.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area occurs while the extended repeat area is specified and the SARIE or DARIE bit in DMINT of DMACm is set to 1, an interrupt by an extended repeat area overflow is requested. When the interrupt is requested, the DMA transfer is terminated, the DTE bit in DMCNT of DMACm is set to 0, and the ESIF flag in DMSTS of DMACm is set to 1. If the ESIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC.
Even if an interrupt by an extended repeat area overflow is requested during a read cycle, the following write cycle is performed.
In block transfer mode, even if an interrupt by an extended repeat area overflow is requested during a 1-block transfer, the remaining data in the block is transferred; transfer is terminated after a block transfer.
Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUA).

18.5 Interrupts

Each DMAC channel can output an interrupt request to the CPU or the DTC after transfer in response to one request is completed. When the transfer destination is the external bus or the on-chip peripheral bus, an interrupt request is generated upon completion of data write to the write buffer not to the actual transfer destination.

Table 18.8 lists the relation among the interrupt sources, the interrupt status flags, and the interrupt enable bits. Figure 18.13 shows the schematic logic diagram of interrupt outputs (DMAC0 to DMAC3). Figure 18.14 shows the schematic logic diagram of interrupt outputs (DMAC4 to DMAC7). Figure 18.15 shows the DMAC interrupt handling routine to resume/terminate DMA transfer.

Table 18.8 Relation among Interrupt Sources, Interrupt Status Flags, and Interrupt Enable Bits

Interrupt Sources		Interrupt Enable Bits	Interrupt Status Flags	Request Output Enable Bits
Transfer end		—	DMACm.DMSTS.DTIF	DMACm.DMINT.DTIE
Escape transfer end	Repeat size end	DMACm.DMINT.RPTIE	DMACm.DMSTS.ESIF	DMACm.DMINT.ESIE
	Source address extended repeat area overflow	DMACm.DMINT.SARIE		
	Destination address extended repeat area overflow	DMACm.DMINT.DARIE		

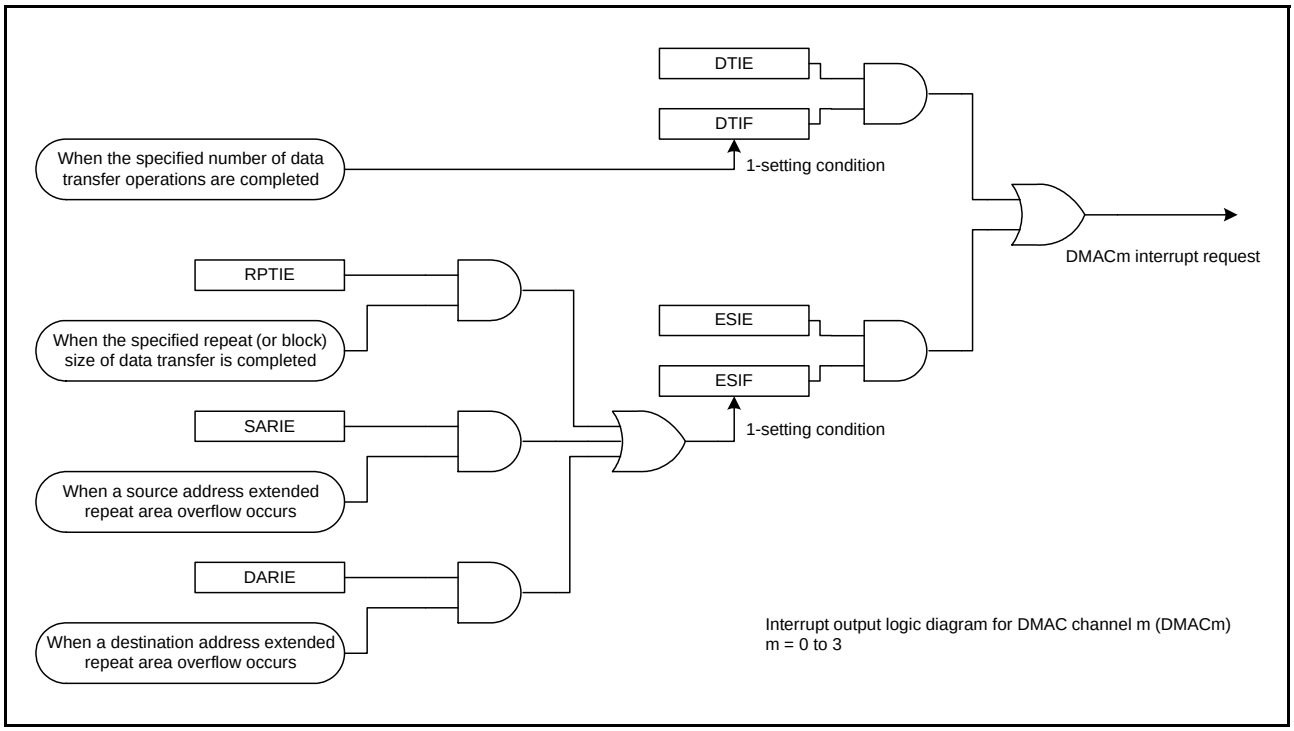


Figure 18.13 Schematic Logic Diagram of Interrupt Outputs (DMAC0 to DMAC3)

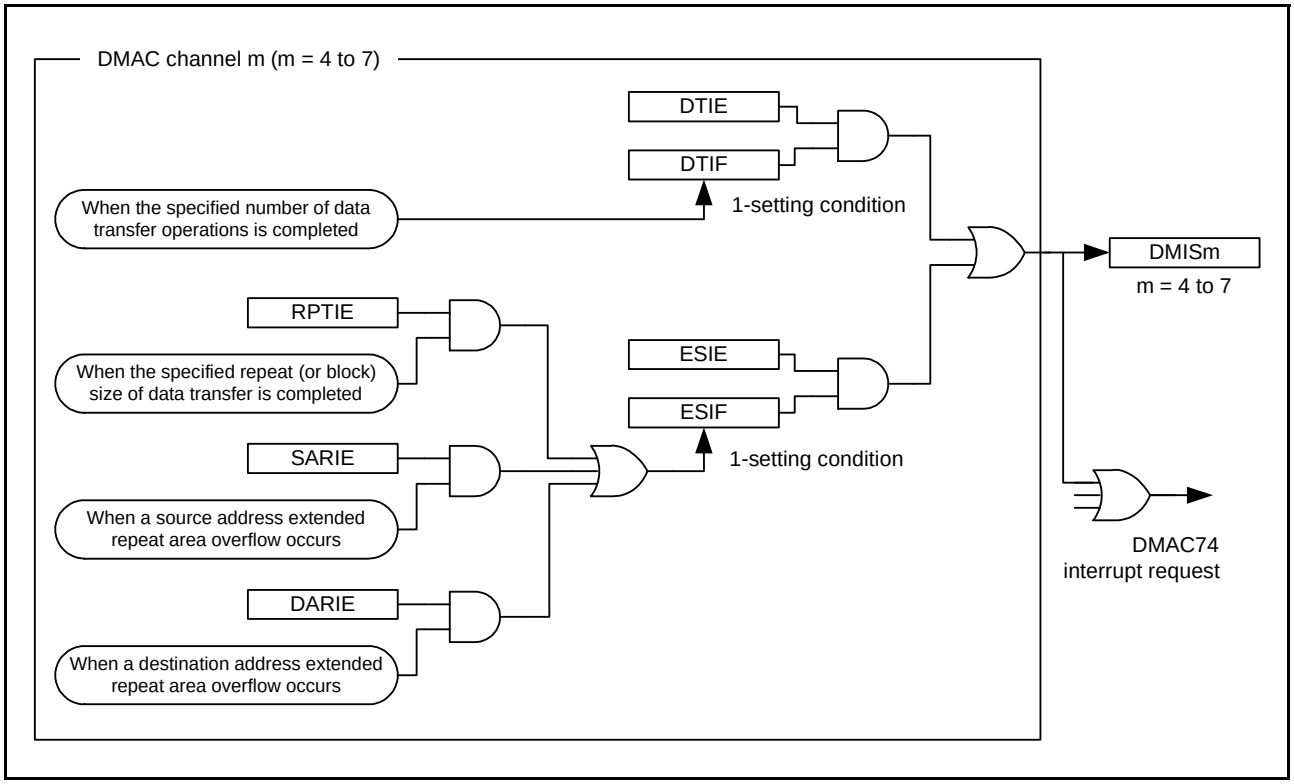


Figure 18.14 Schematic Logic Diagram of Interrupt Outputs (DMAC4 to DMAC7)

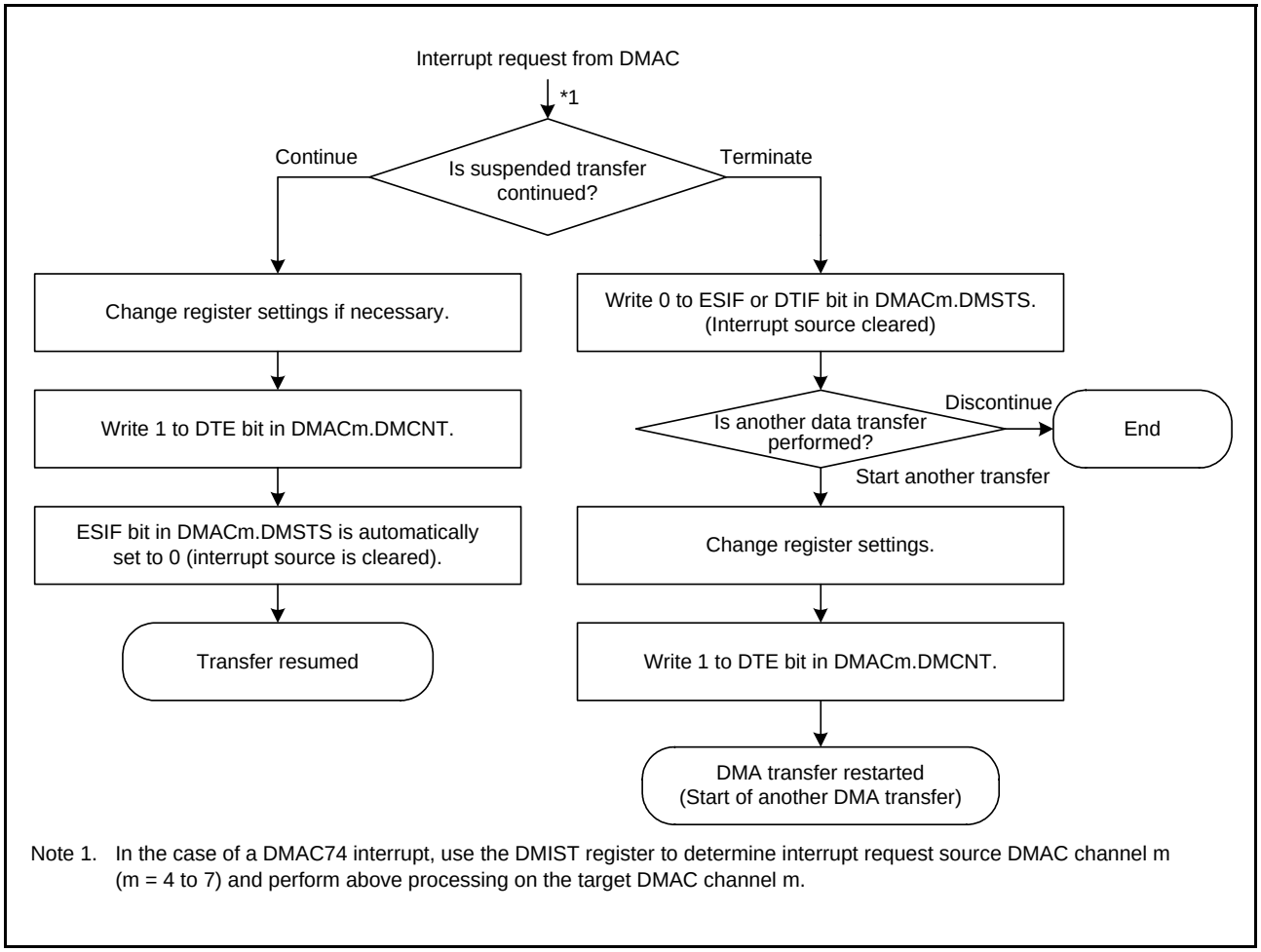
Specifically, the different procedures are used for canceling an interrupt to restart DMA transfer in the following two cases: (1) discontinuing or terminating DMA transfer and (2) continuing DMA transfer.

(1) When Discontinuing or Terminating DMA Transfer

Write 0 to the DTIF bit in DMSTS of DMACm to clear a transfer end interrupt, and to the ESIF bit in DMSTS of DMACm to clear a repeat size interrupt and an extended repeat area overflow interrupt. The DMACm remains in the stop state. When starting another DMA transfer after that, set the appropriate registers, and set the DTE bit in DMCNT of DMACm to 1 (DMA transfer enabled).

(2) When Continuing DMA Transfer

Write 1 to the DTE bit in DMCNT of DMACm. The ESIF bit in DMSTS of DMACm is automatically set to 0 (interrupt source cleared), and DMA transfer is resumed.



Note 1. In the case of a DMAC74 interrupt, use the DMIST register to determine interrupt request source DMAC channel m (m = 4 to 7) and perform above processing on the target DMAC channel m.

Figure 18.15 DMAC Interrupt Handling Routine to Resume/Terminate DMA Transfer

18.6 Event Link

Each DMAC channel outputs an event link request signal each time the channel completes data transfer (or block transfer in block transfer mode). However, when the transfer destination is the external bus or internal peripheral bus, an event link request signal is generated when the writing to the write buffer is accepted.

18.7 Low-Power Consumption Function

Before transition to the module-stop state, all-module clock stop mode, software standby mode, or deep software standby mode, clear the DMST bit in DMAST to 0 (the DMAC suspended), and then perform the following.

(1) Module-Stop Function

Writing 1 to the MSTPA28 bit (transition to the module-stop state) in MSTPCRA enables the module-stop function of the DMAC. If DMA transfer is in progress at the time a 1 is written to the MSTPA28 bit, the transition to the module-stop state proceeds after DMA transfer has ended. While the MSTPA28 bit is 1, accessing the DMAC registers are prohibited.
Writing 0 to the MSTPA28 bit releases the DMAC from the module-stop state.

(2) All-Module Clock Stop Mode

Make settings in accord with the procedure under section 11.6.2.1, Transition to All-Module Clock Stop Mode, in section 11, Low Power Consumption.
If DMA transfer operations are in progress at the time the WAIT instruction is executed, the transition to all-module clock stop mode follows the completion of DMA transfer.
The DMAC is released from the module-stop state by writing 0 to the MSTPCRA.MSTPA28 bit following recovery from all-module clock stop mode.

(3) Software Standby and Deep Software Standby Modes

Make settings in accord with the procedure under section 11.6.3.1, Transition to Software Standby Mode or section 11.6.4.1, Transition to Deep Software Standby Mode, in section 11, Low Power Consumption.
If DMA transfer operations are in progress at the time the WAIT instruction is executed, the transition to software standby or deep software standby follows the completion of DMA transfer.

(4) Note on Low-Power Consumption Function

For the WAIT instruction and the register setting procedure, see section 11.7.6, Timing of Wait Instructions in section 11, Low Power Consumption.
To perform DMA transfer after returning from low-power consumption mode, set the DMST bit in DMAST to 1 again.
To use a request that is generated in all-module clock-stop mode and software standby mode as an interrupt request to the CPU but not as a DMAC startup request, specify the CPU as the interrupt request destination in accordance with the description in section 15, Interrupt Controller (ICUA), and then execute the WAIT instruction.

18.8 Usage Notes

18.8.1 DMA Transfer to External Devices

In DMA transfer to an external device, the ACT bit in DMSTS of DMACm may be set to 0 (DMAC transfer suspended) during the period from the beginning of the final data write to the end of the external bus access.

18.8.2 DMA Transfer to Peripheral Modules

In DMA transfer to a peripheral module, the ACT bit in DMSTS of DMACm may be set to 0 (DMAC transfer suspended) during the period from the beginning of the final data write to the end of the peripheral bus access.

18.8.3 Access to the Registers during DMA Transfer

Do not write to the DMSAR, DMDAR, DMCRA, DMCRB, DMTMD, DMINT, DMAMD, DMOFR, and DMCSL registers of DMACm while the ACT bit in DMSTS of the same channel is set to 1 (DMAC active state) or the DTE bit in DMCNT of the same channel is set to 1 (DMA transfer enabled).

18.8.4 DMA Transfer to Reserved Areas

DMA transfer to the reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on the reserved areas, see section 4, Address Space.

18.8.5 Interrupt Request by the DMA Request Source Flag Control Register (DMCSL) at the End of Each Transfer

While the DMACm.DMCSL.DISEL bit is 1, an interrupt request is issued to the CPU at the end of each transfer that has been triggered by one DMA request. Unlike the transfer end interrupt that the DMAC outputs or the escape end interrupt, the interrupt of this type is issued to the CPU at the end of DMA transfer without clearing the interrupt status flag of the DMA request source to 0 by changing the interrupt request destination to the CPU. In this case, since the interrupt status flag is not set to 0 at the end of DMAC transfer, it should be set to 0 by the CPU interrupt routine.
The interrupt flag is cleared when the CPU interrupt is accepted.
For the change of the settings on the interrupt flag or the interrupt request destination, see section 15, Interrupt Controller (ICUA). For the DMACm.DMCSL.DISEL bit setting, see section 18.2.12, DMA Request Source Flag Control Register (DMCSL).

18.8.6 Setting of DMAC Trigger Select Register of the Interrupt Controller (ICU.DMRSRm)

The DMAC trigger select register (ICU.DMRSRm) should be set while the DMA transfer enable bit (DMACm.DMCNT.DTE) is 0 (DMA transfer is disabled). Moreover, the DTC transfer request enable register (ICU.DTCERm) that corresponds to the same vector number that has been set by the ICU.DMRSRm register should not be set to 1. For details on the ICU.DTCERn and ICU.DMRSRm, see section 15, Interrupt Controller (ICUA).

18.8.7 Suspending or Restarting DMA Transfer

To suspend a DMA transfer request, write 0 to the interrupt enable bit for the request source (ICU.IERn.IENj bit). To restart the DMA transfer, write 1 to the ICU.IERn.IENj bit with the setting shown in section 18.3.7, Activating the DMAC.

19. EXDMA Controller (EXDMACa)

This MCU incorporates a 2-channel direct memory access controller (EXDMAC) designed exclusively for external bus transfer. The EXDMAC is a module to transfer data without the CPU. When a DMA transfer request is generated, the EXDMAC transfers data stored at the transfer source address to the transfer destination address.

19.1 Overview

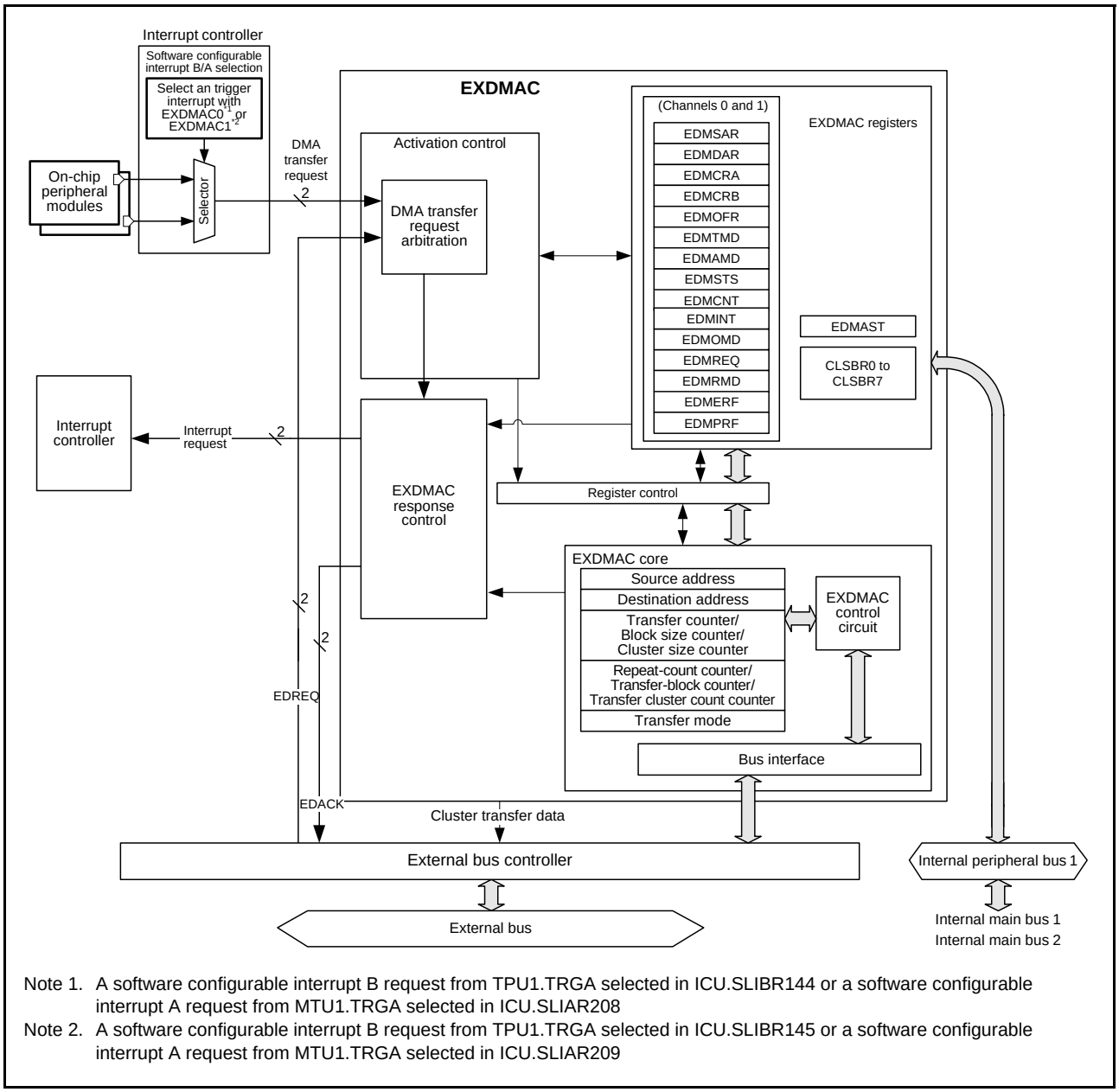
Table 19.1 lists the specifications of the EXDMAC, and Figure 19.1 shows a block diagram of the EXDMAC.

Table 19.1 Specifications of EXDMAC (1/2)

Item		Description
Number of channels		2 (EXDMAC0 and EXDMAC1)
Transfer space		512 Mbytes (External areas at addresses 0000 0000h to 0FFF FFFFh and F000 0000h to FFFF FFFFh excluding reserved areas)
Maximum transfer volume		1 M data (Maximum number of transfer operations in block transfer mode: 1,024 data × 1,024 blocks)
DMA request source		• Request source selectable from the following three sources for each channel Software trigger External DMA transfer request input DMA transfer request from peripheral modules (TPU1.TRGA or MTU1.TRGA) (Channel 0: a software configurable interrupt B request from TPU1.TRGA selected in ICU.SLIBR144 or a software configurable interrupt A request from MTU1.TRGA selected in ICU.SLIAR208; Channel 1: a software configurable interrupt B request from TPU1.TRGA selected in ICU.SLIBR145 or a software configurable interrupt A request from MTU1.TRGA selected in ICU.SLIAR209)
Channel priority		Channel 0 > Channel 1 (Channel 0: Highest)
Transfer data	Single data	Bit length: 8, 16, 32 bits
	Block size	Number of data: 1 to 1,024 data
	Cluster size	Number of data: 1 to 8 data
Transfer mode	Normal transfer mode	• One data transfer by one DMA transfer request • Free running mode (setting in which total number of data transfer operations is not specified) settable
	Repeat transfer mode	• One data transfer by one DMA transfer request • Program returns to the transfer start address on completion of the repeat size of data transfer specified for the transfer source or destination. • Maximum settable repeat size: 1,024 data
	Block transfer mode	• One block data transfer by one DMA transfer request • Maximum settable block size: 1,024 data
	Cluster transfer mode	• One cluster data transfer by one DMA transfer request • Maximum settable cluster size: 8 data (32 bytes)
Address mode	Single address mode	• Transfers data by accessing the transfer source or destination peripheral device with the EDACKn signal (n = 0, 1) and specifying the address of the other peripheral device. • Available in normal transfer mode, repeat transfer mode, and block transfer mode.
	Dual address mode	• Transfers data by specifying the addresses of transfer source and destination. • Available in normal transfer mode, repeat transfer mode, block transfer mode, and cluster transfer mode.
Selective functions	Extended repeat area function	• Function in which data can be transferred by repeating the address values in the specified range with the upper bit values in the transfer address register fixed • Area of 2 bytes to 128 Mbytes separately settable as extended repeat area for transfer source and destination

Table 19.1 Specifications of EXDMAC (2/2)

Item		Description
Interrupt request	Transfer end interrupt	Generated when the specified number of transfers is completed in normal transfer mode Generated when the specified repeat count of transfers is completed in repeat transfer mode Generated when the specified block count of transfers is completed in block transfer mode Generated when the specified cluster count of transfers is completed in cluster transfer mode
	Transfer escape end interrupt	Generated when the repeat size of data transfer is completed or the extended repeat area overflows.
Low-power consumption function		The module-stop state can be set.



Note 1. A software configurable interrupt B request from TPU1.TRGA selected in ICU.SLIBR144 or a software configurable interrupt A request from MTU1.TRGA selected in ICU.SLIAR208

Note 2. A software configurable interrupt B request from TPU1.TRGA selected in ICU.SLIBR145 or a software configurable interrupt A request from MTU1.TRGA selected in ICU.SLIAR209

Figure 19.1 Block Diagram of EXDMAC

Table 19.2 lists the input/output pins of the EXDMAC.

Table 19.2 Pin Configuration of EXDMAC

Channel	Pin Name	I/O	Description
EXDMAC0	EDREQ0	Input	EXDMAC0 external DMA transfer request
	EDACK0	Output	EXDMAC0 single address transfer acknowledge
EXDMAC1	EDREQ1	Input	EXDMAC1 external DMA transfer request
	EDACK1	Output	EXDMAC1 single address transfer acknowledge

19.2 Register Descriptions

19.2.1 EXDMA Source Address Register (EDMSAR)

Address(es): EXDMAC0.EDMSAR 0008 2800h, EXDMAC1.EDMSAR 0008 2840h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer source start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

This register is used to set the start address of the transfer source.
Set the EDMSAR register while EXDMAC stops (EDMAST.DMST bit = 0) or DMA transfer is disabled (EDMCNT.DTE bit = 0).
Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading the EDMSAR register returns the extended value.

19.2.2 EXDMA Destination Address Register (EDMDAR)

Address(es): EXDMAC0.EDMDAR 0008 2804h, EXDMAC1.EDMDAR 0008 2844h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer destination start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

This register is used to set the start address of the transfer destination.
Set the EDMDAR register while EXDMAC stops (EDMAST.DMST bit = 0) or DMA transfer is disabled (EDMCNT.DTE bit = 0).
Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading the EDMDAR register returns the extended value.

19.2.3 EXDMA Transfer Count Register (EDMCRA)

Address(es): EXDMAC0.EDMCRA 0008 2808h, EXDMAC1.EDMCRA 0008 2848h

· Normal transfer mode

	EDMCRAH															
	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	EDMCRAL															
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

· Repeat transfer mode, block transfer mode

	EDMCRAH															
	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	EDMCRAL															
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	0	0	0	0	0	0										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: The function differs depending on the transfer mode.

· Cluster transfer mode

	EDMCRAH															
	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	EDMCRAL															
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	0	0	0	0	0	0	0	0	0	0	0	0	0			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Symbol	Bit Name	Description	R/W
EDMCRAL	Lower bits of transfer count	Specifies the number of transfer operations.	R/W
EDMCRAH	Upper bits of transfer count		R/W

Note: Set the same value for EDMCRAH and EDMCRAL in repeat transfer mode, block transfer mode, and cluster transfer mode.

This register is used to specify the transfer count of the DMA. The function of this register depends on transfer mode.

(1) Normal transfer mode (EXDMACn.EDMTMD.MD[1:0] bits = 00b)

EDMCRAL functions as a 16-bit transfer counter.
The number of transfer operations is one when the setting is 0001h, and 65535 when it is FFFFh. The value is decremented by one each time data is transferred.
When the setting is 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode).
EDMCRAH is not used in normal transfer mode. Write 0000h to EDMCRAH.

(2) Repeat transfer mode (EXDMACn.EDMTMD.MD[1:0] bits = 01b)

EDMCRAH specifies the repeat size and EDMCRAL functions as a 10-bit transfer counter.
The number of transfer operations is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In repeat transfer mode, a value in the range of 001h to 3FFh (the number of transfer operations: 1 to 1024) can be set for EDMCRAH and EDMCRAL.
Setting bits 15 to 10 in EDMCRAL is invalid. Write 0 to these bits.
The value in EDMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in EDMCRAH is loaded into EDMCRAL.

(3) Block transfer mode (EXDMACn.EDMTMD.MD[1:0] bits = 10b)

EDMCRAH specifies the block size and EDMCRAL functions as a 10-bit block size counter.
The block size is 1 when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In block transfer mode, a value in the range of 001h to 3FFh (the number of transfer operations: 1 to 1024) can be set for EDMCRAH and EDMCRAL.
Setting bits 15 to 10 in EDMCRAL is invalid. Write 0 to these bits.
The value in EDMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in EDMCRAH is loaded into EDMCRAL.

(4) Cluster transfer mode (EXDMACn.EDMTMD.MD[1:0] bits = 11b)

EDMCRAH specifies the cluster size and EDMCRAL functions as a 3-bit cluster size counter.
The cluster size is 1 when the setting is 001h, 7 when it is 007h, and 8 when it is 000h. In cluster transfer mode, a value in the range of 000h to 007h (the number of transfer operations: 1 to 8) can be set for EDMCRAH and EDMCRAL.
Setting bits 15 to 3 in EDMCRAL is invalid. Write 0 to these bits.
The value in EDMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in EDMCRAH is loaded into EDMCRAL.

19.2.4 EXDMA Block Transfer Count Register (EDMCRB)

Address(es): EXDMAC0.EDMCRB 0008 280Ch, EXDMAC1.EDMCRB 0008 284Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b9 to b0	Specifies the block count of transfers in block transfer mode, the repeat count in repeat transfer mode, or the transfer cluster count in cluster transfer mode.	001h to 3FFh (1 to 1023) 000h (1024)	R/W
b15 to b10	Reserved	These bits are always read as 0. The write value should be 0.	R/W

This register is used to specify the transfer block count in block transfer mode, the repeat count in repeat transfer mode, or the transfer cluster count in cluster transfer mode.

Decremented (by 1) when the last data of 1 repeat size is transferred in repeat transfer mode.

Decremented (by 1) when the last data of 1 block size is transferred in block transfer mode.

Decremented (by 1) when the last data of 1 cluster size is transferred in cluster transfer mode.

In normal transfer mode, EDMCRB is not used and setting this register is invalid.

19.2.5 EXDMA Transfer Mode Register (EDMTMD)

Address(es): EXDMAC0.EDMTMD 0008 2810h, EXDMAC1.EDMTMD 0008 2850h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MD[1:0]		DTS[1:0]		—	—	SZ[1:0]		—	—	—	—	—	—	DCTG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DCTG[1:0]	Transfer Request Source Select	b1 b0 0 0: Software 0 1: Setting prohibited 1 0: External DMA transfer request pin (EDREQn) 1 1: DMA transfer request from peripheral modules (TPU1.TRGA or MTU1.TRGA) *1	R/W
b7 to b2	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b9, b8	SZ[1:0]	Transfer Data Size Select	b9 b8 0 0: 8 bits 0 1: 16 bits 1 0: 32 bits 1 1: Setting prohibited	R/W
b11, b10	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b13, b12	DTS[1:0]	Repeat Area Select	b13 b12 0 0: The destination is specified as the repeat area or block area. 0 1: The source is specified as the repeat area or block area. 1 0: The repeat area or block area is not specified. 1 1: Setting prohibited	R/W
b15, b14	MD[1:0]	Transfer Mode Select	b15 b14 0 0: Normal transfer 0 1: Repeat transfer 1 0: Block transfer 1 1: Cluster transfer	R/W

Note 1. EXDMAC0: A software configurable interrupt B request from TPU1.TRGA selected in ICU.SLIBR144 or a software configurable interrupt A request from MTU1.TRGA selected in ICU.SLIAR208
EXDMAC1: A software configurable interrupt B request from TPU1.TRGA selected in ICU.SLIBR145 or a software configurable interrupt A request from MTU1.TRGA selected in ICU.SLIAR209
For the setting procedure, see section 19, Request Sources, (3).

This register is used to set DMA transfer mode.

DCTG[1:0] Bits (Transfer Request Source Select)

These bits select the startup source of the EXDMAC from software, the external DMA transfer request pin, or a DMA transfer request generated from a peripheral module.

SZ[1:0] Bits (Transfer Data Size Select)

These bits select data size for a single data transfer from 8, 16, and 32 bits.

DTS[1:0] Bits (Repeat Area Select)

DTS[1:0] select either the source or destination as the repeat area in repeat, block, or cluster transfer mode. In normal transfer mode, setting these bits is invalid.

MD[1:0] Bits (Transfer Mode Select)

These bits specify DMA transfer mode from normal, repeat, block, and cluster.

19.2.6 EXDMA Output Setting Register (EDMOMD)

Address(es): EXDMAC0.EDMOMD 0008 2812h, EXDMAC1.EDMOMD 0008 2852h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	DACKS	DACKE	DACKW	DACKSEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DACKSEL	EDACKn Pin Toggling Select	0: EDACKn pin toggle is disabled. 1: EDACKn pin toggle is enabled.	R/W
b1	DACKW	EDACKn Pin Negate Wait	0: The EDACKn pin is negated at the same time as the RD# or WRn# pin is negated. 1: The EDACKn is negated one BCLK cycle before the RD# pin is negated or one BCLK cycle after the WRn# pin is negated.	R/W
b2	DACKE	EDACKn Pin Output Enable	0: EDACKn output is disabled. 1: EDACKn output is enabled.	R/W
b3	DACKS	EDACKn Pin Output Polarity Select	0: EDACKn pin polarity is active low. 1: EDACKn pin polarity is active high.	R/W
b7 to b4	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

This register is used to set the output signal of the EXDMAC.

DACKSEL Bit (EDACKn Pin Toggling Select)

This bit is used to disable or enable toggled outputs on the EDACKn pin during normal, repeated, or block transfer to and from the SDRAM area in single-address mode.

If the DACKSEL bit is 0 during normal, repeated, or block transfer to and from the SDRAM area in single-address mode (i.e. while the EDMAMD.AMS bit is 1), the signal on the EDACKn pin is asserted throughout the interval where the data is valid (data-valid interval). If the DACKSEL bit is 1, the signal on the EDACKn pin is only asserted for one-half of the SDCLK period in the latter half of the data-valid interval.

The value of the DACKSEL pin has no effect in the case of CS areas.

The value of the DACKSEL pin also has no effect for transfer in dual-address mode and in cluster mode.

In these cases, there is no output on the EDACKn pin.

DACKW Bit (EDACKn Pin Negate Wait)

DACKW selects the EDACKn pin negation timing in single address mode during normal, repeat, or block transfer to/from the CS area.

During the above transfer in single address mode (EDMAMD.AMS bit = 1), the EDACKn pin is negated at the same time as the RD# or WRn# pin is negated if this bit is 0; and the EDACKn pin is negated one BCLK cycle before the RD# pin is negated, or one BCLK cycle after the WRn# pin is negated if this bit is 1. In the SDRAM area, setting this bit is invalid. EDACKn pin negation timing cannot be changed. Setting this bit is also invalid in dual address mode and during cluster transfer. In these cases, the EDACKn pin does not provide output.

DACKE Bit (EDACKn Pin Output Enable)

DACKE enables or disables EDACKn pin output. Setting this bit is invalid in dual address mode and during cluster transfer (EDACKn pin output not provided).

19.2.7 EXDMA Interrupt Setting Register (EDMINT)

Address(es): EXDMAC0.EDMINT 0008 2813h, EXDMAC1.EDMINT 0008 2853h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the destination address 1: Enables an interrupt request for an extended repeat area overflow on the destination address	R/W
b1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the source address 1: Enables an interrupt request for an extended repeat area overflow on the source address	R/W
b2	RPTIE	Repeat Size End Interrupt Enable	0: Disables the repeat size end interrupt request. 1: Enables the repeat size end interrupt request.	R/W
b3	ESIE	Transfer Escape End Interrupt Enable	0: Disables the transfer escape end interrupt request. 1: Enables the transfer escape end interrupt request.	R/W
b4	DTIE	Transfer End Interrupt Enable	0: Disables the transfer end interrupt request. 1: Enables the transfer end interrupt request.	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

This register is used to set the interrupt request output of the EXDMAC.

DARIE Bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the destination address occurs while this bit is set to 1, the EDMCNT.DTE bit is set to 0 (DMA transfer is disabled). At the same time, the EDMSTS.ESIF flag is set to 1 to indicate that an interrupt by an extended repeat area overflow on the destination address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 (DMA transfer is enabled) in the EDMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the destination address, this bit is ignored.

SARIE Bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the source address occurs while this bit is set to 1, the DTE bit in EDMCNT is set to 0 (DMA transfer is disabled). At the same time, the EDMSTS.ESIF flag is set to 1 to indicate that an interrupt by an extended repeat area overflow on the source address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 (DMA transfer is enabled) in the EDMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the source address, this bit is ignored.

RPTIE Bit (Repeat Size End Interrupt Enable)

When this bit is set to 1 in repeat transfer mode, the EDMCNT.DTE bit is set to 0 (DMA transfer is disabled) after completion of a 1-repeat size data transfer. At the same time, the EDMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the EDMTMD.DTS[1:0] bits are 10b (repeat area or block area is not specified).
When this bit is set to 1 in block transfer mode, the EDMCNT.DTE bit is set to 0 after completion of a 1-block data transfer in the same way as repeat transfer mode. At the same time, the EDMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the EDMTMD.DTS[1:0] bits are 10b.
When this bit is set to 1 in cluster transfer mode, the EDMCNT.DTE bit is set to 0 after completion of a 1-cluster data transfer in the same way as repeat transfer mode. At the same time, the EDMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the EDMTMD.DTS[1:0] bits are 10b.

ESIE Bit (Transfer Escape End Interrupt Enable)

This bit enables or disables the transfer escape end interrupt requests (repeat size end interrupt request and extended repeat area overflow interrupt request) that are generated during DMA transfer.
The transfer escape end interrupt is generated when the ESIF flag in EXDMACn.EDMSTS is set to 1 with this bit set to1. The transfer escape end interrupt is cleared by setting this bit or the EDMSTS.ESIF flag to 0.

DTIE Bit (Transfer End Interrupt Enable)

This bit enables or disables the transfer end interrupt request to be generated on completion of a specified number of data transfers.
The transfer end interrupt is generated when the EDMSTS.DTIF flag is set to 1 with this bit set to 1. The transfer end interrupt is cleared by setting this bit or the EDMSTS.DTIF flag to 0.

19.2.8 EXDMA Address Mode Register (EDMAMD)

Address(es): EXDMAC0.EDMAMD 0008 2814h, EXDMAC1.EDMAMD 0008 2854h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AMS	DIR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SM[1:0]		—	SARA[4:0]					DM[1:0]		—	DARA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DARA[4:0]	Destination Address Extended Repeat Area Set	Specifies the extended repeat area on the destination address. For details on the settings, see Table 19.3.	R/W
b5	—	Reserved	This bit is always read as 0. The write value should be 0.	R/W
b7, b6	DM[1:0]	Destination Address Update Mode Set	b7 b6 0 0: Destination address is fixed. 0 1: Offset addition*1 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
b12 to b8	SARA[4:0]	Source Address Extended Repeat Area Set	Specifies the extended repeat area on the source address. For details on the settings, see Table 19.3.	R/W
b13	—	Reserved	This bit is always read as 0. The write value should be 0.	R/W
b15, b14	SM[1:0]	Source Address Update Mode Set	b15 b14 0 0: Destination address is fixed. 0 1: Offset addition*1 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
b16	DIR	Single Address Direction Select	0: Data is transferred in single address mode using the EDMSAR register value as the transfer source address. EDACKn is output to the transfer destination. 1: Data is transferred in single address mode using the EDMDAR register value as the transfer destination address. EDACKn is output to the transfer destination.	R/W
b17	AMS	Address Mode Select	0: Dual address mode 1: Single address mode	R/W
b31 to b18	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

Note 1. Offset addition can be specified only for EXDMAC0.

This register is used to set address mode of the EXDMACn.

DARA[4:0] Bits (Destination Address Extended Repeat Area Set)

These bits specify the extended repeat area on the destination address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 21 (2 bytes) and 217 (128 Mbytes).

When the lower address overflows the extended repeat area by address increment, the top address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the bottom address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer, block transfer, or cluster transfer is selected, or when EXDMACn.EDMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat area or block area), write 00000b in the DARA[4:0] bits.

An interrupt can be requested when an overflow occurs in the extended repeat area with the DARIE bit in EDMINT set to 1. Table 19.3 shows the settings and the corresponding extended repeat areas.

DM[1:0] Bits (Destination Address Update Mode Set)

These bits select the mode of updating the destination address.
When increment is selected and the SZ[1:0] bits in EDMTMD are set to 00b, 01b, and 10b, the destination address is incremented by 1, 2, and 4, respectively.
When decrement is selected and the SZ[1:0] bits in EDMTMD are set to 00b, 01b, and 10b, the destination address is decremented by 1, 2, and 4, respectively.
When offset addition is selected, the offset specified by the EXDMAC0.EDMOFR register is added to the address. Offset addition can be specified only for EXDMAC0.

SARA[4:0] Bits (Source Address Extended Repeat Area Set)

These bits specify the extended repeat area on the source address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 21 (2 bytes) and 217 (128 Mbytes).
When the lower address overflows the extended repeat area by address increment, the top address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the bottom address of the extended repeat area is set.
When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer, block transfer, or cluster transfer is selected, or when EXDMACn.EDMTMD.DTS[1:0] = 01b (the transfer destination is specified as the repeat area or block area), write 00000b in the SARA[4:0] bits.
An interrupt can be requested when an overflow occurs in the extended repeat area with the SARIE bit in EDMINT set to 1. Table 19.3 shows the settings and the corresponding extended repeat areas.

SM[1:0] Bit (Source Address Update Mode Set)

These bits select the mode of updating the source address.
When increment is selected and the SZ[1:0] bits in EDMTMD are set to 00b, 01b, and 10b, the destination address is incremented by 1, 2, and 4, respectively.
When decrement is selected and the SZ[1:0] bits in EDMTMD are set to 00b, 01b, and 10b, the destination address is decremented by 1, 2, and 4, respectively.
When offset addition is selected, the offset specified by the EXDMAC0.EDMOFR register is added to the address. Offset addition can be specified only for EXDMAC0.

DIR Bit (Single Address Direction Select)

This bit selects the transfer destination or source, to which the addresses should be output in single address mode.

- Normal/Repeat/Block Transfer

When this bit is set to 0, data is transferred in single address mode using the EDMSAR register value as the transfer source address. Here, EDACKn can be output to the transfer destination device by setting the DACKE bit in EDMOMD to 1 (EDACKn output is enabled).
When this bit is set to 1, data is transferred in single address mode using the EDMDAR register value as the transfer destination address. Here, EDACKn can be output to the transfer source device by setting the DACKE bit in EDMOMD to 1 (EDACKn output is enabled). Setting the DIR bit is valid when the AMS bit in EDMAMD is 1 (single address mode); setting the DIR bit is invalid when the AMS bit is 0 (dual address mode).

- Cluster Transfer

When this bit is set to 0, data is transferred in cluster transfer read address mode using the EDMSAR register value as the transfer source address. Here, data can be transferred to the cluster buffers from the external device.
When this bit is set to 1, data is transferred in cluster transfer write address mode using the EDMDAR register value as

the transfer destination address. Here, data can be transferred to the external device from the cluster buffers. Setting the DIR bit is valid when the AMS bit in EDMAMD is 1 (single address mode); setting the DIR bit is invalid when the AMS bit is 0 (dual address mode).

AMS Bit (Address Mode Select)

This bit selects the address mode.

- Normal/Repeat/Block Transfer

When this bit is set to 0, dual address mode is selected and when set to 1, single address mode is selected.
When using single address mode, select the transfer source or destination device to which the addresses should be output using the DIR bit in EDMAMD.

- Cluster Transfer

When this bit is set to 0, dual address mode is selected and when set to 1, read or write address mode is selected. Select read or write address mode using the DIR bit in EDMAMD.

Table 19.3 Settings and Range of Extended Repeat Areas

SARA[4:0] or DARA[4:0]	Extended Repeat Area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 Kbyte specified as extended repeat area by the lower 10 bits of the address
01011b	2 Kbytes specified as extended repeat area by the lower 11 bits of the address
01100b	4 Kbytes specified as extended repeat area by the lower 12 bits of the address
01101b	8 Kbytes specified as extended repeat area by the lower 13 bits of the address
01110b	16 Kbytes specified as extended repeat area by the lower 14 bits of the address
01111b	32 Kbytes specified as extended repeat area by the lower 15 bits of the address
10000b	64 Kbytes specified as extended repeat area by the lower 16 bits of the address
10001b	128 Kbytes specified as extended repeat area by the lower 17 bits of the address
10010b	256 Kbytes specified as extended repeat area by the lower 18 bits of the address
10011b	512 Kbytes specified as extended repeat area by the lower 19 bits of the address
10100b	1 Mbyte specified as extended repeat area by the lower 20 bits of the address
10101b	2 Mbytes specified as extended repeat area by the lower 21 bits of the address
10110b	4 Mbytes specified as extended repeat area by the lower 22 bits of the address
10111b	8 Mbytes specified as extended repeat area by the lower 23 bits of the address
11000b	16 Mbytes specified as extended repeat area by the lower 24 bits of the address
11001b	32 Mbytes specified as extended repeat area by the lower 25 bits of the address
11010b	64 Mbytes specified as extended repeat area by the lower 26 bits of the address
11011b	128 Mbytes specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	Setting prohibited

19.2.9 EXDMA Offset Register (EDMOFR)

Address(es): EXDMAC0.EDMOFR 0008 2818h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination.	0000 0000h to 00FF FFFFh (0 bytes to (16M – 1) bytes) FF00 0000h to FFFF FFFFh (–16M bytes to –1 byte)	R/W

This register is used to set the address offset value.
Write to this register while the EXDMAC operation is stopped or DMA transfer is disabled (not during data transfer).
Setting bits 31 to 25 is invalid; a value of bit 24 is extended to bits 31 to 25. Reading EDMOFR returns the extended value.

19.2.10 EXDMA Transfer Enable Register (EDMCNT)

Address(es): EXDMAC0.EDMCNT 0008 281Ch, EXDMAC1.EDMCNT 0008 285Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DTE	DMA Transfer Enable	0: Disables DMA transfer. 1: Enables DMA transfer.	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

This register enables or disables the DMA transfer to the corresponding channel.

DTE Bit (DMA Transfer Enable)

When the EDMAST.DMST bit is set to 1 (EXDMAC module start) and this bit is set to 1 (DMA transfer is enabled), DMA transfer can be started for the corresponding channel.

When the DTE bit is set to 0 during DMA transfer, DMA transfer is suspended after completion of the current data transfer corresponding to a single transfer request. DMA transfer is resumed by setting the DTE bit to 1 again.

While the DTE bit is 1, writing to registers other than the DTE bit of the EXDMAC channel is prohibited.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When 0 is written to this bit.
- When the specified total volume of data transfer is completed.
- When DMA transfer is stopped by the repeat size end interrupt.
- When DMA transfer is stopped by the extended repeat area overflow interrupt.

19.2.11 EXDMA Software Start Register (EDMREQ)

Address(es): EXDMAC0.EDMREQ 0008 281Dh, EXDMAC1.EDMREQ 0008 285Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CLRS	—	—	—	SWREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SWREQ	DMA Software Start	0: DMA transfer is not requested. 1: DMA transfer is requested.	R/W
b3 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W
b4	CLRS	DMA Software Start Bit Auto Clear Select	0: SWREQ bit is cleared after DMA transfer is started by software. 1: SWREQ bit is not cleared after DMA transfer is started by software.	R/W
b7 to b5	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

This register is used to execute the DMA by software.

SWREQ Bit (DMA Software Start)

When 1 is written to this bit, a DMA transfer request is generated. After DMA transfer is started in response to the request, this bit is set to 0 if the CLRS bit is set to 0. This bit is not set to 0 while the CLRS bit is set to 1. In this case, a DMA transfer request can be issued again after completion of a transfer.
Note that, however, setting this bit is valid and DMA transfer by software is enabled only when the DCTG[1:0] bits in EDMTMD are set to 00b (DMA request source is software).
Setting this bit is invalid when the DCTG[1:0] bits in EDMTMD are set to a value other than 00b.
To start DMA transfer by software with CLRS set to 0, check that the SWREQ bit is 0 and then write 1 to SWREQ.
[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When a DMA transfer request by software is accepted and DMA transfer is started while the CLRS bit is set to 0 (the SWREQ bit is cleared after DMA transfer is started by software).
- When 0 is written to this bit.

CLRS Bit (DMA Software Start Bit Auto Clear Select)

This bit specifies whether to clear the SWREQ bit to 0 after DMA transfer is started in response to the DMA transfer request generated by setting the SWREQ bit to 1. With this bit set to 0, the SWREQ bit is set to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not set to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

19.2.12 EXDMA Status Register (EDMSTS)

Address(es): EXDMAC0.EDMSTS 0008 281Eh, EXDMAC1.EDMSTS 0008 285Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	ACT	—	—	DTIF	—	—	—	ESIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESIF	Transfer Escape End Interrupt Flag	0: A transfer escape end interrupt has not been generated. 1: A transfer escape end interrupt has been generated.	R/W
b3 to b1	—	Reserved	These bits are always read as 0. Writing to these bits has no effect.	R
b4	DTIF	Transfer End Interrupt Flag	0: A transfer end interrupt has not been generated. 1: A transfer end interrupt has been generated.	R/W
b6, b5	—	Reserved	These bits are always read as 0. Writing to these bits has no effect.	R
b7	ACT	EXDMA Active Flag	0: EXDMAC operation is suspended. 1: EXDMAC is operating.	R

This register indicates the state of the DMA.

ESIF Flag (Transfer Escape End Interrupt Flag)

This flag indicates that the transfer escape end interrupt has been generated.

[Setting conditions]

- When 1-repeat size data transfer is completed in repeat transfer mode with the EDMINT.RPTIE bit set to 1.
- When 1-block data transfer is completed in block transfer mode with the EDMINT.RPTIE bit set to 1.
- When 1-cluster data transfer is completed in cluster transfer mode with the EDMINT.RPTIE bit set to 1.
- When an extended repeat area overflow on the source address occurs while the EDMINT.SARIE bit is set to 1 and the EDMAMD.SARA[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer source address)
- When an extended repeat area overflow on the destination address occurs while the EDMINT.DARIE bit is set to 1 and the EDMAMD.DADR[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer destination address)

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the EDMCNT.DTE bit.

DTIF Flag (Transfer End Interrupt Flag)

This flag indicates that the transfer end interrupt has been generated.

[Setting conditions]

- When the specified number of unit-transfers are completed in normal transfer mode (the value of EDMCRAL becoming 0 on completion of transfer)
- When the specified number of repeat transfer operations are completed in repeat transfer mode (the value of EDMCRB becoming 0 on completion of transfer)
- When the specified number of blocks have been transferred in block transfer mode (the value of EDMCRB becoming 0 on completion of transfer)
- When the specified number of clusters have been transferred in cluster transfer mode (the value of EDMCRB becoming 0 on completion of transfer)

[Clearing conditions]

- When 0 is written to this bit
- When 1 is written to the DTE bit in EDMCNT

ACT Flag (EXDMA Active Flag)

- This flag indicates whether the EXDMACn is in the idle or active state.

[Setting condition]

- When the EXDMACn starts data transfer operation

[Clearing condition]

- When data transfer in response to one transfer request is completed

19.2.13 EXDMA External Request Sense Mode Register (EDMRMD)

Address(es): EXDMAC0.EDMRMD 0008 2820h, EXDMAC1.EDMRMD 0008 2860h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DREQS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DREQS[1:0]	Request Input Sense Mode Set	b1 b0 0 0: Rising edge 0 1: Falling edge 1 0: Low level 1 1: (Setting prohibited)	R/W
b7 to b2	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

This register is used to set sense mode of the EDREQn pin.

DREQS[1:0] Bits (Request Input Sense Mode Set)

These bits specify the sense mode for the external DMA transfer request signal (EDREQn pin).

19.2.14 EXDMA External Request Flag Register (EDMERF)

Address(es): EXDMAC0.EDMERF 0008 2821h, EXDMAC1.EDMERF 0008 2861h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	EREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EREQ	Peripheral Module Request Flag	0: No request 1: Requested	R/(W) *1
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

Note 1. Writing 0 has no effect.

This register detects a request generated by the EDREQn pin.

EREQ Flag (Peripheral Module Request Flag)

This flag indicates the DMA transfer request from the external DMA transfer request signal (the EDREQn pin).

[Setting conditions]

- When the level on the EDREQn pin changes from 0 to 1 while EXDMACn.EDMRMD.DREQS[1:0] = 00b
- When the level on the EDREQn pin changes from 1 to 0 while EXDMACn.EDMRMD.DREQS[1:0] = 01b
- When the level on the EDREQn pin is 0 while EXDMACn.EDMRMD.DREQS[1:0] = 10b (low level)

[Clearing conditions]

- When the DMA transfer is started while EXDMACn.EDMRMD.DREQS[1:0] = 00b (rising edge) or 01b (falling edge) and then the DMA transfer is started
- When 1 is written to this flag while EXDMACn.EDMRMD.DREQS[1:0] = 00b (rising edge) or 01b (falling edge)
- When the EDREQn pin is set to 1 while EXDMACn.EDMRMD.DREQS[1:0] = 10b (low level)

19.2.15 EXDMA Peripheral Request Flag Register (EDMPRF)

Address(es): EXDMAC0.EDMPRF 0008 2822h, EXDMAC1.EDMPRF 0008 2862h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PREQ	Peripheral Module Request Flag	0: No request 1: Requested	R/(W) *1
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

Note 1. Writing 0 has no effect.

This register detects a DMA transfer request generated by a peripheral module.

PREQ Flag (Peripheral Module Request Flag)

This flag detects the DMA transfer request from the peripheral modules.

[Setting condition]

- When the DMA transfer request is generated from the peripheral modules

[Clearing conditions]

- When the DMA transfer request is generated from the peripheral modules and the DMA transfer is started
- When 1 is written to this flag

19.2.16 EXDMAC Module Start Register (EDMAST)

Address(es): 0008 2A00h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DMST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DMST	EXDMAC Module Start	0: EXDMAC module stop 1: EXDMAC module start	R/W
b7 to b1	—	Reserved	These bits are always read as 0. The write value should be 0.	R/W

This register enables or disables the startup of all channels of the EXDMAC.

DMST Bit (EXDMAC Module Start)

When this bit is set to 1, EXDMAC is ready to accept transfer requests for all the channels.
When 1 is written to the DTE bit in EDMCNT (DMA transfer is enabled) of all the EXDMACn channels and then this bit is set to 1, all the channels can be placed in the transfer request acceptable state at the same time.
When the DMST bit is set to 0 during DMA transfer, DMA transfer for all channels is suspended after completion of the current data transfer corresponding to a single transfer request. DMA transfer is resumed by setting the DMST bit to 1 again.
[Setting condition]

- When 1 is written to this bit

[Clearing condition]

- When 0 is written to this bit

19.2.17 Cluster Buffer Register y (CLSBRy) (y = 0 to 7)

Address(es): CLSBR0 0008 2BE0h, CLSBR1 0008 2BE4h, CLSBR2 0008 2BE8h, CLSBR3 0008 2BECh, CLSBR4 0008 2BF0h, CLSBR5 0008 2BF4h, CLSBR6 0008 2BF8h, CLSBR7 0008 2BFCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	R/W
b31 to b0	Buffer area for cluster transfer.	R/W

CLSBRy are buffer registers for cluster transfer.
During cluster transfer, transferred data is sequentially stored in CLSBRy starting from CLSBR0. The cluster-transferred data or data written by the CPU is retained until another cluster transfer or data write by the CPU. When reading the cluster-transferred data with the CPU, confirm that cluster transfer has been completed and only refer to the data of the specified size for cluster; the other data is invalid.
During cluster transfer, the same CLSBRy is used for all the channels. If a conflict occurs between the write to CLSBRy by the CPU and cluster transfer, transferred data is not guaranteed. If a channel is set to cluster transfer in read or write address mode and another channel is set to cluster transfer, data to be transferred may be erroneously modified.
Data is stored in cluster buffers in the different manner depending on the transfer size setting (EDMTMD.SZ[1:0] bits).

(1) Transfer Size is 8 Bits (EXDMACn.EDMTMD.SZ[1:0] = 00b)

Data is stored in the lower 8 bits in the cluster buffers. Here, the upper 24 bits are invalid. When the maximum cluster size is set to 8, 8-byte data is one cluster.
Data is stored in CLSBR in the order of CLSBR0 to CLSBRj (j = cluster size value – 1).

(2) Transfer Size is 16 Bits (EXDMACn.EDMTMD.SZ[1:0] = 01b)

Data is stored in the lower 16 bits in the cluster buffers. Here, the upper 16 bits are invalid. When the maximum cluster size is set to 8, 16-byte data is one cluster.
Data is stored in CLSBR in the order of CLSBR0 to CLSBRj (j = cluster size value – 1).

(3) Transfer Size is 32 Bits (EXDMACn.EDMTMD.SZ[1:0] = 10b)

Data is stored in all the 32 bits in the cluster buffers. When the maximum cluster size is set to 8, 32-byte data is one cluster.
Data is stored in CLSBR in the order of CLSBR0 to CLSBRj (j = cluster size value – 1).

19.3 Operation

19.3.1 Transfer Mode

(1) Normal Transfer Mode

In normal transfer mode, one data is transferred by one transfer request. A maximum of 65535 can be set as the number of transfer operations using EDMCRA of EXDMACn. When the EXDMACn.EDMCRAL bits are set to 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode). Setting EDMCRB of EXDMACn is invalid in normal transfer mode. Except in free running mode, a transfer end interrupt request can be generated after completion of the specified number of transfer operations.
Table 19.4 summarizes the register update operation in normal transfer mode, and Figure 19.2 shows the operation in normal transfer mode.

Table 19.4 Register Update Operation in Normal Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request
EXDMACn.EDMSAR	Transfer source address	Increment/decrement/fixed/offset addition*1
EXDMACn.EDMDAR	Transfer destination address	Increment/decrement/fixed/offset addition*1
EXDMACn.EDMCRAL	Transfer counter	Decremented by one/not updated (in free running mode)
EXDMACn.EDMCRAH	—	Not updated (Not used in normal transfer mode)
EXDMACn.EDMCRB	—	Not updated (Not used in normal transfer mode)

Note 1. Offset addition can be specified only for EXDMAC0.

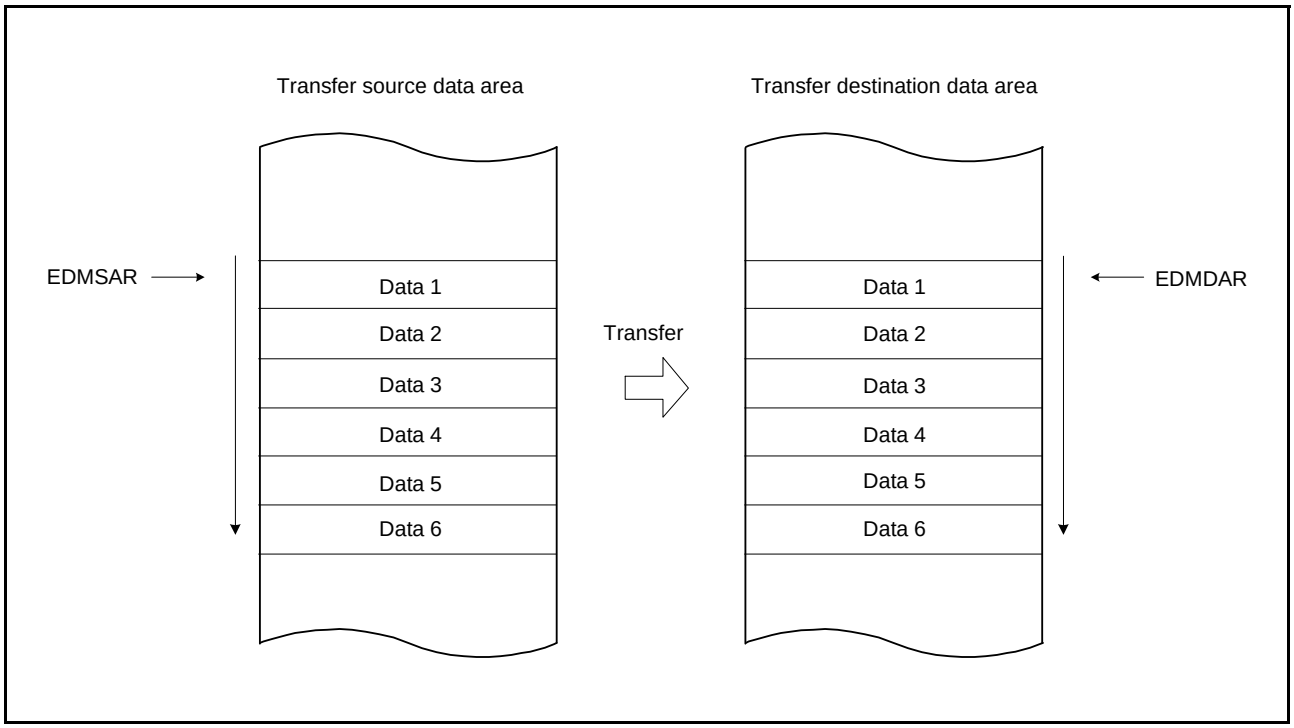


Figure 19.2 Operation in Normal Transfer Mode

(2) Repeat Transfer Mode

In repeat transfer mode, one data is transferred by one transfer request.
A maximum of 1K data can be set as a total repeat transfer size using EDMCRA of the EXDMACn.
A maximum of 1K can be set as the number of repeat transfer operations using EDMCRB of the EXDMACn; therefore, a maximum of 1K data × 1K = 1M can be set as a total data transfer size.
Either the transfer source or transfer destination can be specified as a repeat area. When transfer of the repeat size data is completed, the address of the specified repeat area (EDMSAR or EDMDAR of the EXDMACn) returns to the transfer start address. When data of the specified repeat size has all been transferred in repeat transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DTE bit in EDMCNT of EXDMACn in the repeat size end interrupt handling.
A transfer end interrupt request can be generated after completion of the specified number of repeat transfer operations.
Table 19.5 summarizes the register update operation in repeat transfer mode, and Figure 19.3 shows the operation in repeat transfer mode.

Table 19.5 Register Update Operation in Repeat Transfer Mode

		Update Operation after Completion of a Transfer by One Transfer Request	
Register	**Function**	**When EXDMACn.EDMCRAL is not 1**	**When EXDMACn.EDMCRAL is 1 (Transfer of the Last Data in Repeat Size)**
EXDMACn.EDMSAR	Transfer source address	Increment/decrement/fixed/ offset addition*1	• EXDMACn.EDMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition*1 • EXDMACn.EDMTMD.DTS[1:0] = 01b Initial value of EXDMACn.EDMSAR • EXDMACn.EDMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
EXDMACn.EDMDAR	Transfer destination address	Increment/decrement/fixed/ offset addition*1	• EXDMACn.EDMTMD.DTS[1:0] = 00b Initial value of EXDMACn.EDMDAR • EXDMACn.EDMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition*1 • EXDMACn.EDMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
EXDMACn.EDMCRAH	Repeat size	Not updated	Not updated
EXDMACn.EDMCRAL	Transfer counter	Decremented by one	EXDMACn.EDMCRAH
EXDMACn.EDMCRB	Repeat-count counter	Not updated	Decremented by one

Note 1. Offset addition can be specified only for EXDMAC0.

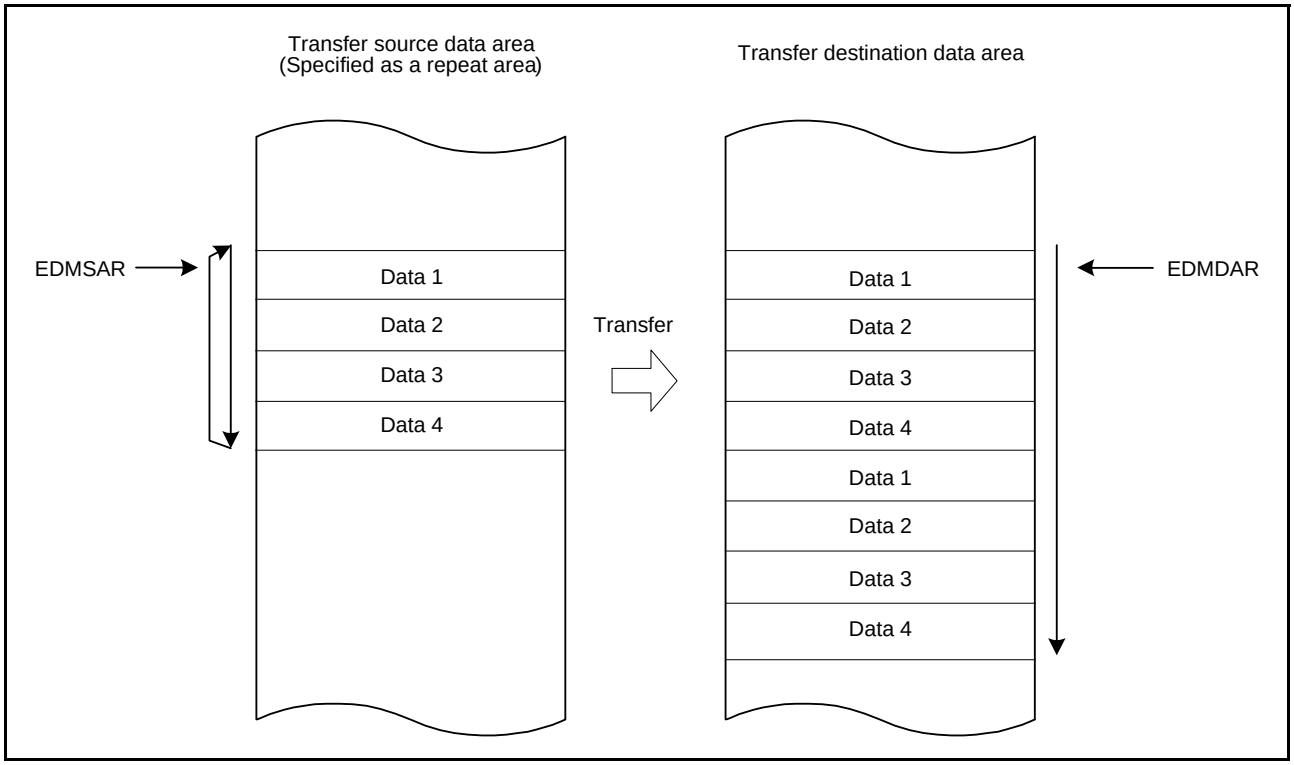


Figure 19.3 Operation in Repeat Transfer Mode

(3) Block Transfer Mode

In block transfer mode, a single block data is transferred by one transfer request.

A maximum of 1K data can be set as a total block transfer size using EDMCRA of the EXDMACn.

A maximum of 1K can be set as the number of block transfer operations using EDMCRB of the EXDMACn; therefore, a maximum of 1K data × 1K blocks = 1M can be set as a total data transfer size.

Either the transfer source or transfer destination can be specified as a block area. When transfer of a single block data is completed, the address of the specified block area (EDMSAR or EDMDAR of the EXDMACn) returns to the transfer start address. When a single block data has all been transferred in block transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DTE bit in EDMCNT of EXDMACn in the repeat size end interrupt handling.

Transfer end interrupt request can be generated after completion of the specified number of block transfer operations.

Table 19.6 summarizes the register update operation in block transfer mode, and Figure 19.4 shows the operation in block transfer mode.

Table 19.6 Register Update Operation in Block Transfer Mode

Register	Function	Update Operation after Completion of Single-Block Transfer by One Transfer Request
EXDMACn.EDMSAR	Transfer source address	• EXDMACn.EDMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition*1 • EXDMACn.EDMTMD.DTS[1:0] = 01b Initial value of EXDMACn.EDMSAR • EXDMACn.EDMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
EXDMACn.EDMDAR	Transfer destination address	• EXDMACn.EDMTMD.DTS[1:0] = 00b Initial value of EXDMACn.EDMDAR • EXDMACn.EDMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition*1 • EXDMACn.EDMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
EXDMACn.EDMCRAH	Block size	Not updated
EXDMACn.EDMCRAL	Block size counter	EXDMACn.EDMCRAH
EXDMACn.EDMCRB	Transfer-block counter	Decremented by one

Note 1. Offset addition can be specified only for EXDMAC0.

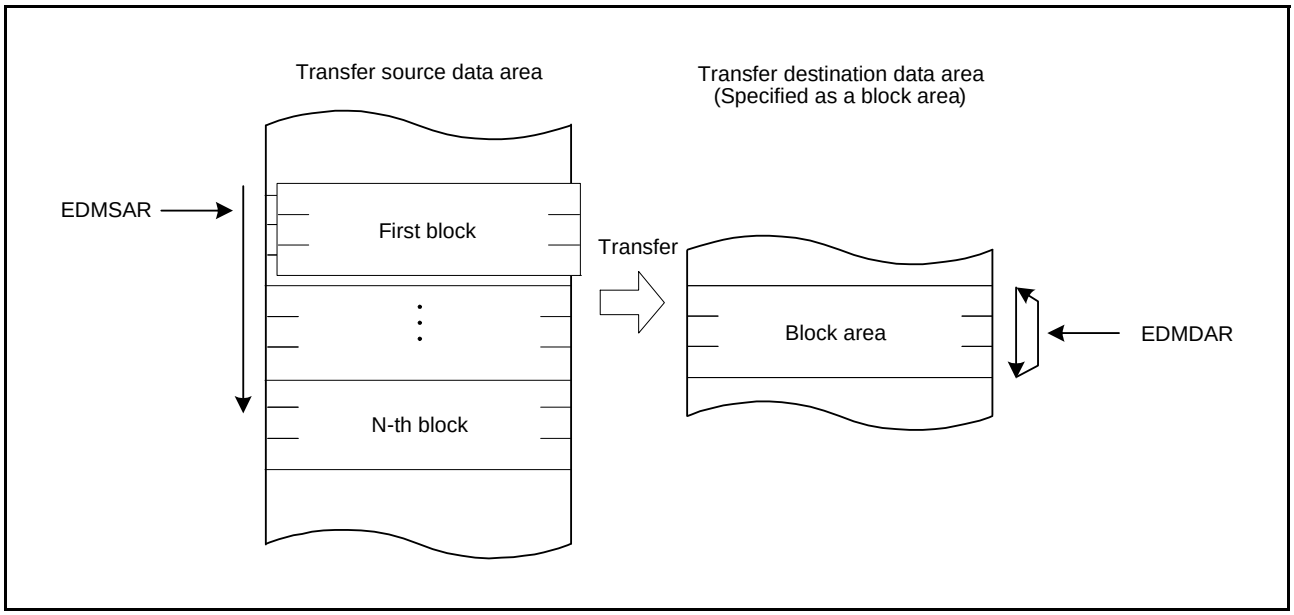


Figure 19.4 Operation in Block Transfer Mode

(4) Cluster Transfer Mode

In cluster transfer mode, a single cluster data is transferred by one transfer request. A maximum of 8 data can be set as a total cluster transfer size using EDMCRA of the EXDMACn.
A maximum of 1K can be set as the number of cluster transfer operations using EDMCRB of the EXDMACn; therefore, a maximum of 8 data × 1K = 8K can be set as a total data transfer size.
The cluster transfer mode can be selected from among cluster transfer dual address mode, cluster transfer read address mode, and cluster transfer write address mode.

- Cluster transfer dual address mode
  (EXDMACn.EDMTMD.MD[1:0] = 11b, EXDMACn.EDMAMD.AMS = 0)
  A single cluster data is transferred by one transfer request from the transfer source address to the cluster buffers. A single cluster data is then transferred from the cluster buffers to the transfer destination address.
- Cluster transfer read address mode
  (EXDMACn.EDMTMD.MD[1:0] = 11b, EXDMACn.EDMAMD.AMS = 1, EXDMACn.EDMAMD.DIR = 0)
  A single cluster data is transferred by one transfer request from the transfer source address to the cluster buffers.
- Cluster transfer write address mode
  (EXDMACn.EDMTMD.MD[1:0] = 11b, EXDMACn.EDMAMD.AMS = 1, EXDMACn.EDMAMD.DIR = 1)
  A single cluster data is transferred by one transfer request from the cluster buffers to the transfer destination address.

In cluster-transfer mode, DMA transfer stops on completion of the transfer of each cluster of data, and a repeat-size-completed interrupt request can be generated. DMA transfer can be restarted by writing 1 to the EXDMACn.EDMCNT.DTE bit during processing of the repeat-size-completed interrupt.
A repeat-size-completed interrupt request can also be generated on completion of transfer of clusters the specified number of times.

Table 19.7 summarizes the register update operation in cluster transfer mode, and Figure 19.5 shows the operation in cluster transfer mode.

Table 19.7 Register Update Operation in Cluster Transfer Mode (Dual Address Mode)

Register	Function	Update Operation after Completion of Single-Cluster Transfer by One Transfer Request
EXDMACn.EDMSAR	Transfer source address	• EXDMACn.EDMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition*1 • EXDMACn.EDMTMD.DTS[1:0] = 01b Initial value of EXDMACn.EDMSAR • EXDMACn.EDMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
EXDMACn.EDMDAR	Transfer destination address	• EXDMACn.EDMTMD.DTS[1:0] = 00b Initial value of EXDMACn.EDMDAR • EXDMACn.EDMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition*1 • EXDMACn.EDMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
EXDMACn.EDMCRAH	Cluster size	Not updated
EXDMACn.EDMCRAL	Cluster size counter	EXDMACn.EDMCRAH
EXDMACn.EDMCRB	Transfer cluster count counter	Decremented by one

Note 1. Offset addition can be specified only for EXDMAC0.
In read address mode, the transfer destination address EXDMACn.EDMDAR is fixed (invalid).
In write address mode, the transfer destination address EXDMACn.EDMSAR is fixed (invalid).

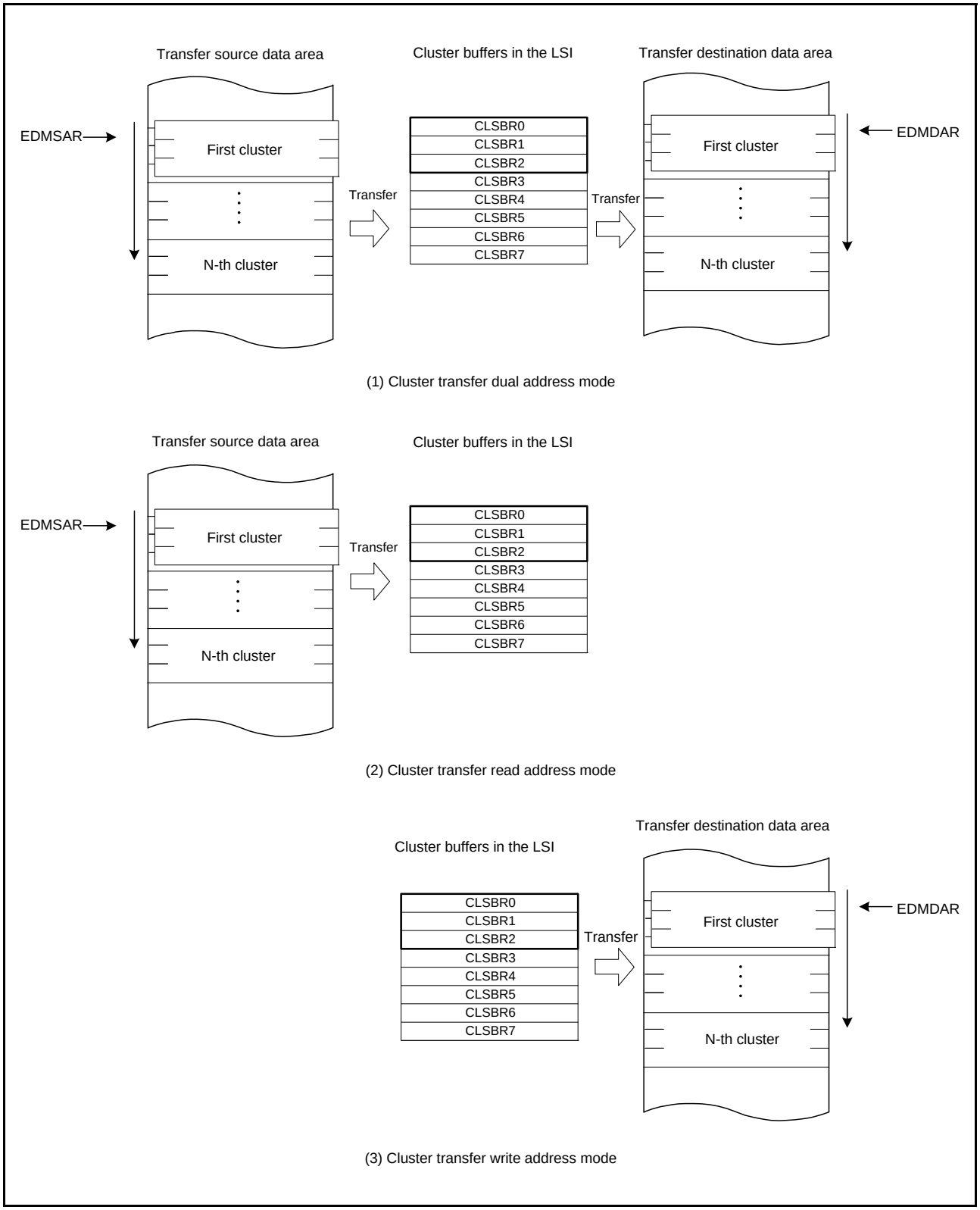


Figure 19.5 Operation in Cluster Transfer Mode

19.3.2 Extended Repeat Area Function

The EXDMAC supports a function to specify the extended repeat areas on the transfer source and destination addresses. With the extended repeat areas set, the address registers repeatedly indicate the addresses of the specified extended repeat areas.

The extended repeat areas can be specified separately to the transfer source address register (EDMSAR) and transfer destination address register (EDMDAR) of EXDMACn.

The extended repeat area on the source address is specified by the SARA[4:0] bits in EDMAMD of EXDMACn. The extended repeat area on the destination address is specified by the DARA[4:0] bits in EDMAMD of EXDMACn. The size can be specified separately for the source and destination sides. However, do not specify a repeat area or block area that is also an extended repeat area.

When the address register value reaches the end address of the extended repeat area and the extended repeat area overflows, DMA transfer is stopped and an interrupt by an extended repeat area overflow can be requested. When an overflow occurs in the extended repeat area on the transfer source while the SARIE bit in EDMINT of EXDMACn is set to 1, the ESIF flag in EDMSTS of EXDMACn is set to 1 and the DTE bit in EDMINT of EXDMACn is set to 0 to stop DMA transfer. At this time, if the ESIE bit in EDMINT of EXDMACn is set to 1, an interrupt by an extended repeat area overflow is requested.

When the DARIE bit in EDMINT of EXDMACn is set to 1, an overflow on the extended repeat area set in EDMDAR occurs, meaning that the destination side is a target. DMA transfer can be resumed by writing 1 to the DTE bit in EDMCNT of EXDMACn in the extended repeat area overflow interrupt handling.

Figure 19.6 shows an example of the extended repeat area operation.

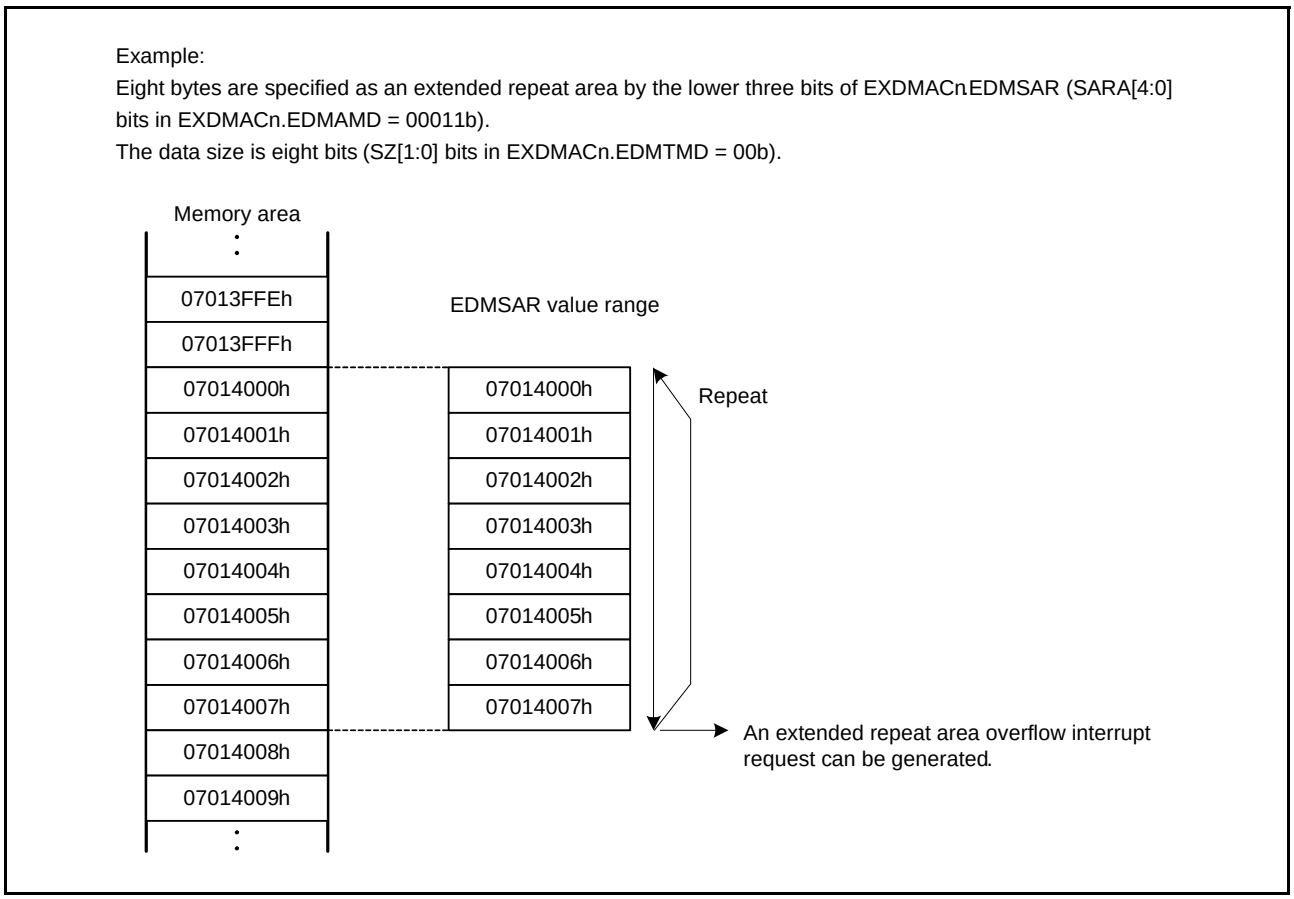


Figure 19.6 Example of Extended Repeat Area Operation

When an interrupt by an extended repeat area overflow is used in block transfer mode or cluster transfer mode, the following should be taken into consideration.
When a transfer is stopped by an interrupt by an extended repeat area overflow, the address register must be set so that the block size (or cluster size) is a power of 2 or the block size (or cluster size) boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block (or one cluster), the interrupt by the overflow is suspended until transfer of the block (or the cluster) is completed, and the transfer overruns.

Figure 19.7 shows an example when the extended repeat area function is used in block transfer mode.

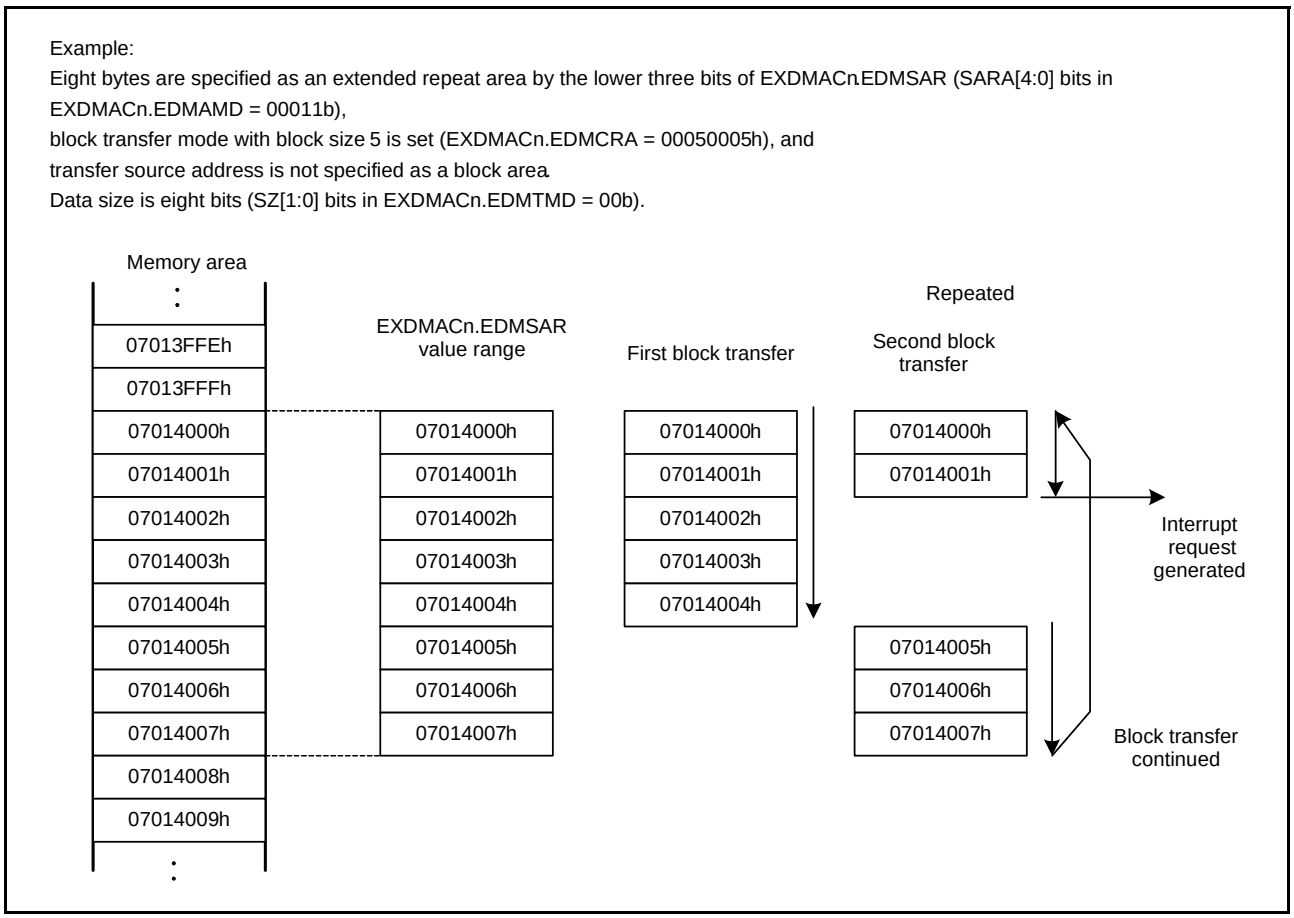


Figure 19.7 Example of Extended Repeat Area Function in Block Transfer Mode

19.3.3 Address Update Function using Offset

The source and destination addresses can be updated by fixing, increment, decrement, or offset addition. When the offset addition is selected, the offset specified by the EXDMA offset register (EDMOFR of EXDMAC0) is added to the address every time the EXDMAC performs one data transfer. This function realizes a data transfer where addresses are allocated to separated areas.

Offset subtraction can also be realized by setting a negative value in EDMOFR of EXDMAC0. In this case, the negative value must be 2's complement.

Address update function using offset can be specified only for the EXDMAC0 channel.

Table 19.8 shows the address update method in each address update mode.

Table 19.8 Address Update Method in Each Address Update Mode

Address Update Mode	Settings of EXDMACn.EDMAMD.SM[1:0] and EXDMACn.EDMAMD.DM[1:0] for Address Update Modes	Address Update Method (for Different SZ[1:0] Settings in EDMTMD of EXDMACn)		
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b
Address fixed	00b	Fixed		
Offset addition	01b	+EXDMAC0.EDMOFR*1		
Increment	10b	+1	+2	+4
Decrement	11b	–1	–2	–4

Note 1. When setting a negative value in the DMA offset register, the value must be 2’s complement. The 2's complement is obtained by the following formula.
2’s complement of a negative offset value = ~ (offset) + 1 (~: bit inversion)

(1) Basic Transfer Using Offset Addition

Figure 19.8 shows an example of address updating using offset addition.

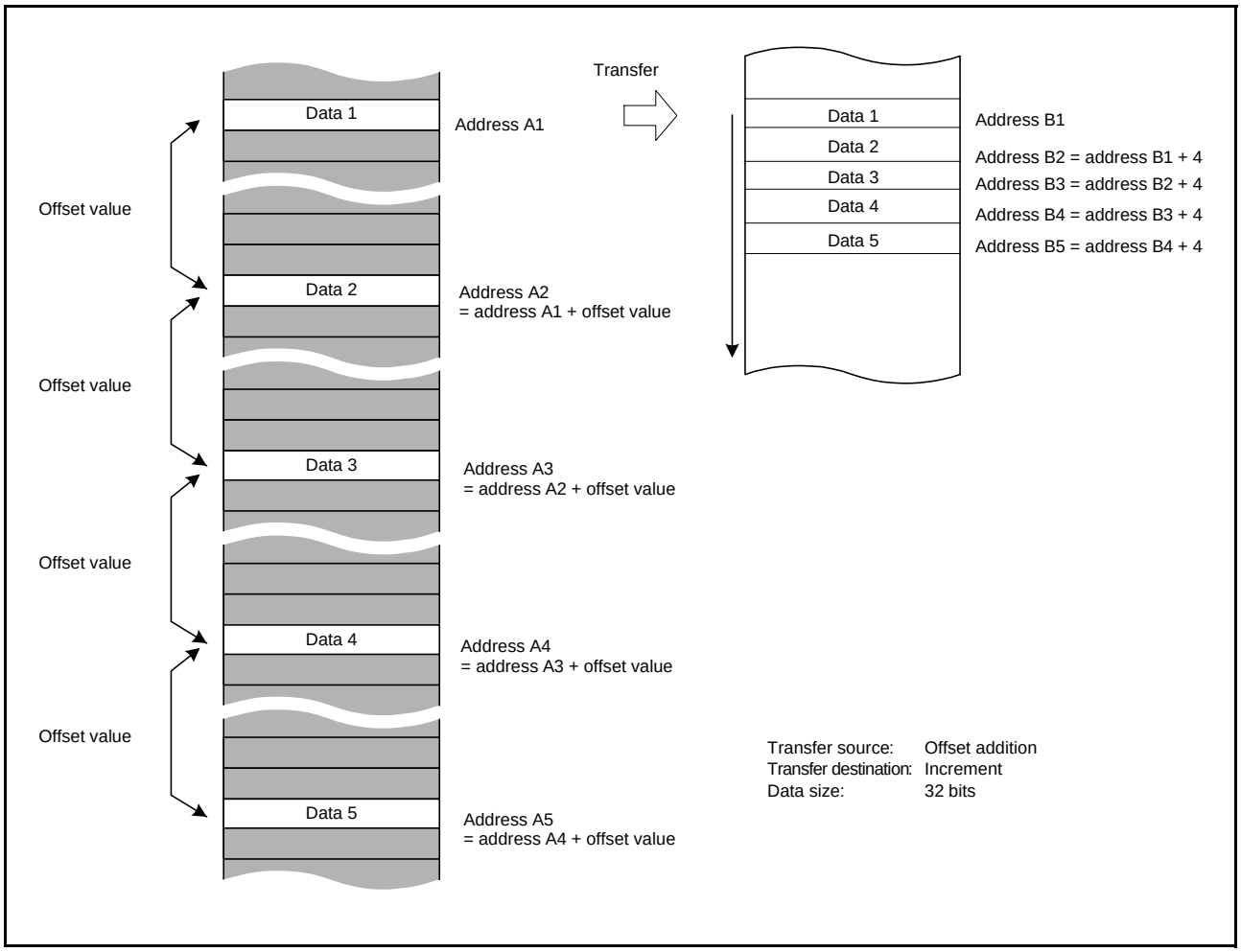


Figure 19.8 Example of Address Updating by Offset Addition

In Figure 19.8, the transfer data is 32 bits long, and offset addition and increment are set as the transfer source address update mode and transfer destination address update mode, respectively. The second and subsequent data is each read from the transfer source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to the continuous locations on the destination.

(2) Example of XY Conversion Using Offset Addition

Figure 19.9 shows the XY conversion using offset addition in repeat transfer mode.
The settings are as follows.

- EXDMAC0.EDMAMD register: Source address update mode (offset addition)
- EXDMAC0.EDMAMD register: Source address update mode (incremented)
- EXDMAC0.EDMTMD register: Transfer data size select (32-bit transfer)
- EXDMAC0.EDMTMD register: Transfer mode select (repeat transfer)
- EXDMAC0.EDMTMD register: Repeat area select (the source is specified as the repeat area)
- EXDMAC0.EDMOFR register: Address offset setting (10h)
- EXDMAC0.EDMCRA register: The number of repeat transfer setting (4h)
- EXDMAC0.EDMINT register: Repeat size end interrupt enable

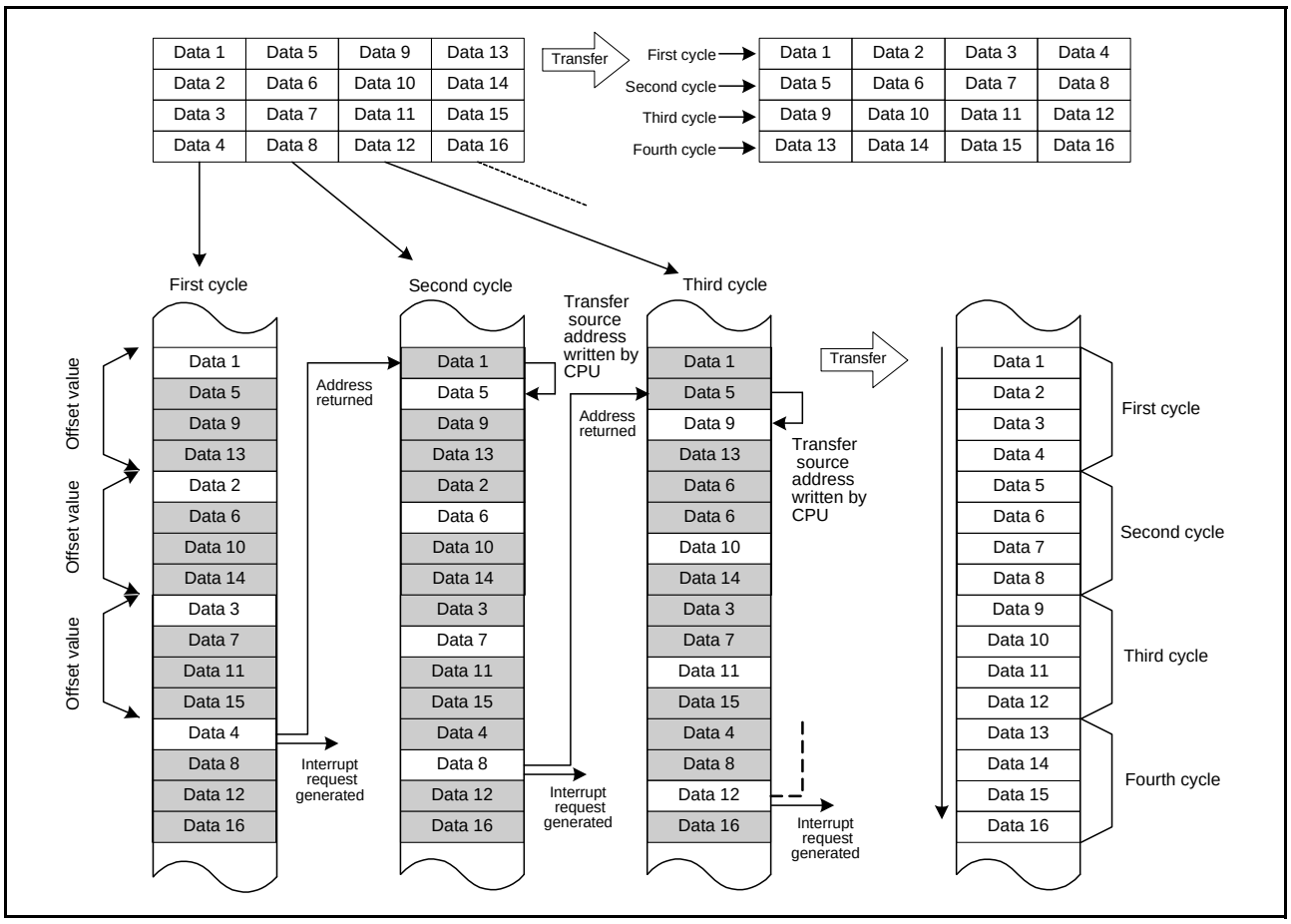


Figure 19.9 XY Conversion Operation Using Offset Addition in Repeat Transfer Mode

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to the destination continuous addresses. When data 4 is transferred, which means that the repeat size of transfers is completed, EXDMAC returns the transfer source address to the transfer start address (address of data 1 on the transfer source) and a repeat size end interrupt is requested. While this interrupt stops the transfer temporarily, perform the following.

- EXDMAC0.EDMSAR: Rewrite the DMA transfer source address to the address of data 5 (with the above example, the data 1 address + 4).
- EXDMAC0. EDMCNT: Set the DTE bit to 1.

The DMA transfer is resumed from the state when the DMA transfer is stopped. After that, the operations described above are repeated until the transfer source data is transposed to the destination area (XY conversion)

Figure 19.10 shows a flowchart of the XY conversion.

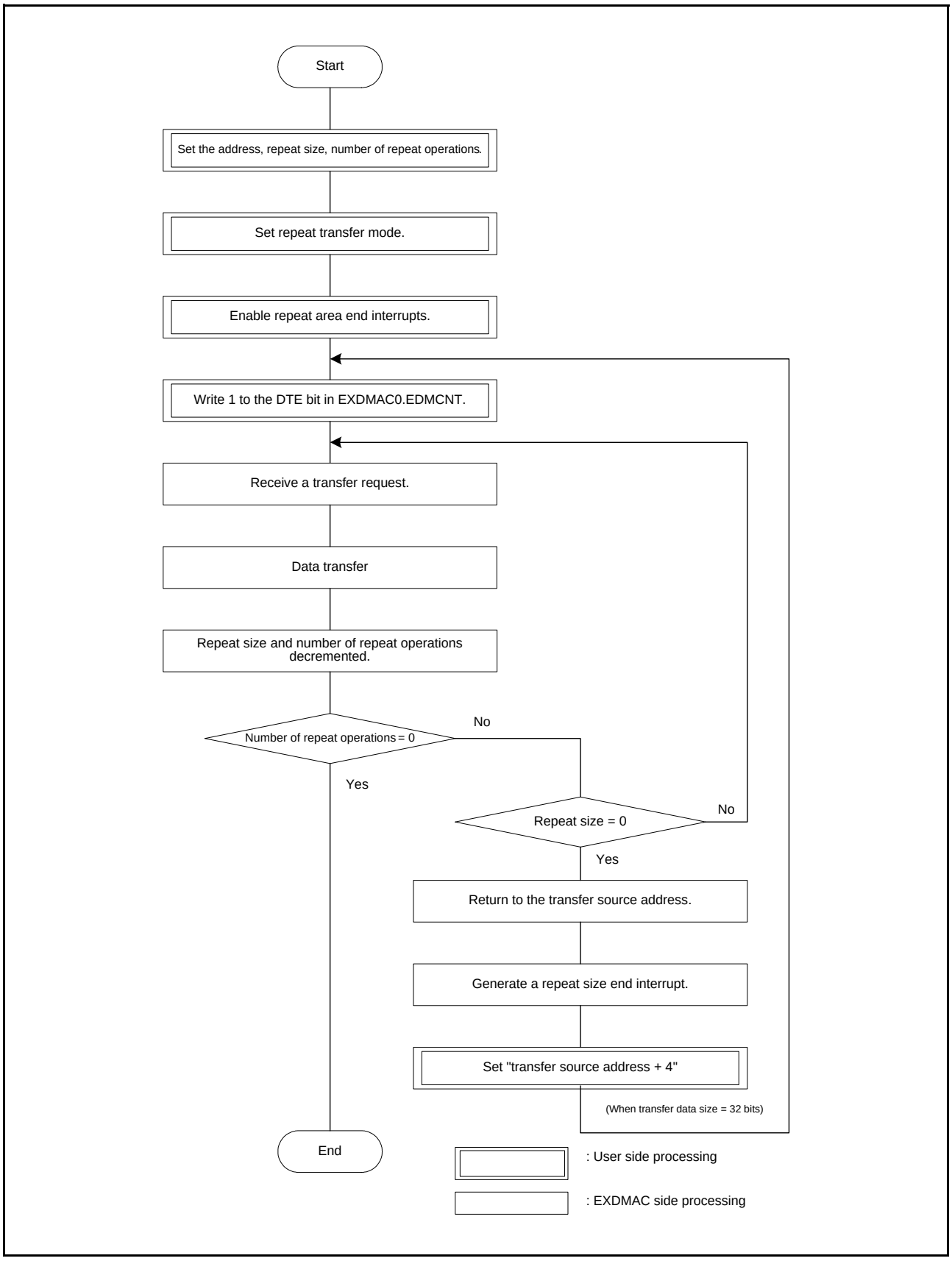


Figure 19.10 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode

19.3.4 Address Modes

The EXDMAC provides dual and single address modes (dual, read, and write address modes in cluster transfer), either of which can be selectable. Table 19.9 shows the relationship between transfer modes and address modes.

Table 19.9 Relationship between Transfer Modes and Address Modes

Transfer Mode	Address Mode	Single Address Direction	EXDMAC Operation
Normal transfer mode (EDMTMD.MD[1:0] = 00b)	Dual address mode (EDMAMD.AMS = 0)	—	Reads and then writes data.
	Single address mode (EDMAMD.AMS = 1)	Transfer source (EDMAMD.DIR = 0)	Only reads data; outputs EDACKn to the device to be written to. This MCU receives no read data.
		Transfer destination (EDMAMD.DIR = 1)	Only writes data; outputs EDACKn to the device to be read from. This MCU outputs no write data.
Repeat transfer mode (EDMTMD.MD[1:0] = 01b)	Dual address mode (EDMAMD.AMS = 0)	—	Reads and then writes data.
	Single address mode (EDMAMD.AMS = 1)	Transfer source (EDMAMD.DIR = 0)	Only reads data; outputs EDACKn to the device to be written to. This MCU receives no read data.
		Transfer destination (EDMAMD.DIR = 1)	Only writes data; outputs EDACKn to the device to be read from. This MCU outputs no write data.
Block transfer mode (EDMTMD.MD[1:0] = 10b)	Dual address mode (EDMAMD.AMS = 0)	—	Reads and writes data alternately for every data transfer specified by EDMTMD.SZ[1:0] (transfer data size)
	Single address mode (EDMAMD.AMS = 1)	Transfer source (EDMAMD.DIR = 0)	Only reads data; outputs EDACKn to the device to be written to. This MCU receives no read data.
		Transfer destination (EDMAMD.DIR = 1)	Only writes data; outputs EDACKn to the device to be read from. This MCU outputs no write data.
Cluster transfer mode (EDMTMD.MD[1:0] = 11b)	Dual address mode (EDMAMD.AMS = 0)	—	Reads data of cluster size and then writes data of cluster size.
	Read address mode (EDMAMD.AMS = 1)	Transfer source (EDMAMD.DIR = 0)	Only reads data of cluster size. Transfers data to the cluster buffers.
	Write address mode (EDMAMD.AMS = 1)	Transfer destination (EDMAMD.DIR = 1)	Only writes data of cluster size. Transfers data from the cluster buffers.

19.4 Transfer Operation

Descriptions of examples of operations in transfer by the EXDMAC are given in the following passages.
Operations of the EXDMAC are synchronized by the external bus clock (BCLK). The examples that follow are for cases where the external bus clock (BCLK) and the signal output on the BCLK pin are at the same frequency unless there is a particular reason for doing otherwise.

Note: An idle cycle is inserted between two adjacent accesses depending on the external bus recovery cycle setting. One idle cycle, however, is always inserted between a read access and a write access even if the recovery cycle is set to 0. See section 16, Buses, for details on the recovery cycle.

19.4.1 Normal/Repeat Transfer Operation

(1) Dual Address Mode

Figure 19.11 shows the bus cycle example in normal-transfer dual address mode. In the example, a 16-bit data (SZ[1:0] = 01b in EDMTMD of EXDMACn) is transferred from a device with 16-bit 2-cycle access to another device with 16-bit 2-cycle access, which is started at the falling edge of EDREQn.
The bus cycles in repeat-transfer dual address mode are the same as those in normal-transfer dual address mode.

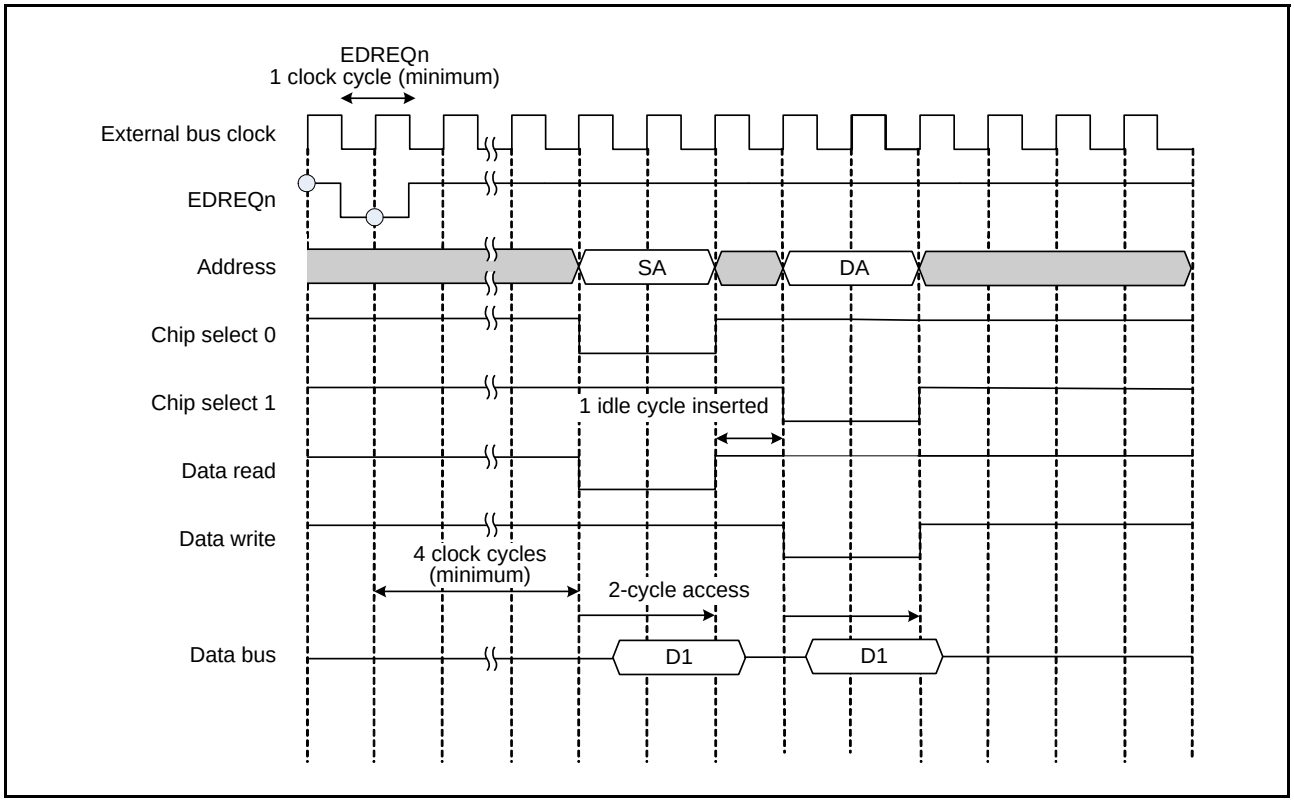


Figure 19.11 Bus Cycle Example in Normal-Transfer Dual Address Mode

(2) Single Address Mode

In single address mode, data is read from the transfer source address and directly transferred to the transfer destination device without being taken in the LSI. Here, EDACKn is output to one of the external transfer-destination and transfer-source devices, and the address is simultaneously output to the other transfer device for access.
With the DIR bit in EDMAMD of EXDMACn set to 0, the transfer source address is output to the external bus and EDACKn is output to the transfer destination. With the DIR bit in EDMAMD of EXDMACn set to 1, the transfer destination address is output to the external bus and the EDACKn is output to the transfer source. Figure 19.12 shows the data flow in single address mode.

Figure 19.13 shows the bus cycle example in normal-transfer single address mode. In the example, one data is transferred in 2-cycle access when the DIR bit in EDMAMD of EXDMACn is set to 1 (transfer destination address is output) and when set to 0 (transfer source address is output).

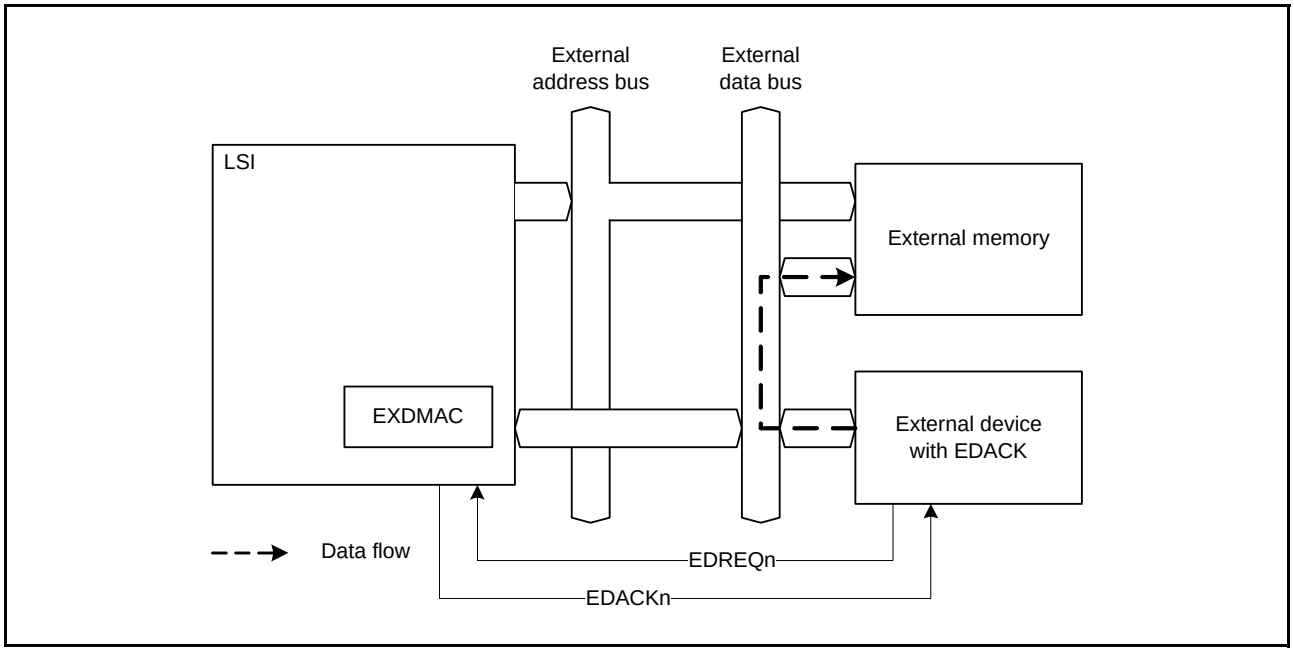


Figure 19.12 Data Flow in Single Address Mode (when EDMAMD.DIR = 1)

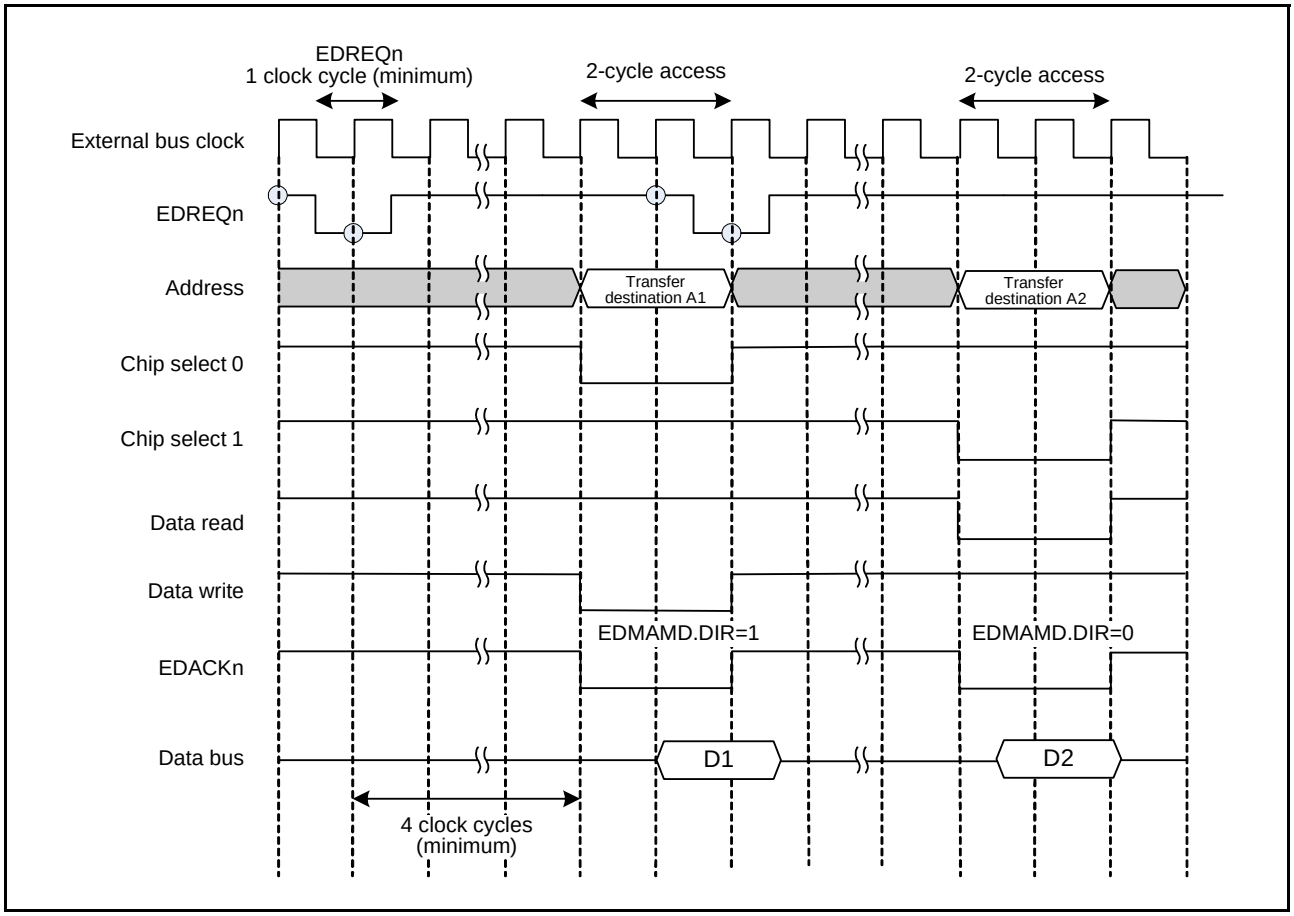


Figure 19.13 Bus Cycle Example in Normal-Transfer Single Address Mode

19.4.2 Block Transfer Operation

(1) Dual Address Mode

Figure 19.14 shows the bus cycle example in block-transfer dual address mode. In the example, a 16-bit data (SZ[1:0] = 01 in EDMTMD of EXDMACn) is transferred from a device with 16-bit 2-cycle access to another device with 16-bit 2-cycle access when the block size is 3.

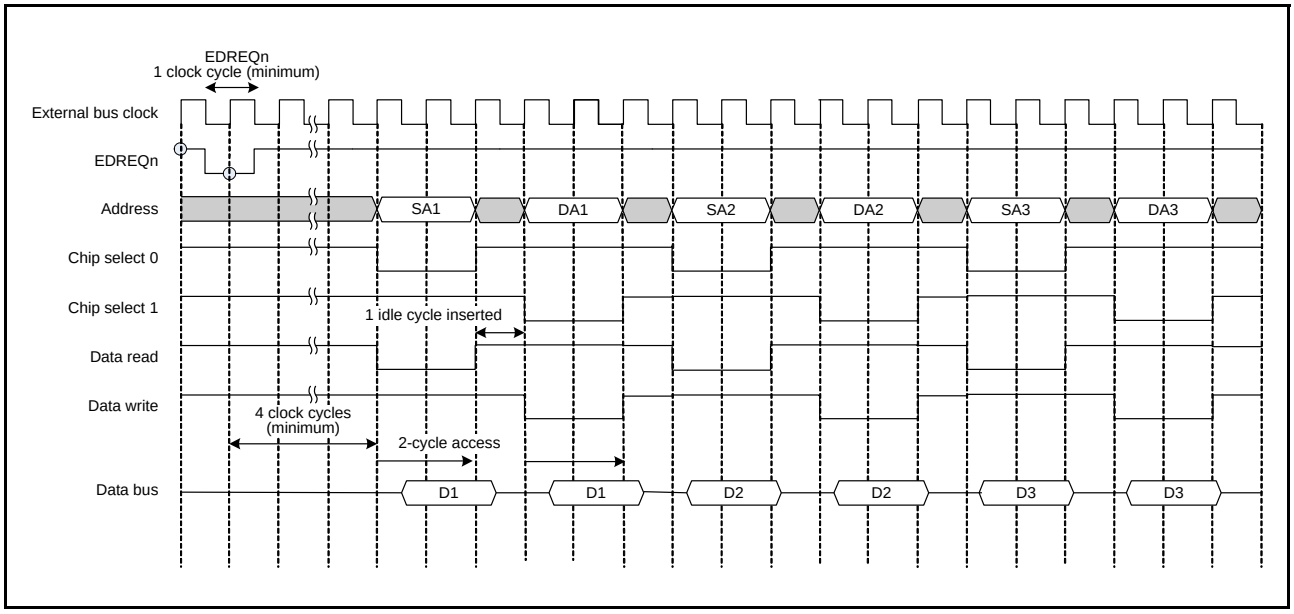


Figure 19.14 Bus Cycle Example in Block-Transfer Dual Address Mode

(2) Single Address Mode

Figure 19.15 shows the bus cycle example in block-transfer single address mode. In the example, a 16-bit data (SZ[1:0] = 01 in EDMTMD of EXDMACn) is transferred in three bus clock cycles from a device with EDACKn with 16-bit access to another device with 16-bit access when the block size is 3.

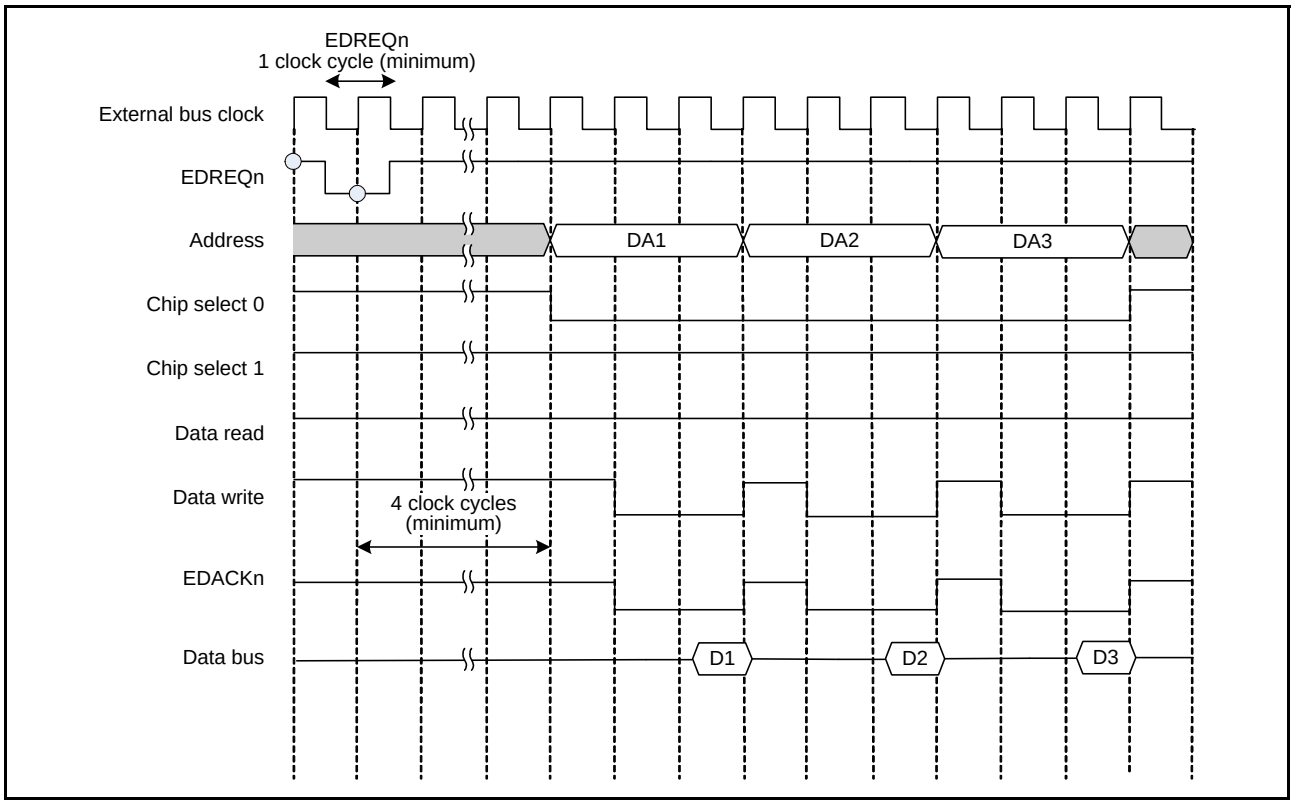


Figure 19.15 Data Flow in Block Transfer Single Address Mode

19.4.3 Cluster Transfer Operation

(1) Dual Address Mode

In cluster-transfer dual address mode, cluster-size data is transferred from the external transfer source device to the external transfer destination device via the cluster buffers. Figure 19.16 shows the data flow in cluster-transfer dual address mode and Figure 19.17 shows the bus cycle example, in which one cluster is transferred in two clock cycles when the cluster size is three.

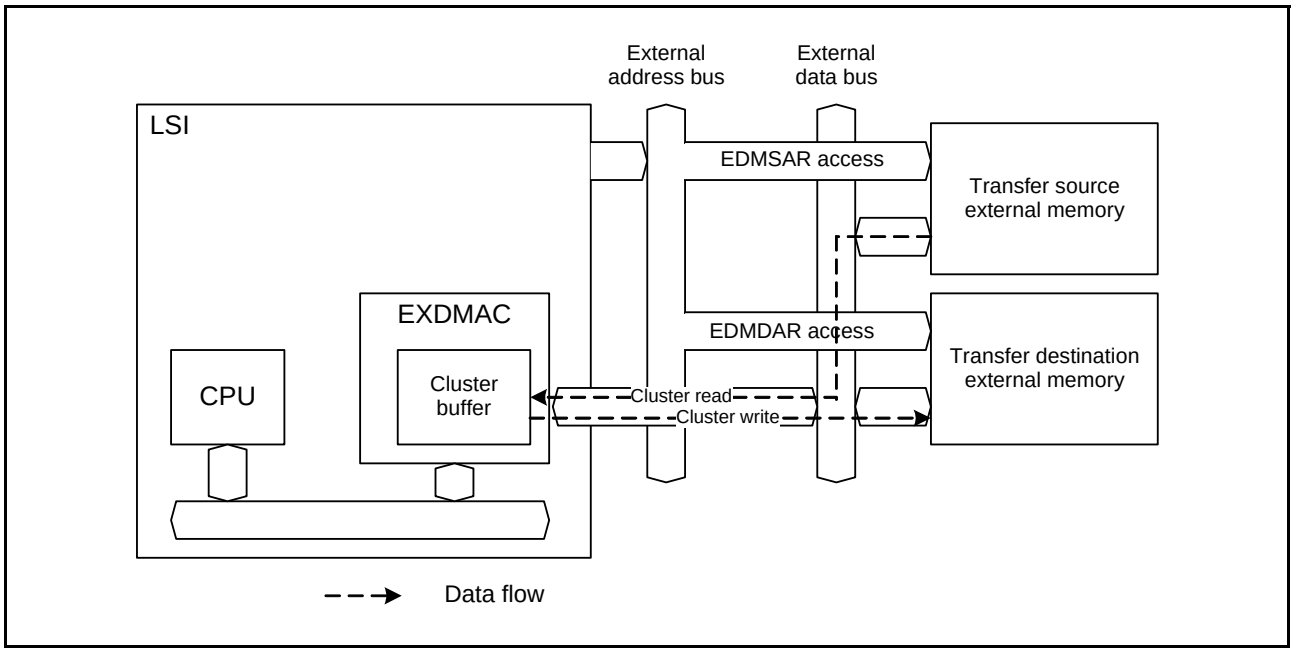


Figure 19.16 Data Flow in Cluster-Transfer Dual Address Mode

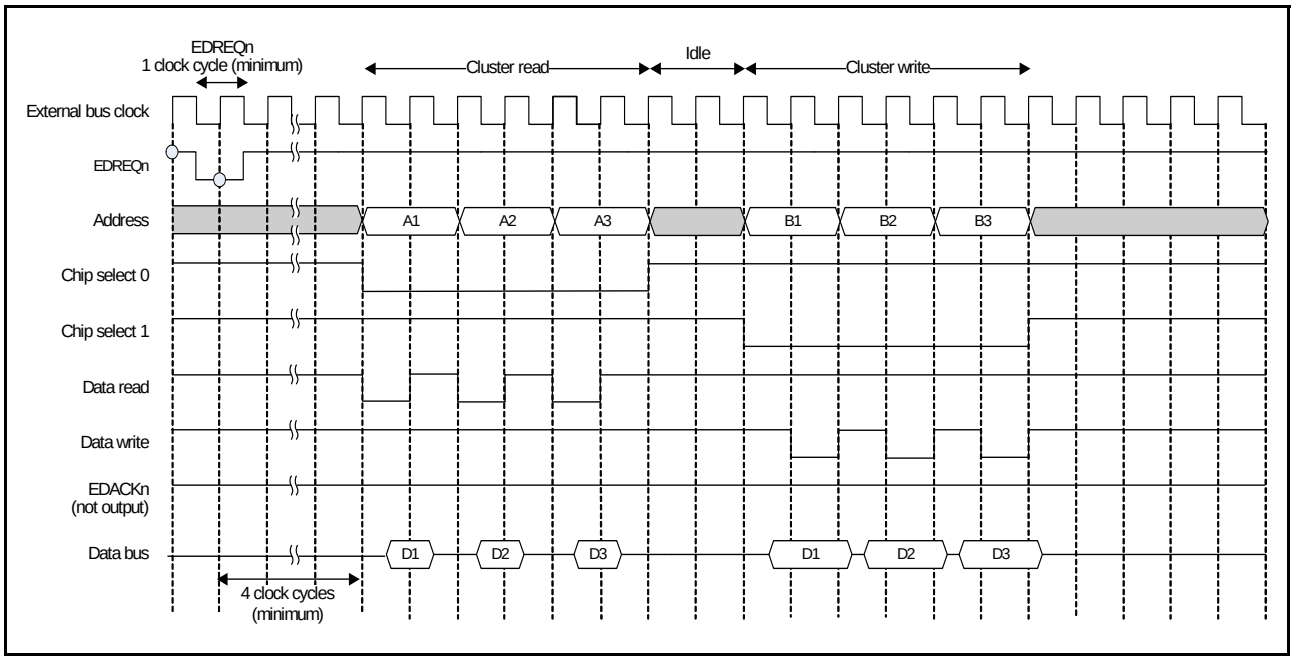


Figure 19.17 Bus Cycle Example in Cluster-Transfer Dual Address Mode

(2) Read Address Mode

In cluster-transfer read address mode, cluster-size data is transferred from the external transfer source device to the cluster buffers. The data transferred in the cluster buffers can be read by the CPU. Figure 19.18 shows the data flow in cluster-transfer read address mode and Figure 19.19 shows the bus cycle example, in which one cluster is transferred in two clock cycles when the cluster size is six.

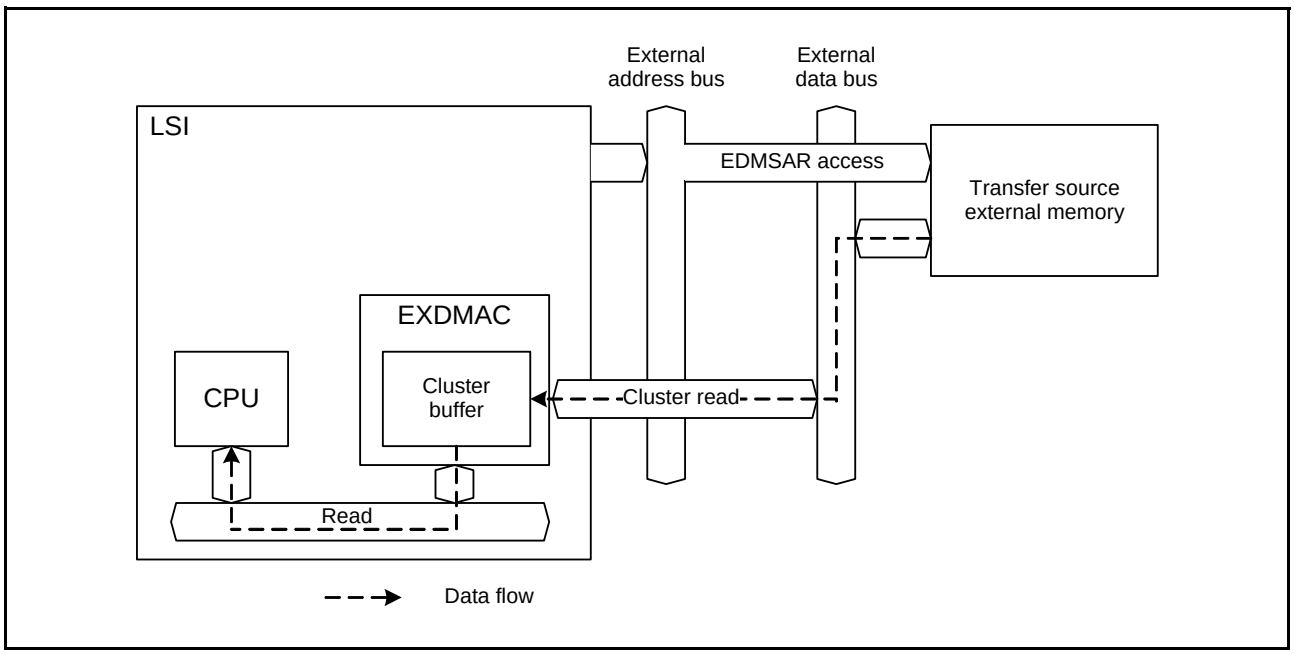


Figure 19.18 Data Flow in Cluster-Transfer Read Address Mode

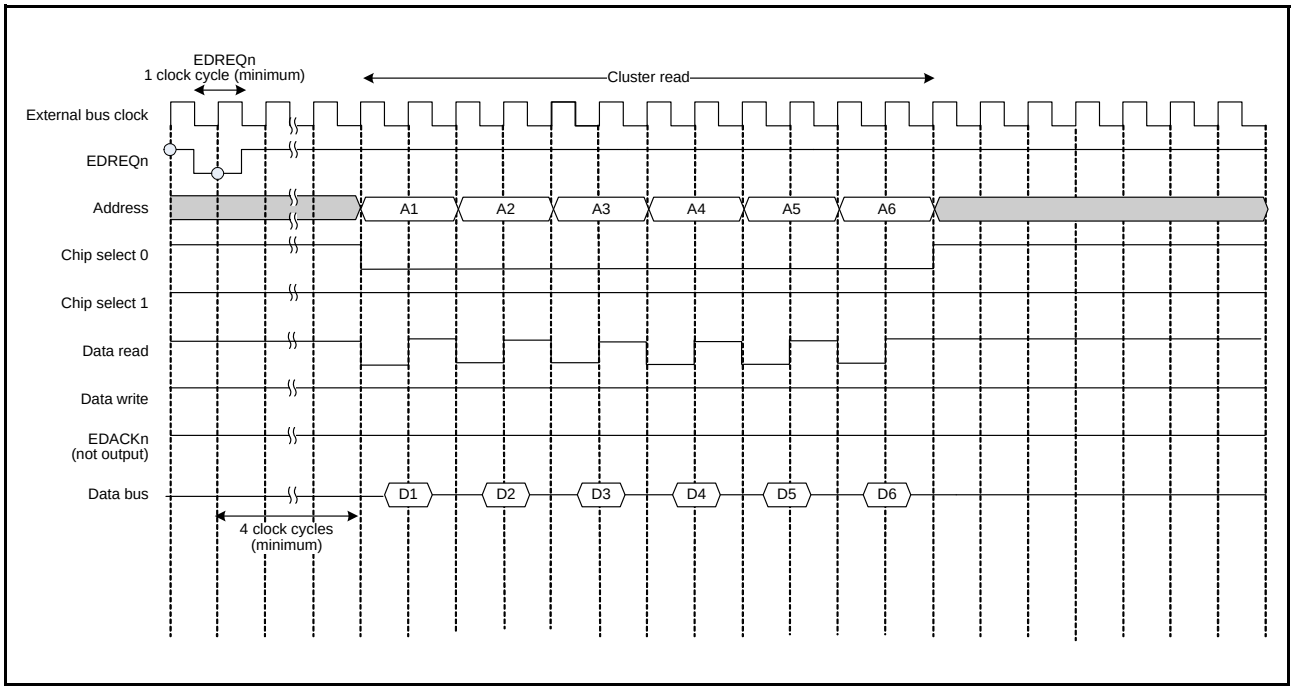


Figure 19.19 Bus Cycle Example in Cluster-Transfer Read Address Mode

(3) Write Address Mode

In cluster-transfer write address mode, the data is written to the cluster buffers by the internal bus master such as the CPU, DMACA, and DTC and then transferred to the external transfer destination device. Figure 19.20 shows the data flow in cluster-transfer write address mode and Figure 19.21 shows the bus cycle example, in which one cluster is transferred in two clock cycles when the cluster size is six.

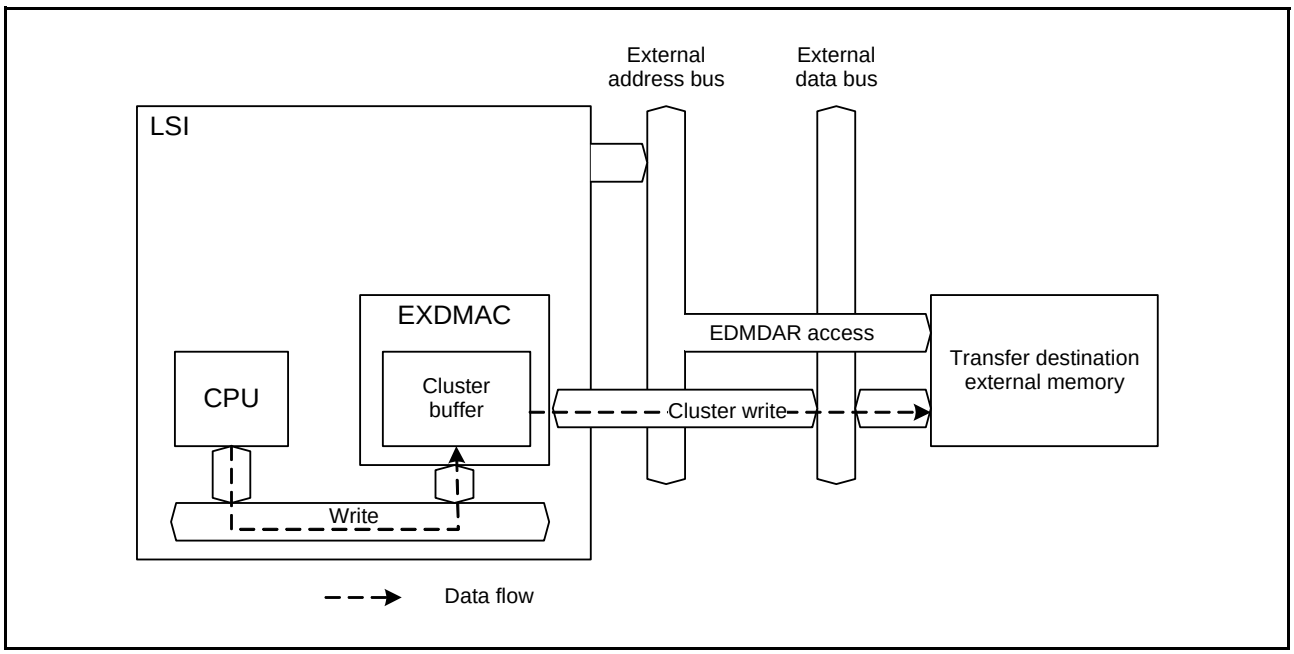


Figure 19.20 Data Flow in Cluster-Transfer Write Address Mode

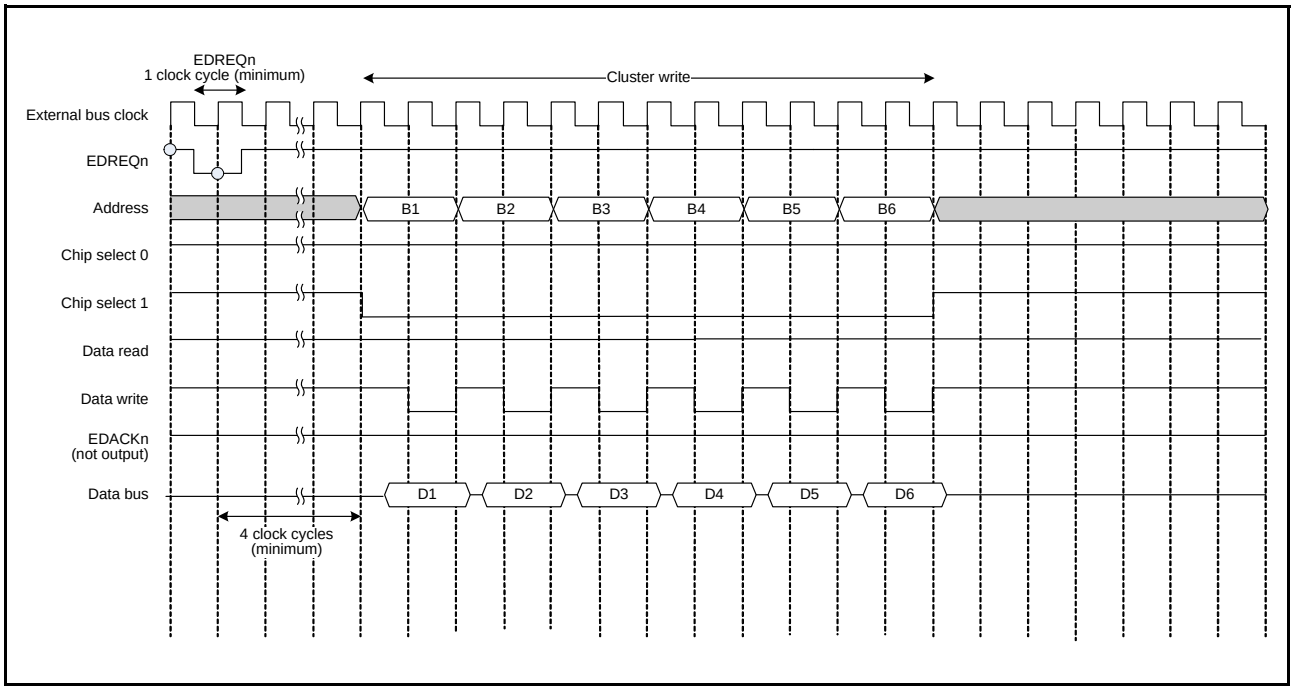


Figure 19.21 Bus Cycle Example in Cluster-Transfer Write Address Mode

19.5 Request Sources and Procedures

19.5.1 Request Sources

The EXDMAC can be triggered by software, by an external DMA transfer request pin (EDREQn pins), or by the DMA transfer requests from the peripheral modules (the software configurable interrupt B source select register 144 (SLIBR144) or the software configurable interrupt A source select register 208 (SLIAR208) should be set for channel 0, and the software configurable interrupt B source select register 145 (SLIBR145) or the software configurable interrupt A source select register 209 (SLIAR209) should be set for channel 1). Setting the DCTG[1:0] bits in EDMTMD of EXDMACn selects the request source.

(1) Trigger by Software

Setting the EXDMACn.EDMREQ.DCTG[1:0] bits to 00b enables the trigger by software.
To start DMA transfer by software, follow the procedure below.

1. Check that the EXDMACn.EDMREQ.SWREQ bit is 0 (DMA transfer is not requested).
2. Set the EXDMACn.EDMTMD.DCTG[1:0] bits to 00b (activation by software).
3. Set the EXDMACn.EDMCNT.DTE bit to 1 (enables DMA transfer).
4. Set the EXDMACn.EDMREQ.CLRS bit (DMA software start bit auto clear select) and also set the EXDMACn.EDMREQ.SWREQ bit to 1 (DMA transfer is requested).

When the EXDMACn.DMREQ.CLRS bit is 0 (SWREQ bit is cleared after DMA transfer is started by software), the EXDMACn.EDMREQ.SWREQ bit is set to 0 after data transfer is started in response to a DMA transfer request. When the CLRS bit is 1 (SWREQ bit is not cleared after DMA transfer is started by software), the SWREQ bit is not set to 0. In this case, a DMA transfer request is issued again after completion of a transfer.

(2) Trigger by external DMA transfer request Pin (EDREQn)

Setting the EXDMACn.EDMTMD.DCTG[1:0] bits to 10b enables the trigger by the external DMA transfer request pins.
To set the trigger by the external DMA transfer request pin, follow the procedure below.

1. Set the detection mode by the EXDMACn.EDMRMD.DREQS[1:0] bits.
2. Set the EXDMACn.EDMTMD.DCTG[1:0] bits to 10b (external DMA transfer request pins).
3. Set the EXDMACn.EDMERF.EREQ flag to 1 to clear the EREQ flag.
4. Set the EXDMACn.EDMCNT.DTE bit to 1 (enables DMA transfer).

When the falling edge or rising edge is selected by using the EXDMACn.EDMRMD.DREQS[1:0] bits, if the external DMA transfer request pin detects an edge, the EXDMACn.EDMERF.EREQ flag is set to 1. The EXDMACn.EDMERF.EREQ flag is set to 0 when the DMA transfer is started by the external request. Moreover, this flag is set to 0 by writing 1 to it.
When the low level detection is set by the EXDMACn.EDMRMD.DREQS[1:0] bits, if the external DMA transfer request pin is low level, the EXDMACn.EDMERF.EREQ flag is set to 1 (DMA transfer is requested). If the external DMA transfer request pin is high level, the EXDMACn.EDMERF.EREQ flag is 0 (DMA transfer is not requested). In case of the low level detection, when the DMA transfer is started by the external request or 1 is written to the flag, the EXDMACn.EDMERF.EREQ flag is not set to 0.
When the EDMAST.DMST bit and the EXDMACn.EDMCNT.DTE bit are set to 1 (enables DMA transfer) while the EXDMACn.EDMERF.EREQ flag is 1 (DMA transfer is requested), the DMA transfer is started.
The value of the EXDMACn.EDMERF.EREQ flag is retained regardless of the settings in the EDMAST.DMST and EXDMACn.EDMCNT.DTE bits.
Figure 19.22 and Figure 19.23 show the external DMA transfer request timings in the falling edge detection mode and low level detection mode, respectively.

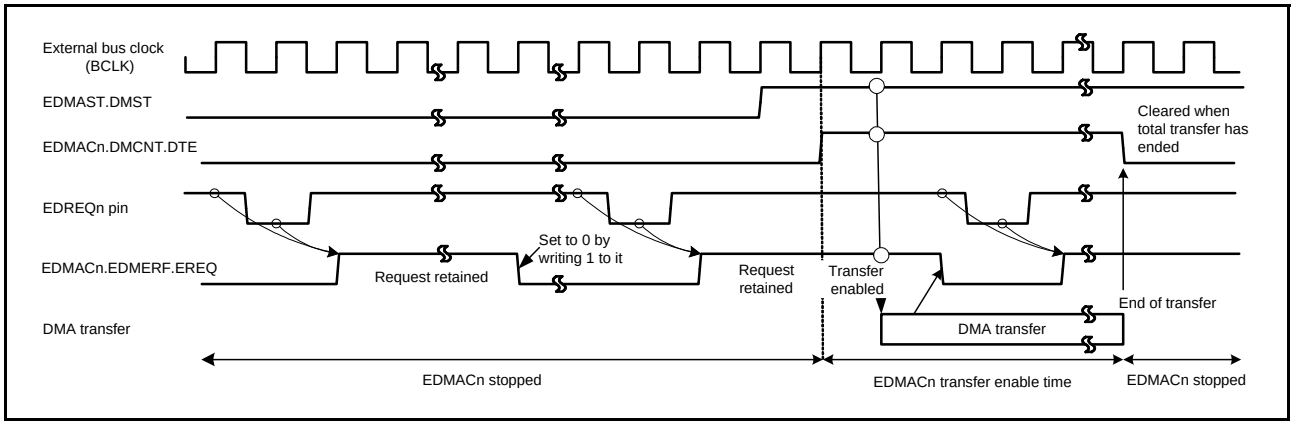


Figure 19.22 External DMA Transfer Request Timing in Falling-Edge Detection Mode

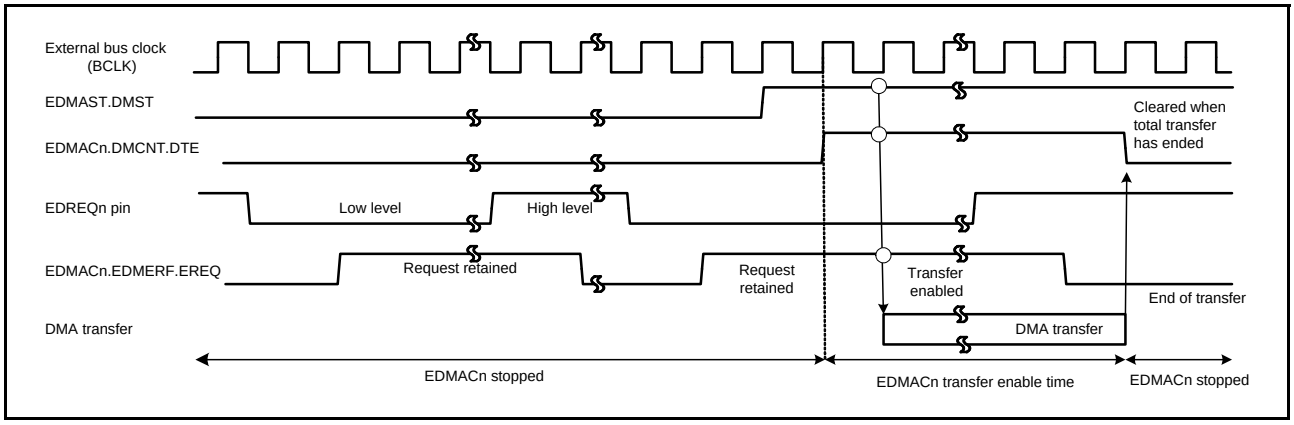


Figure 19.23 External DMA Transfer Request Timing in Low-Level Detection Mode

(3) Trigger by DMA Transfer Requests from Peripheral Modules

Setting the DCTG[1:0] bits in EDMTMD of EXDMACn to 11b enables the trigger by DMA transfer requests from the peripheral modules (an interrupt specified by the software configurable interrupt B source select register (ICU.SLIBR144 or ICU.SLIBR145) or the software configurable interrupt A source select register (ICU.SLIAR208 or ICU.SLIAR209)).

To start DMA transfer by DMA transfer requests from the peripheral modules follow the procedure below.

1. Set the software configurable interrupt B source select register 144 (SLIBR144) or the software configurable interrupt A source select register 208 (SLIAR208) for channel 0, and set the software configurable interrupt B source select register 145 (SLIBR145) or the software configurable interrupt A source select register 209 (SLIAR209) for channel 1 (For the setting procedure, see section 15.4.5, Software Configurable Interrupts, in section 15, Interrupt Controller (ICUA)).
2. Set the EXDMACn.EDMTMD.DCTG[1:0]bits to 11b (DMA transfer requests from the peripheral modules).
3. Set the EXDMACn.EDMPRF.PREQ flag to 1, and then clear it to 0.
4. Set the EXDMACn.EDMCNT.DTE bit to 1 (DMA transfer is enabled).

None of the values of the interrupt request enable bits (IERm.IENj) affect EXDMACn requests initiated by peripheral modules.

When a DMA transfer request is input from the peripheral modules, the EXDMACn.EDMPRF.PREQ flag is set to 1 (DMA transfer is requested). The EXDMACn.EDMPRF.PREQ flag is set to 0 (DMA transfer is not requested) when the DMA transfer is started by the peripheral module request.

This flag is set to 0 by writing 1 to it.
When the EDMAST.DMST bit is set to 1 (EXDMAC module start) and the EXDMACn.EDMCNT.DTE bit is set to 1 (DMA transfer is enabled) while the EXDMACn.EDMPRF.PREQ flag is 1, the DMA transfer is started.
The value of the EXDMACn.EDMPRF.PREQ flag is retained regardless of the settings of the EDMAST.DMST and EXDMACn.EDMCNT.DTE bits.
The EXDMACn.EDMPRF.PREQ flag is set to 1 regardless of the state of the module stop bit when the corresponding EXDMACn is started by an internal peripheral module.

19.5.2 Activating the EXDMAC

Figure 19.24 shows the register setting procedure.

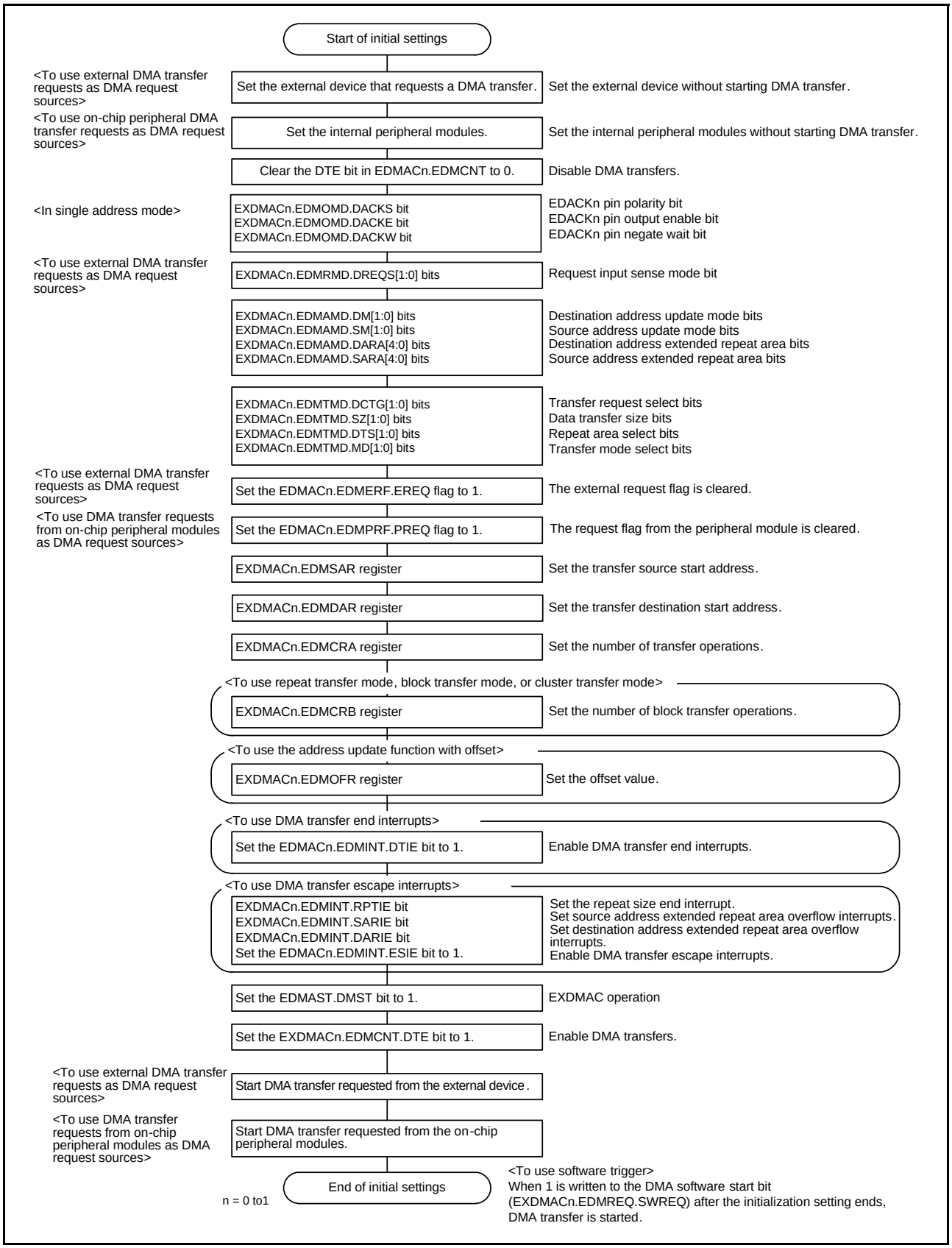


Figure 19.24 Register Setting Procedure

19.5.3 Starting DMA Transfer

Setting the DTE bit in EDMCNT of EXDMACn to 1 (DMA transfer enabled) and setting the DMST bit in EDMAST to 1 (EXDMAC start) enable DMA transfer of channel n (n = 0, 1).
When DMA transfer requests are generated, channel arbitration is performed where a DMA transfer request of higher-priority channel is accepted and DMA transfer of the channel starts. When a DMA transfer request is accepted and DMA transfer starts, the ACT flag in EDMSTS of EXDMACn is set to 1 (DMA transfer is in progress).

19.5.4 Registers during DMA Transfer

The EXDMAC registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers to be updated are EDMSAR, EDMDAR, EDMCRA, EDMCRB, EDMCNT, and EDMSTS of EXDMACn.

(1) EXDMA Source Address Register (EXDMACn.EDMSAR)

When data has been transferred in response to one transfer request, the contents of EDMSAR are updated to the address to be accessed by the next transfer request.
For details on register update operation in each transfer mode, refer to Table 19.4 to Table 19.7.

(2) EXDMA Destination Address Register (EXDMACn.EDMDAR)

When data has been transferred in response to one transfer request, the contents of EDMDAR are updated to the address to be accessed by the next transfer request.
For details on register update operation in each transfer mode, refer to Table 19.4 to Table 19.7.

(3) EXDMA Transfer Count Register (EXDMACn.EDMCRA)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.
For details on register update operation in each transfer mode, refer to Table 19.4 to Table 19.7.

(4) EXDMA Block Transfer Count Register (EXDMACn.EDMCRB)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.
For details on register update operation in each transfer mode, refer to Table 19.4 to Table 19.7.

(5) EXDMA Transfer Enable Bit (EXDMACn.EDMCNT.DTE)

The EXDMACn.EDMCNT.DTE bit provides a way to control enabling or prohibition of transfer by writing to the relevant registers.
The EXDMACn.EDMCNT.DTE bit is set to 0 when any of the following conditions is generated by the DMA transfer.

- When the specified total volume of data transfer is completed
- When DMA transfer is stopped by the repeat size end interrupt
- When DMA transfer is stopped by the extended repeat area overflow interrupt

Writing to the registers of an EXDMAC channel is prohibited if the corresponding EXDMACn.EDMCNT.DTE bit is 1 (except to the EXDMACn.EDMCNT register itself).
Change the settings of the registers as required after writing 0 to the EXDMACn.EDMCNT.DTE bit.

(6) DMA Active Flag (EXDMACn.EDMSTS.ACT)

The ACT flag in EDMSTS of EXDMACn indicates whether the EXDMACn is in the idle or active state.
This flag is set to 1 when the EXDMACn starts data transfer, and is set to 0 when data transfer in response to one transfer request is completed.
Even when DMA transfer is stopped by writing 0 to the DTE bit in EDMCNT of EXDMACn, this flag remains 1 until DMA transfer is completed.

(7) Transfer End Interrupt Flag (EXDMACn.EDMSTS.DTIF)

The DTIF flag in EDMSTS of EXDMACn is set to 1 after transfer of the total transfer size of data is completed.
When both this flag and the DTIE bit in EDMINT of EXDMACn are set to 1, a transfer end interrupt is requested.
This flag is set to 1 when the DMA transfer bus cycle is completed and the ACT flag in EDMSTS of EXDMACn is set to 0 indicating the DMA transfer end.
This flag is automatically set to 0 when the DTE bit in EDMCNT of EXDMACn is set to 1 during the interrupt handling.

(8) Transfer Escape End Interrupt Flag (EXDMACn.EDMSTS.ESIF)

The ESIF flag in EDMSTS of EXDMACn is set to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When this flag and the ESIE bit in EDMINT of EXDMACn are set to 1, a transfer escape end interrupt is requested.
This flag is set to 1 when the bus cycle of the DMA transfer having caused the interrupt request is completed and the ACT flag in EDMSTS of EXDMACn is set to 0 indicating the DMA transfer end.
This flag is automatically set to 0 when the DTE bit in EDMCNT of EXDMACn is set to 1 during an interrupt handling.
Before sending an interrupt request from the EXDMACn to the CPU or the DTC, the interrupt control register must be set.
For details, see section 15, Interrupt Controller (ICUA).

19.5.5 Channel Priority

When multiple DMA transfer requests are present, the EXDMAC determines the priority of channels that have DMA transfer requests. The channel priority is fixed as channel 0 > channel 1.
When a DMA transfer request is generated during data transfer, channel arbitration is started after the final data has been transferred, and DMA transfer of the higher-priority channel starts.

19.6 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the DTE bit in EDMCNT and the ACT flag in EDMSTS of EXDMACn are changed from 1 to 0, indicating that DMA transfer has ended.

19.6.1 Transfer End by Completion of Specified Total Number of Transfer Operations

(1) In Normal Transfer Mode (EXDMACn.EDMTMD.MD[1:0] = 00b)

When the value of the EXDMACn.EDMCRAL register changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in EDMCNT of EXDMACn is set to 0 and the DTIF flag in EDMSTS of EXDMACn is set to 1 at the same time. If the DTIE bit in EDMINT of EXDMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(2) In Repeat Transfer Mode (EXDMACn.EDMTMD.MD[1:0] = 01b)

When the value of DMCRB of EXDMACn changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in EDMCNT of EXDMACn is set to 0 and the DTIF flag in EDMSTS of EXDMACn is set to 1 at the same time. If the DTIE bit in EDMINT of EXDMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(3) In Block Transfer Mode (EXDMACn.EDMTMD.MD[1:0] = 10b)

When the value of EDMCRB of EXDMACn changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in EDMCNT of EXDMACn is set to 0 and the DTIF flag in EDMSTS of EXDMACn is set to 1 at the same time. If the DTIE bit in EDMINT of EXDMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(4) In Cluster Transfer Mode (EXDMACn.EDMTMD.MD[1:0] = 11b)

When the value of EDMCRB of EXDMACn changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in EDMCNT of EXDMACn is set to 0 and the DTIF flag in EDMSTS of EXDMACn is set to 1 at the same time. If the DTIE bit in EDMINT of EXDMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

Before sending an interrupt request from the EXDMACn to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUA).

19.6.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, a repeat size end interrupt is requested when transfer of a 1-repeat size of data is completed while the RPTIE bit in EDMINT of EXDMACn is set to 1. When the interrupt is requested to complete DMA transfer, the DTE bit in EDMCNT of EXDMACn is set to 0 and the ESIF flag in EDMSTS of EXDMACn is set to 1. If the ESIE bit in EDMINT of EXDMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC. Here, the transfer can be resumed by writing 1 to the DTE bit in EDMCNT of EXDMACn.
A repeat size end interrupt can be requested also in block transfer mode (or cluster transfer mode). In block transfer mode (or cluster transfer mode), the interrupt is requested in the same way as in repeat transfer mode when transfer of a 1-block size (or 1-cluster) data is completed.
Before sending an interrupt request from the EXDMACn to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUA).

19.6.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area occurs while the extended repeat area is specified and the SARIE or DARIE bit in EDMINT of EXDMACn is set to 1, an interrupt by an extended repeat area overflow is requested. When the interrupt is requested, the DMA transfer is terminated, the DTE bit in EDMCNT of EXDMACn is set to 0, and the ESIF flag in EDMSTS of EXDMACn is set to 1. If the ESIE bit in EDMINT of EXDMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.
Even if an interrupt by an extended repeat area overflow is requested during a read cycle, the following write cycle is performed.
In block transfer mode (or cluster transfer mode), even if an interrupt by an extended repeat area overflow is requested during a 1-block (or 1-cluster) transfer, the remaining data in the block (or the cluster) is transferred; transfer is terminated after a block (or cluster) transfer.
Before sending an interrupt request from the EXDMACn to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt Controller (ICUA).

19.7 Interrupts

The EXDMAC can output one interrupt request to the CPU or the DTC for each channel. Table 19.10 shows the relation among the interrupt sources, the interrupt status bits, and the interrupt enable bits. Figure 19.25 shows the schematic logic diagram of interrupt outputs. The procedures for suspending or resuming DMA transfer by the EXDMAC interrupt are shown in Figure 19.26.

Table 19.10 Relation among Interrupt Sources, Interrupt Status Bits, and Interrupt Enable Bits

Interrupt Sources		Interrupt Enable Bits	Interrupt Status Flags	Request Output Enable Bits
Transfer end		—	EXDMACn.EDMSTS.DTIF	EXDMACn.EDMINT.DTIE
Escape transfer end	Repeat size end	EXDMACn.EDMINT.RPTIE	EXDMACn.EDMSTS.ESIF	EXDMACn.EDMINT.ESIE
	Source address extended repeat area overflow	EXDMACn.EDMINT.SARIE		
	Destination address extended repeat area overflow	EXDMACn.EDMINT.DARIE		

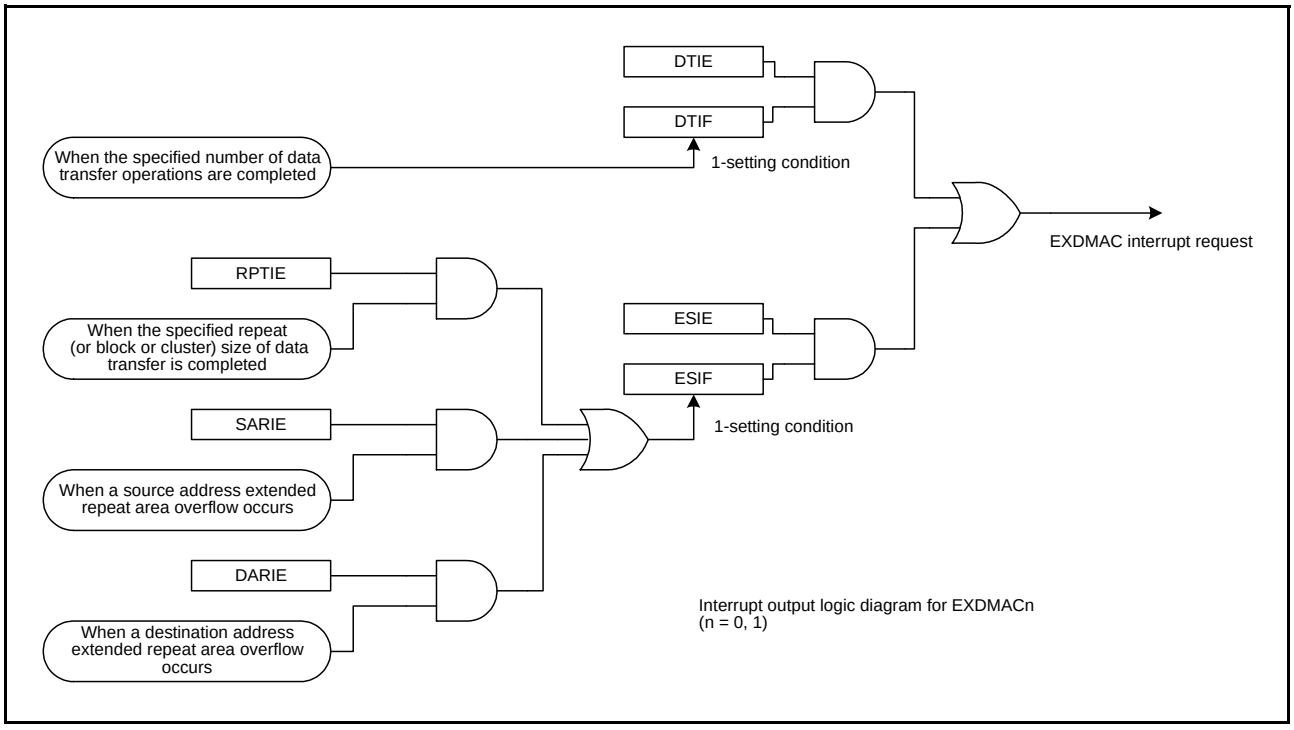


Figure 19.25 Schematic Logic Diagram of Interrupt Outputs

Specifically, the different procedures are used for canceling an interrupt to restart DMA transfer in the following two cases: (1) discontinuing or terminating DMA transfer and (2) continuing DMA transfer.

(1) When Discontinuing or Terminating DMA Transfer

Write 0 to the DTIF flag in EDMSTS of EXDMACn to clear a transfer end interrupt, and to the ESIF flag in EDMSTS of EXDMACn to clear a repeat size interrupt and an extended repeat area overflow interrupt. The EXDMACn remains in the stop state. When starting another DMA transfer after that, set the appropriate registers, and set the DTE bit in EDMCNT of EXDMACn to 1.

(2) When Continuing DMA Transfer

Write 1 to the DTE bit in EDMCNT of EXDMACn. The ESIF flag in EDMSTS of EXDMACn is automatically set to 0 (interrupt source cleared), and DMA transfer is resumed.

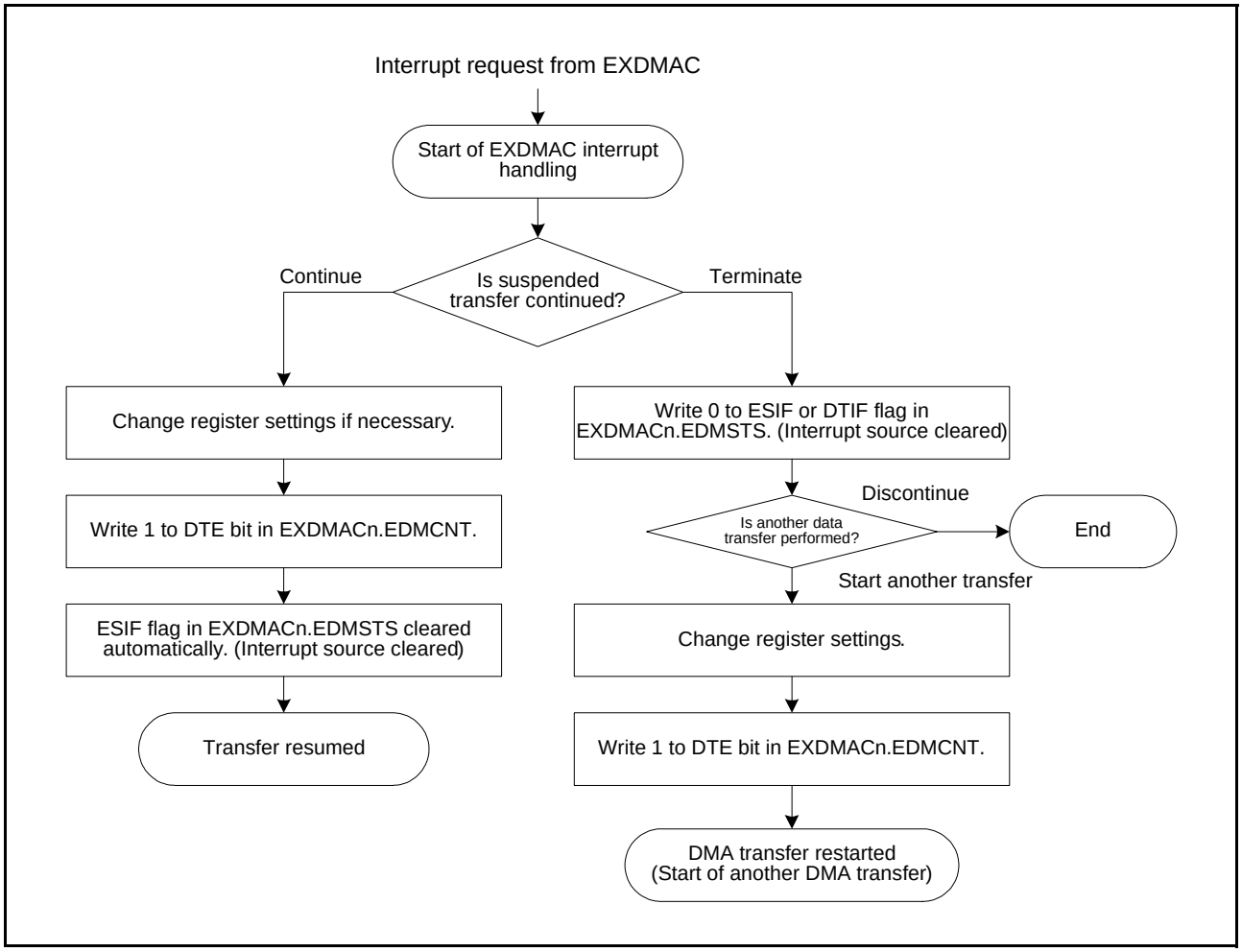


Figure 19.26 Procedures for Suspending or Resuming DMA Transfer by the EXDMAC Interrupt

19.8 Low-Power Consumption Function

To place the EXDMAC in the module-stop state, all-module clock stop mode, software-standby mode, or deep software-standby mode, clear the EDMAST.DMST bit to 0 (EXDMAC stopped), and then perform the following processing.

(1) Module-Stop Function

Writing 1 to the MSTPCRA.MSTPA29 bit (transition to the module-stop state) enables the module-stop function of the EXDMAC. If DMA transfer is in progress at the time a 1 is written to the MSTPA29 bit, the transition to the module-stop state proceeds after DMAC transfer has ended.
Do not access the EXDMAC registers while the MSTPCRA.MSTPA29 bit is 1.
Writing 0 to the MSTPA29 bit releases the EXDMAC from the module-stop state.

(2) All-Module Clock Stop Mode

Follow the procedure in section 11.6.2.1, Transition to All-Module Clock Stop Mode, in section 11, Low Power Consumption. If DMA transfer is in progress at the time the WAIT instruction is executed, the EXDMAC can enter all-module clock stop mode after completion of the current DMA transfer.
After the EXDMAC returns from all-module clock stop mode, writing 0 to the MSTPA29 bit releases the EXDMAC from the module-stop state.

(3) Software Standby and Deep Software Standby Modes

Follow the procedure in section 11.6.3.1, Transition to Software Standby Mode or section 11.6.4.1, Transition to Deep Software Standby Mode, in section 11, Low Power Consumption.
If DMA transfer is in progress at the time the WAIT instruction is executed, the EXDMAC enters software standby mode or deep software stand by mode after completion of the current DMA transfer.

(4) Notes on Low-Power Consumption Function

For the timing of WAIT instruction execution and register settings, see section 11.7.6, Timing of Wait Instructions, in section 11, Low Power Consumption.
To perform DMA transfer after returning from low-power consumption mode, set the EDMAST.DMST bit to 1 again.

19.9 EDACKn Operation in Single Address Mode

In single address mode, EDACKn is output to one of the external transfer-source or transfer-destination devices and the address is simultaneously output to the other transfer device for access.
When the external device receiving EDACKn transfers data to/from the CS area, the EDACKn negation timing can be adjusted by setting the DACKW bit in EDMOMD of EXDMACn. Specifically, the timing can be advanced by one BCLK cycle if the external device is a transfer destination and delayed by one BCLK cycle if the external device is a transfer source. When the external device receiving EDACKn transfers data to/from the SDRAM, the EDACKn negation timing cannot be adjusted by the DACKW bit EDMOMD of EXDMACn. If one of the parties for transfer by the EXDMAC (source or destination) is the SDRAM area, the EXDMACn.EDMOMD.DACKSEL bit can be set to 1 so that the EDACKn signal is only asserted for one-half of the SDCLK cycle in the latter half of the data-valid interval. The EXDMACn.EDMOMD.DACKSEL cannot be used to adjust the period for assertion of the EDACKn signal when a party for transfer is a CS area. For the CS area addresses and SDRAM area addresses, refer to section 4, Address Space.
The following sections show EDACKn operation examples in single address mode, in which data is transferred to/from the CS and SDRAM areas in normal and block transfer modes.

19.9.1 EDACKn Operation Example in Normal-Transfer (CS Area) Single Address Mode

Figure 19.27 shows the operation example in which data is transferred from the CS area to the device with EDACKn in normal transfer mode. Setting the DACKW bit in EDMOMD of EXDMACn to 1 allows EDACKn to be negated one BCLK cycle before the data read signal is negated.
Figure 19.28 shows the operation example in which data is transferred from the device with EDACKn to the CS area in normal transfer mode. Setting the DACKW bit to 1 allows EDACKn to be negated one BCLK cycle after the data write signal is negated.
For the data read signal, data write signal, and CS area access timing setting registers, refer to section 16, Buses.

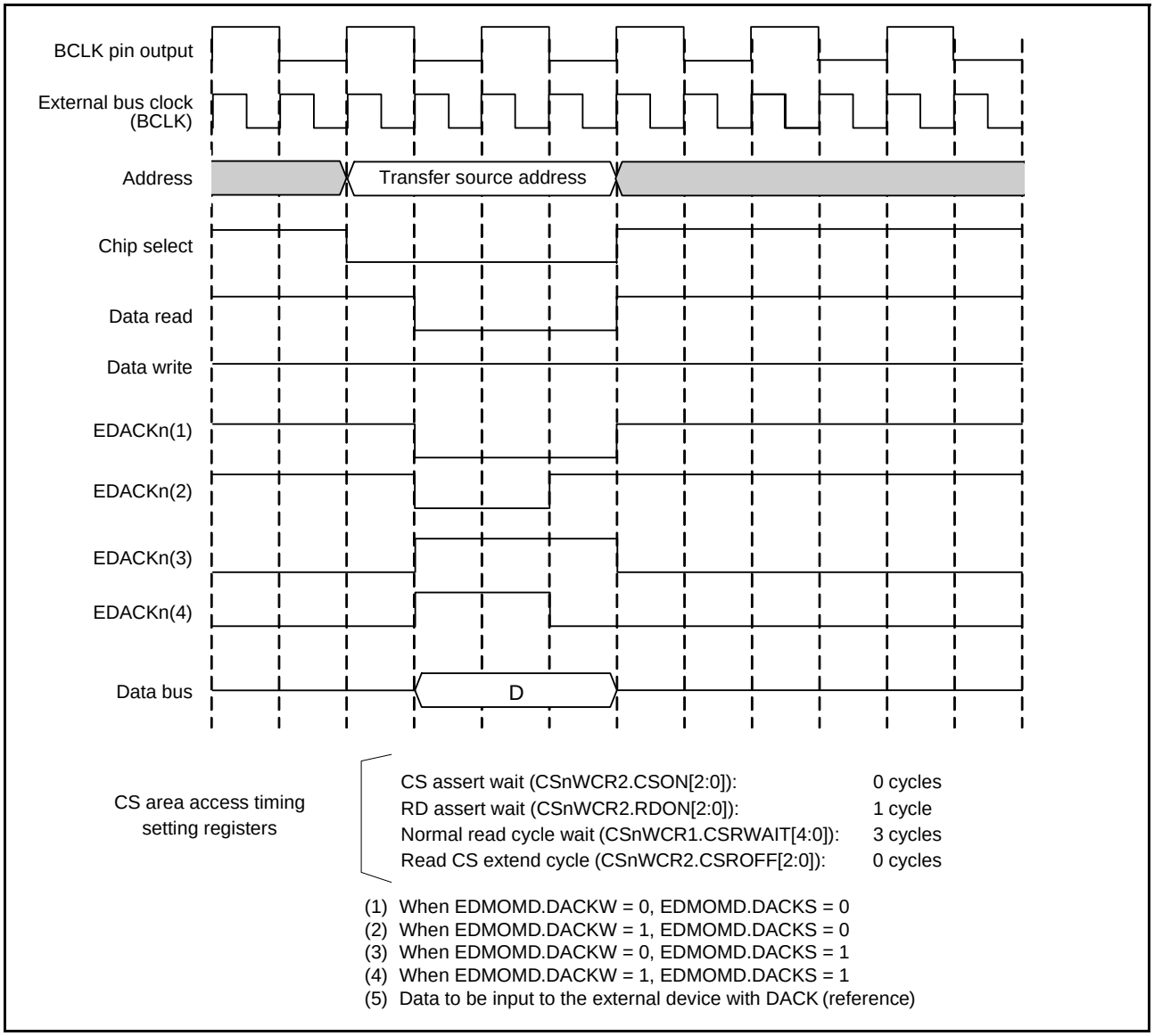


Figure 19.27 Operation Example in Normal-Transfer (CS Area Read) Single Address Mode

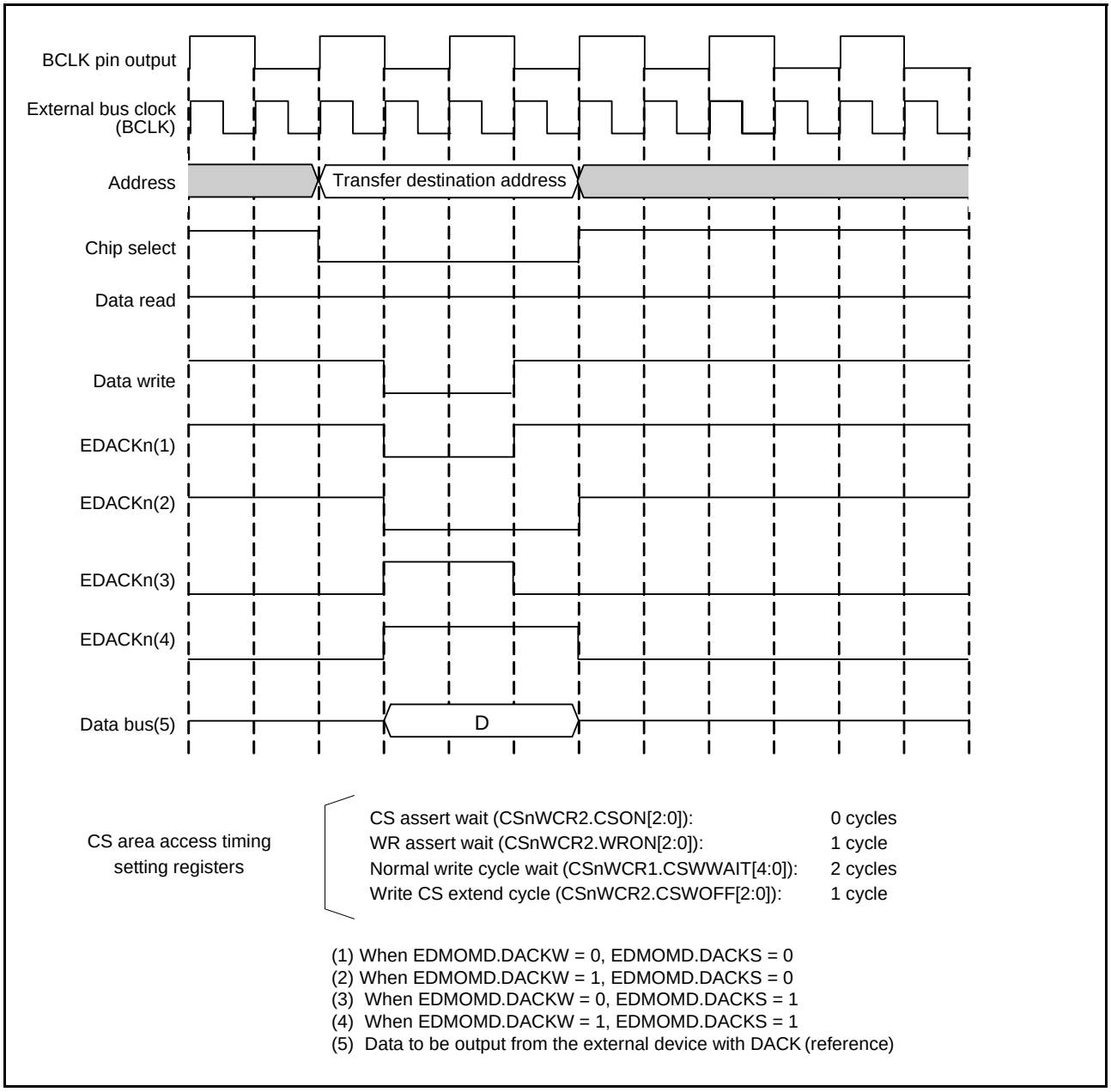


Figure 19.28 Operation Example in Normal-Transfer (CS Area Write) Single Address Mode

19.9.2 EDACKn Operation Example in Normal-Transfer (SDRAM Area) Single Address Mode

Figure 19.29 shows the operation example in which data is transferred from SDRAM to the device with EDACKn in normal transfer mode.

EDACKn is asserted while SDRAM is outputting data.

Figure 19.30 shows the operation example in which data is transferred from the device with EDACKn to SDRAM in normal transfer mode.

EDACKn is asserted while SDRAM is writing data.

For the SDRAM commands and SDRAM access timing setting registers, refer to section 16, Buses.

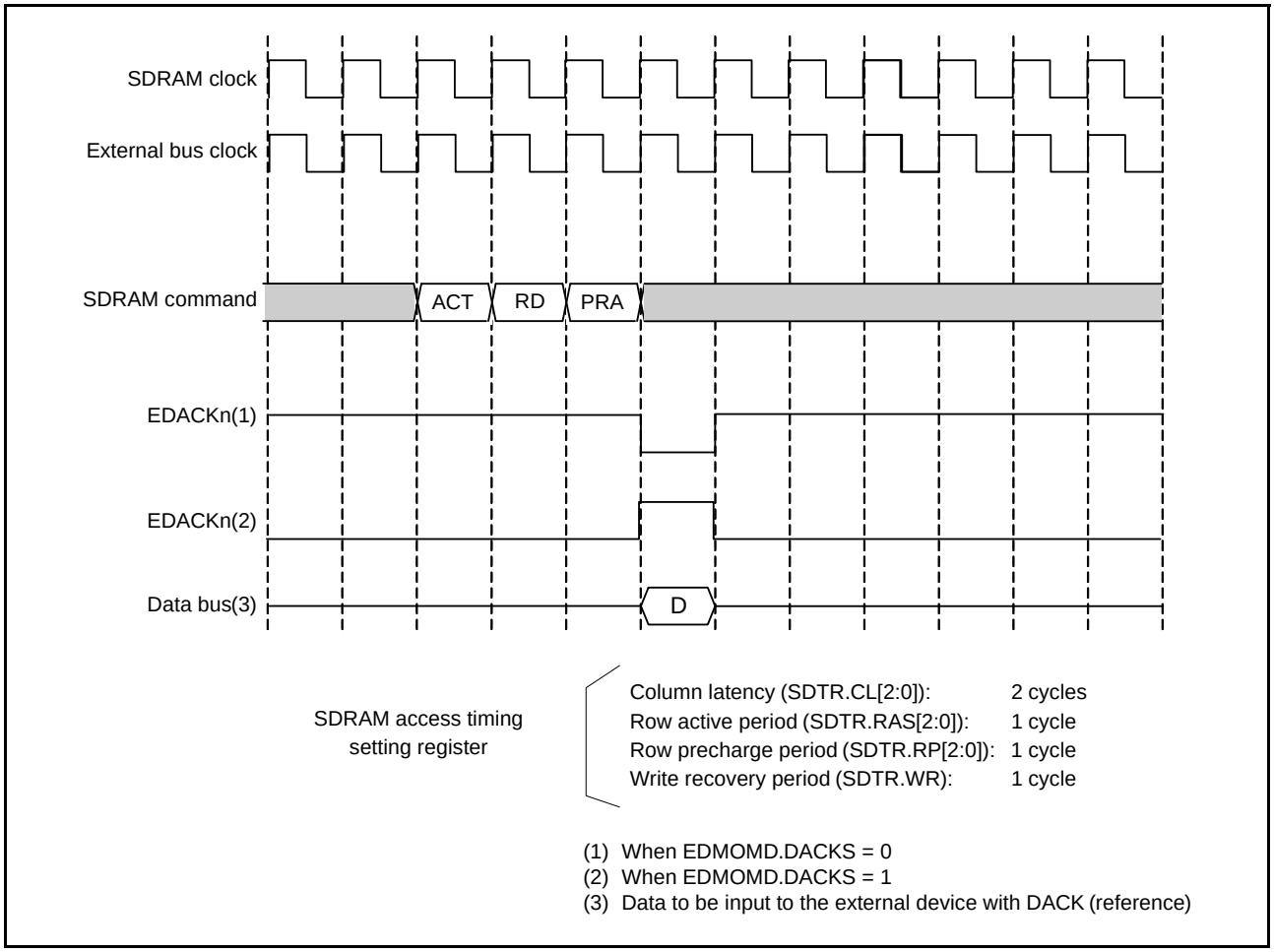


Figure 19.29 Operation Example in Normal-Transfer (SDRAM Area Read) Single Address Mode

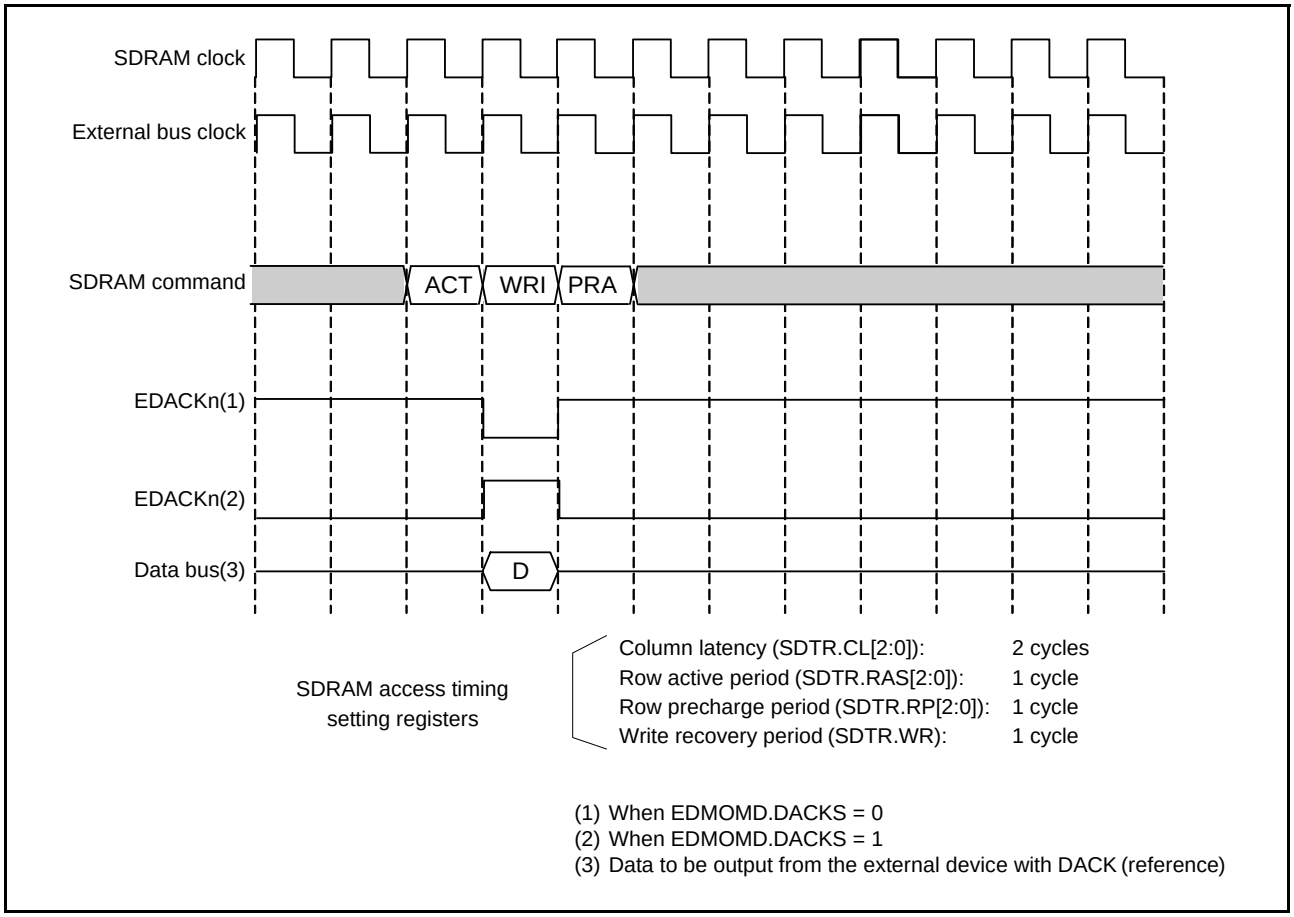


Figure 19.30 Operation Example in Normal-Transfer (SDRAM Area write) Single Address Mode

19.9.3 EDACKn Operation Example in Block-Transfer (CS Area) Single Address Mode

Figure 19.31 shows the operation example in which data is transferred from the CS area to the device with EDACKn in block transfer mode (block size = two). Setting the DACKW bit in EDMOMD of EXDMACn to 1 allows EDACKn to be negated one BCLK cycle before the data read signal is negated.

Figure 19.32 shows the operation example in which data is transferred from the device with EDACKn to the CS area in block transfer mode (block size = two). Setting the DACKW bit to 1 allows EDACKn to be negated one BCLK cycle after the data write signal is negated.

For the data read signal, data write signal, and CS area access timing setting registers, refer to section 16, Buses.

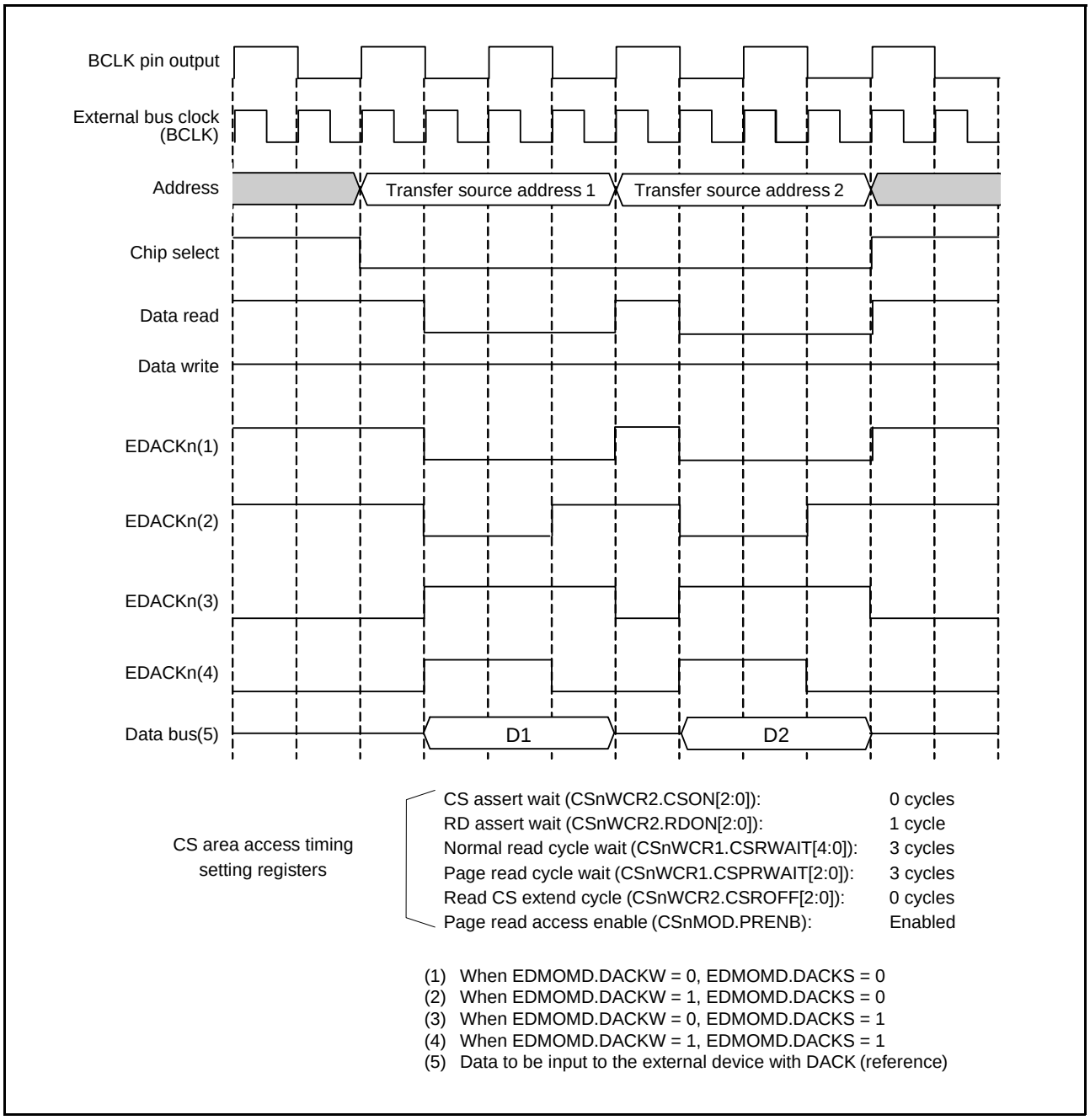


Figure 19.31 Operation Example in Block-Transfer (CS Area Read) Single Address Mode

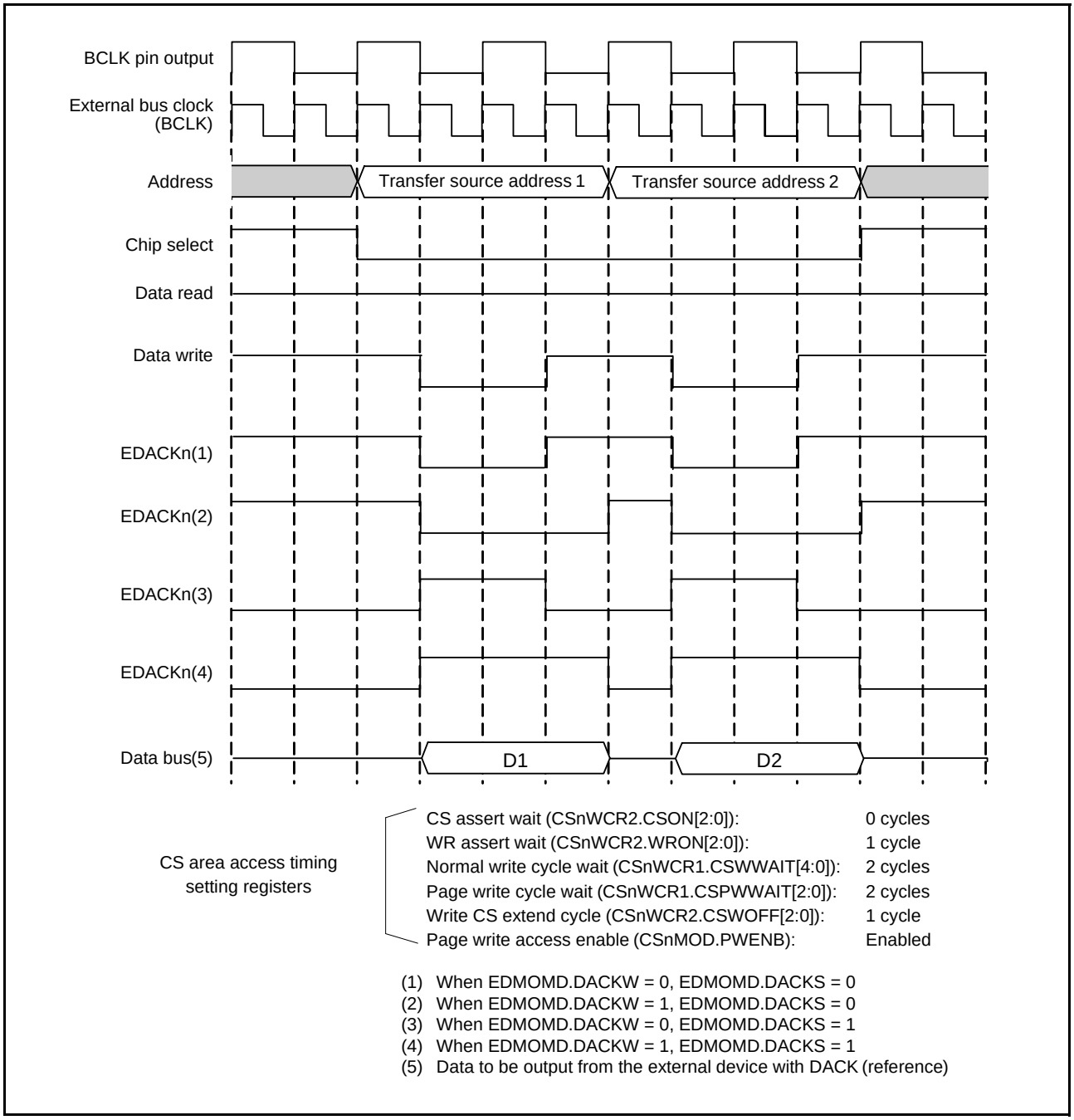


Figure 19.32 Operation Example in Block-Transfer (CS Area Write) Single Address Mode

19.9.4 EDACKn Operation Example in Block-Transfer (SDRAM Area) Single Address Mode

Figure 19.33 shows the operation example in which data is transferred from SDRAM to the device with EDACKn in block transfer mode (block size = four) when the SDRAM continuous access enable bit is enabled (SDAMOD.BE = 1) and EXDMACn.EDMOMD.DACKSEL set to 0.

EDACKn is asserted while SDRAM is outputting data.

Figure 19.34 shows the operation example in which data is transferred from the device with EDACKn to SDRAM in block transfer mode (block size = four) when the SDRAM continuous access enable bit is enabled (SDAMOD.BE = 1) and EXDMACn.EDMOMD.DACKSEL set to 0.

EDACKn is asserted while SDRAM is writing data.
Figure 19.35 shows the operation example in which data is transferred from SDRAM to the device with EDACKn in block transfer mode (block size = four) when the SDRAM continuous access enable bit is enabled (SDAMOD.BE = 1) and EXDMACn.EDMOMD.DACKSEL set to 1.
EDACKn is asserted while SDRAM is outputting data.
Figure 19.36 shows the operation example in which data is transferred from the device with EDACKn to SDRAM in block transfer mode (block size = four) when the SDRAM continuous access enable bit is enabled (SDAMOD.BE = 1) and EXDMACn.EDMOMD.DACKSEL set to 1.
EDACKn is asserted during SDRAM is writing dat.
Figure 19.37 shows the operation example in which data is transferred from SDRAM to the device with EDACKn in block transfer mode (block size = two) when the SDRAM continuous access enable bit is disabled (SDAMOD.BE = 0) and EXDMACn.EDMOMD.DACKSEL set to 0.
Figure 19.38 shows the operation example in which data is transferred from the device with EDACKn to SDRAM in block transfer mode (block size = two) when the SDRAM continuous access enable bit is disabled (SDAMOD.BE = 0) and EXDMACn.EDMOMD.DACKSEL set to 0.
EDACKn is asserted while SDRAM is writing data.
For the SDRAM commands and SDRAM access timing setting registers, refer to section 16, Buses.

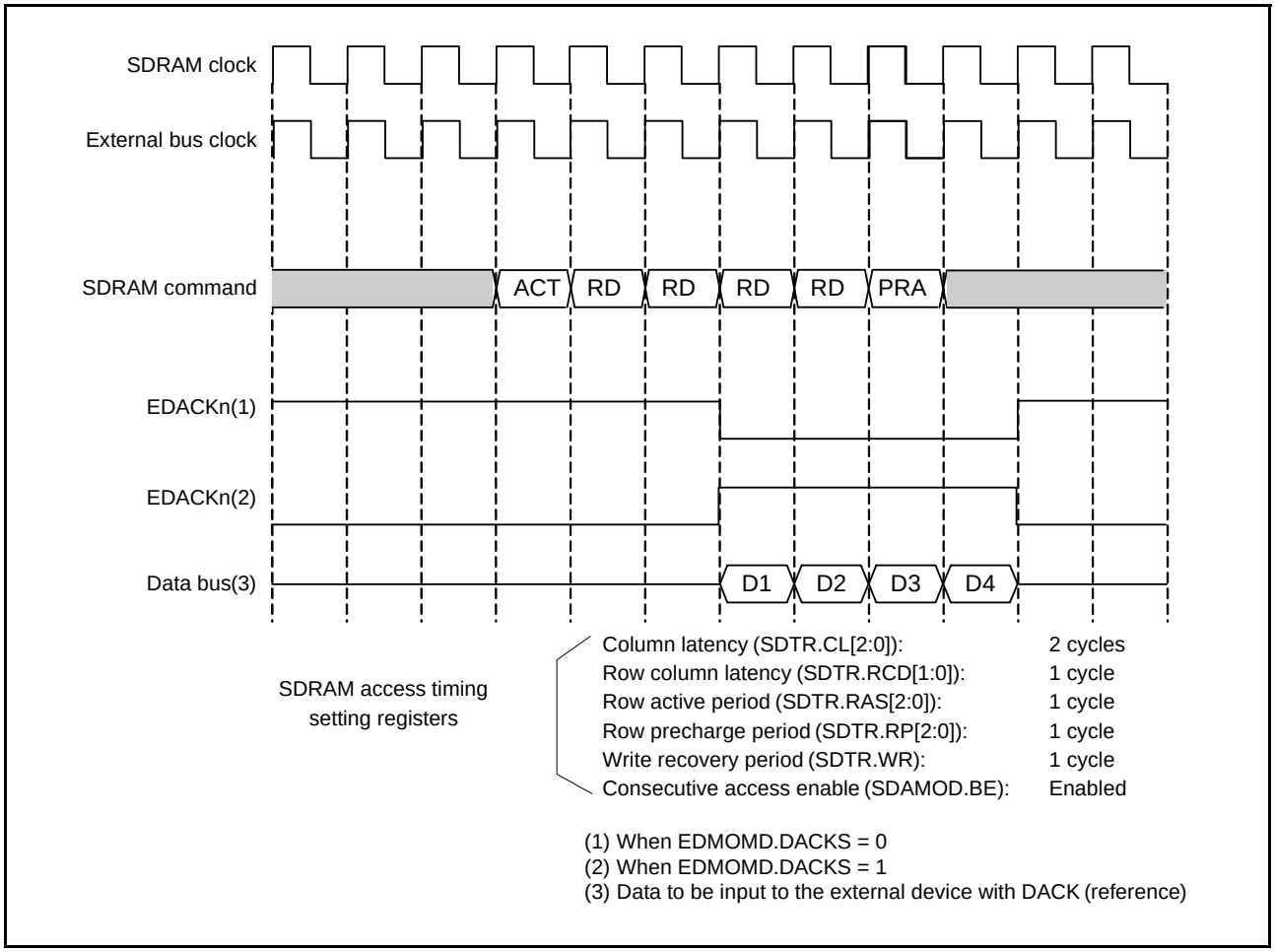


Figure 19.33 Operation Example in Block-Transfer (SDRAM Area Read: Consecutive Access Enabled, EXDMACn.EDMOMD.DACKSEL = 0) Single Address Mode

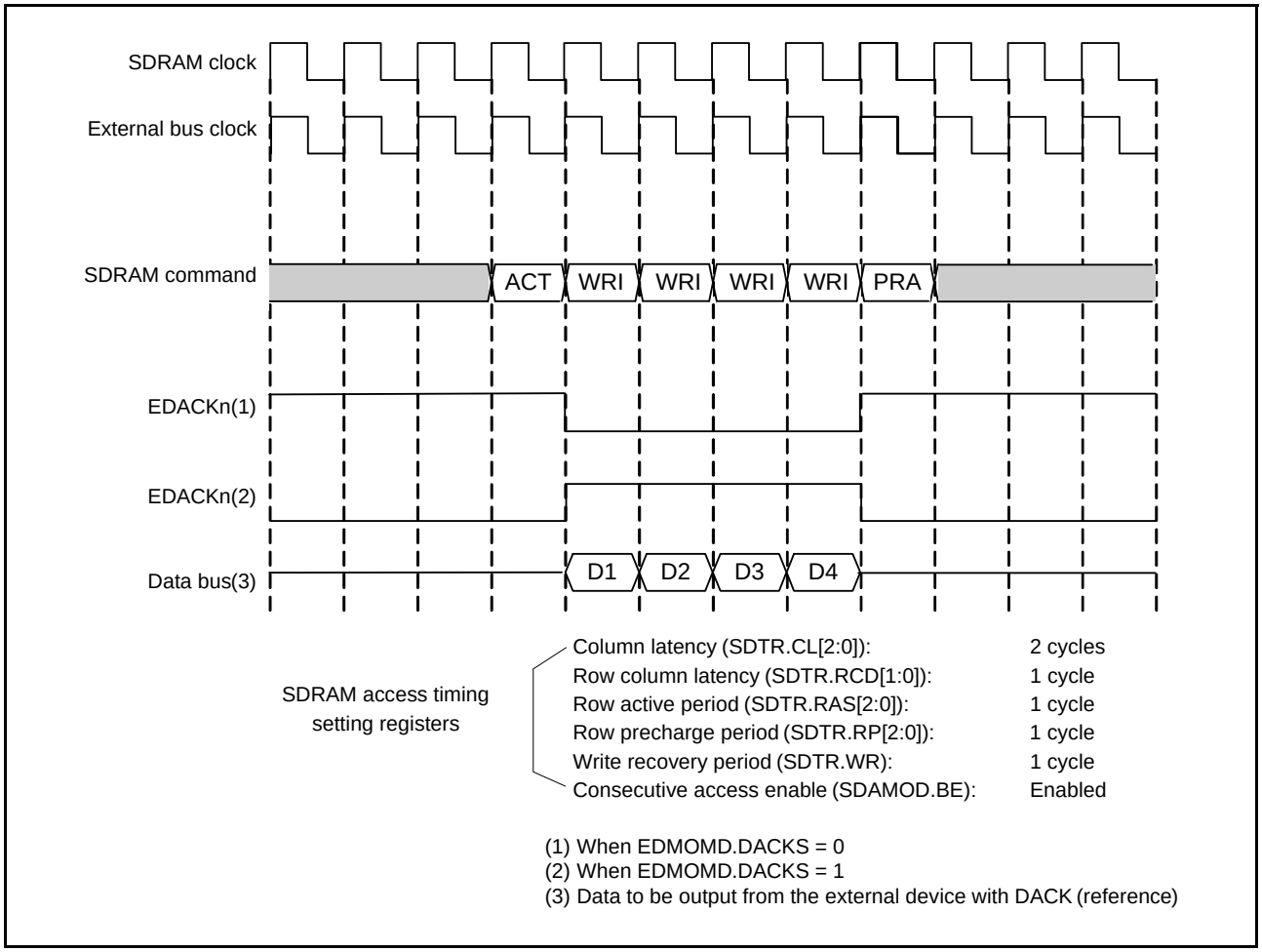


Figure 19.34 Operation Example in Block-Transfer (SDRAM Area Write: Consecutive Access Enabled, EXDMACn.EDMOMD.DACKSEL = 0) Single Address Mode

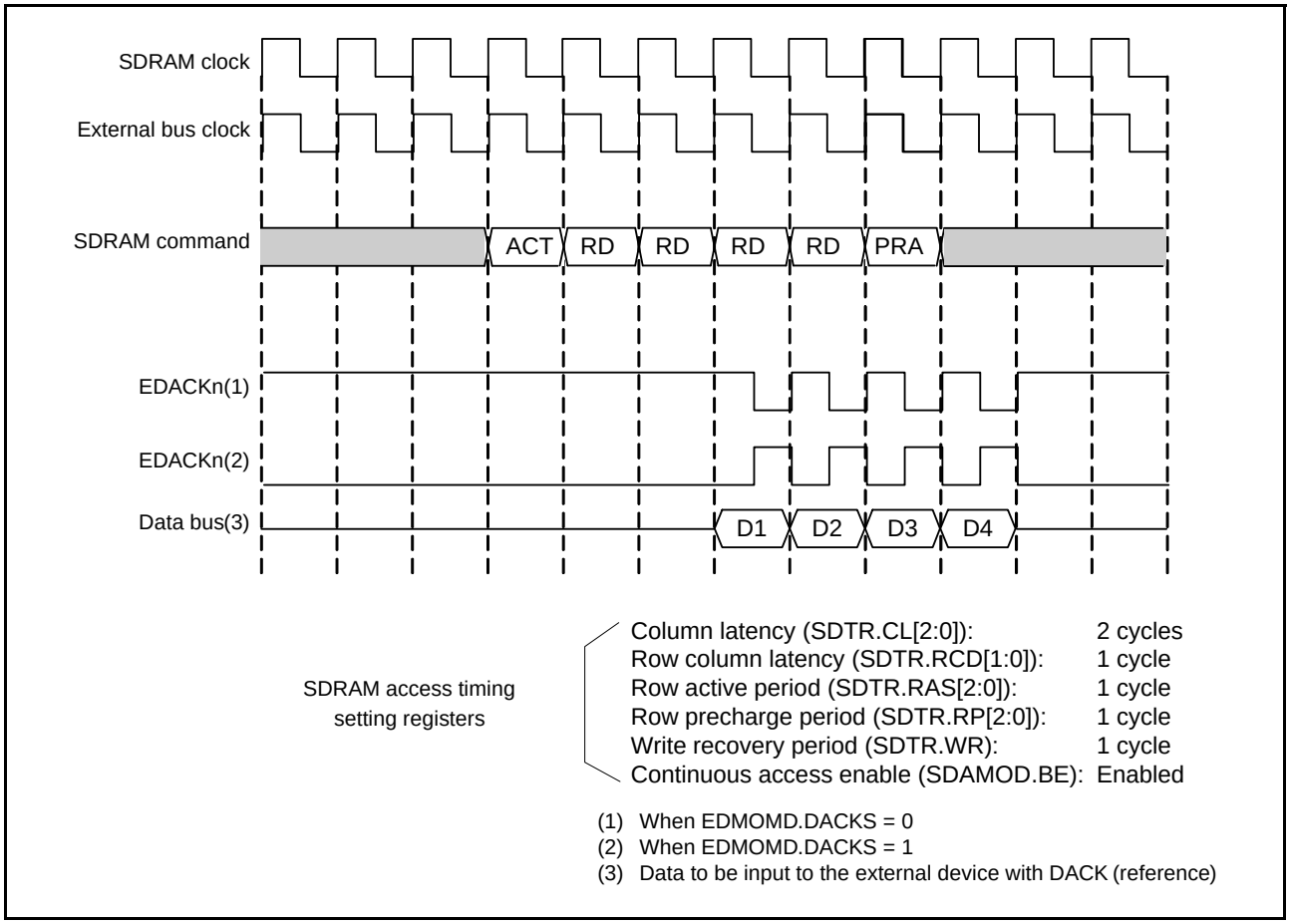


Figure 19.35 Operation Example in Block-Transfer (SDRAM Area Read: Continuous Access Enabled, EXDMACn.EDMOMD.DACKSEL = 1) Single Address Mode

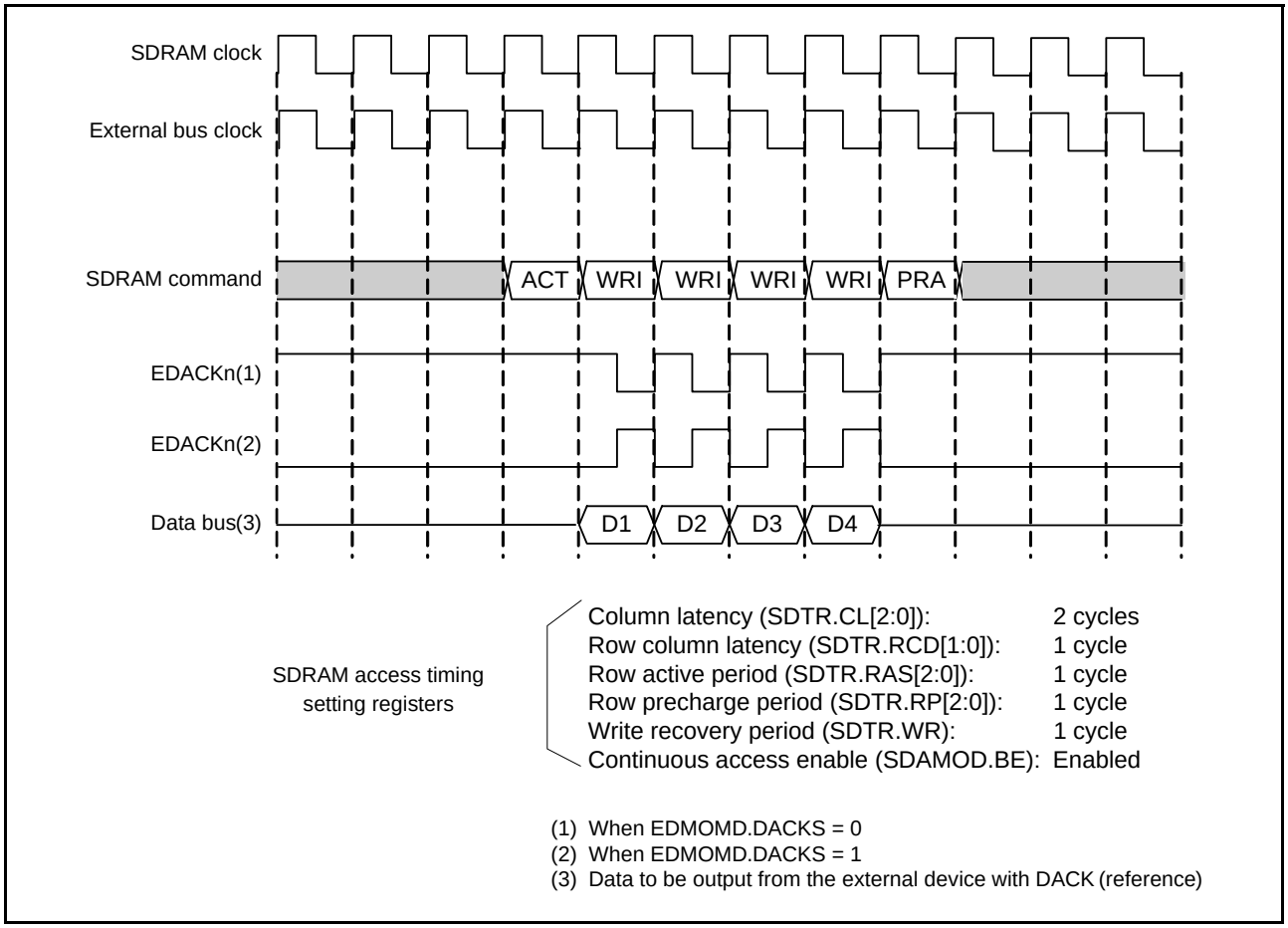


Figure 19.36 Operation Example in Block-Transfer (SDRAM Area Write: Continuous Access Enabled, EXDMACn.EDMOMD.DACKSEL = 1) Single Address Mode

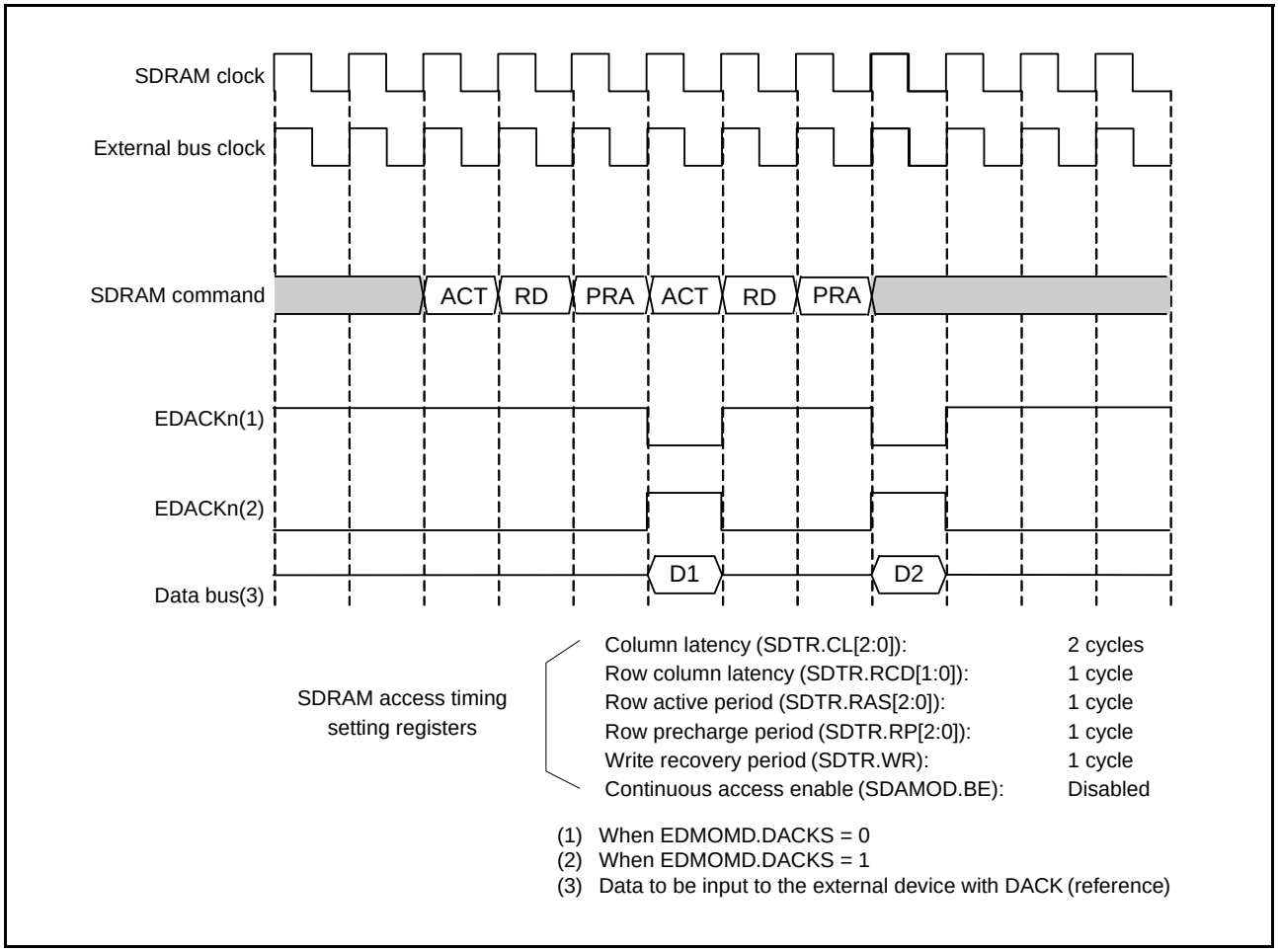


Figure 19.37 Operation Example in Block-Transfer (SDRAM Area Read: Continuous Access Disabled, EXDMACn.EDMOMD.DACKSEL = 0) Single Address Mode

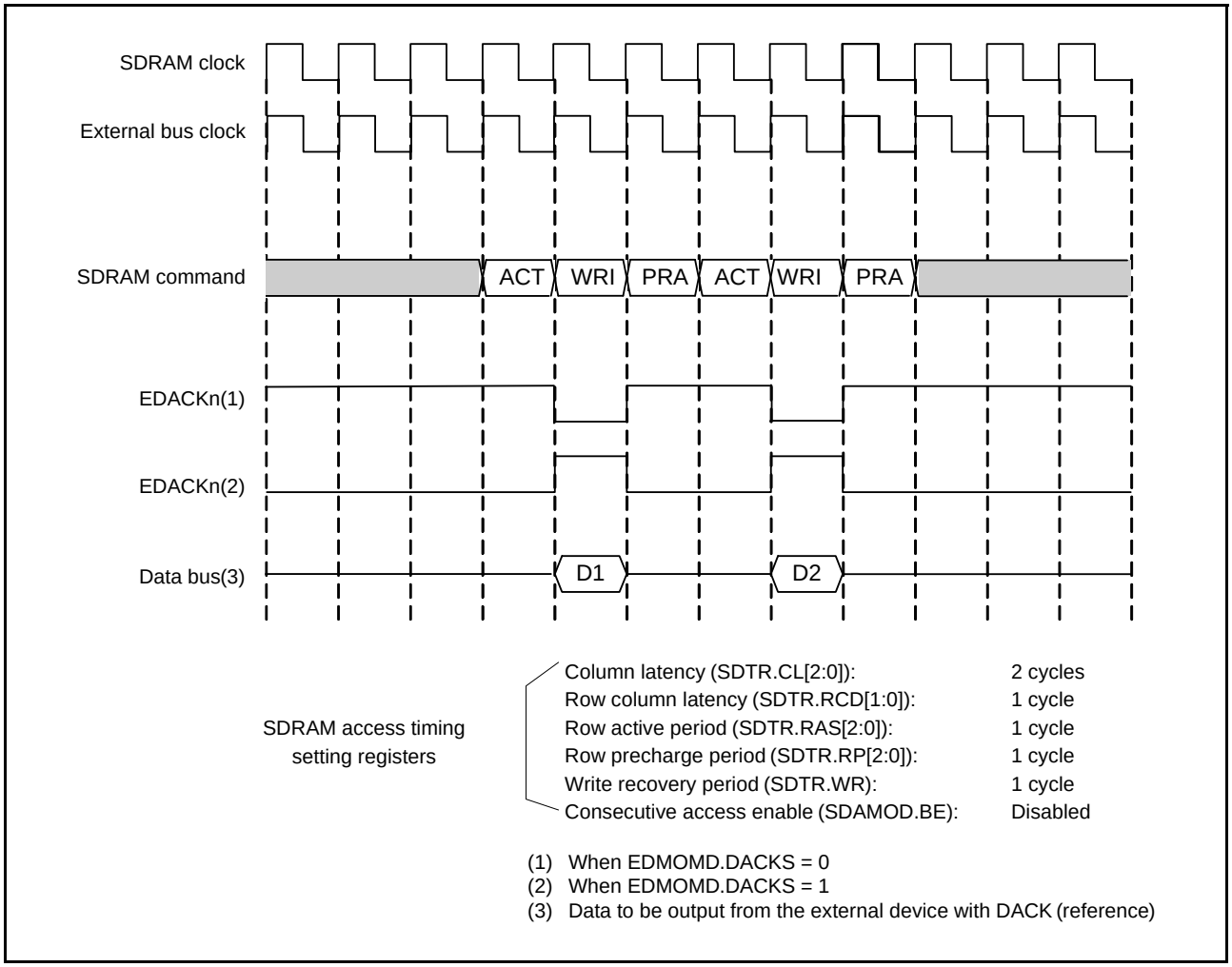


Figure 19.38 Operation Example in Block-Transfer (SDRAM Area Write: Consecutive Access Disabled) Single Address Mode

19.10 Usage Notes

19.10.1 Cluster Buffers

The EXDMAC provides eight 32-bit cluster buffers (CLSBR0 to CLSBR7), in which data is stored in the different manner depending on the transfer size setting (SZ[1:0] bits in EDMTMD of EXDMACn).
Figure 19.39 shows how data is stored in cluster buffers.

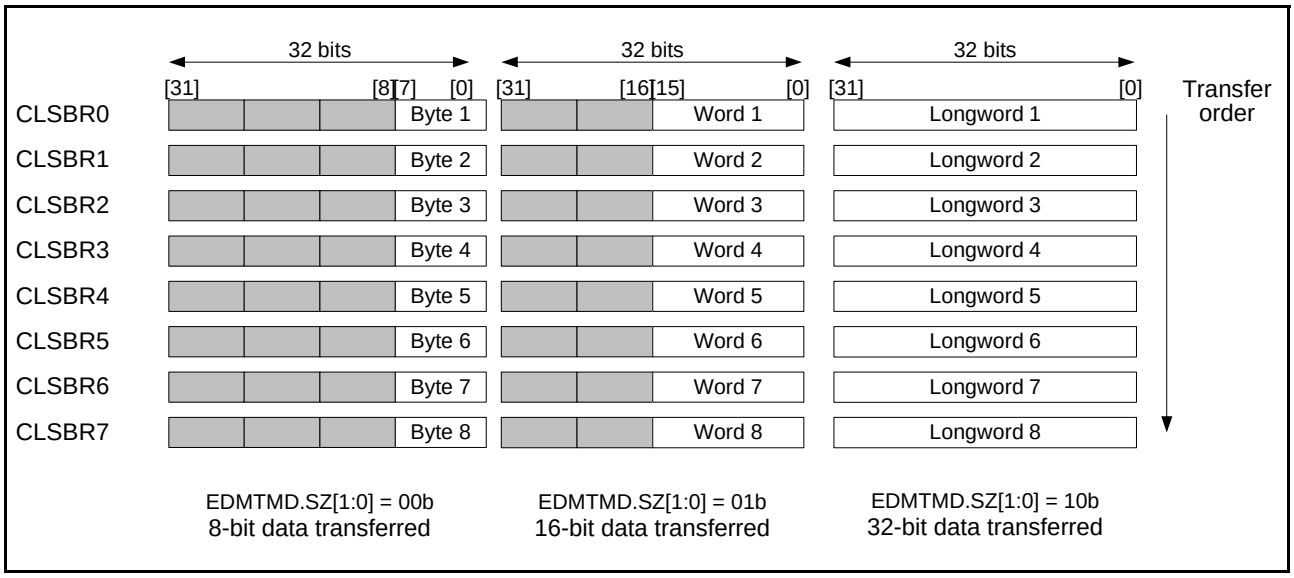


Figure 19.39 Data Storage in Cluster Buffers

19.10.2 Access to the Registers during DMA Transfer

Do not write to the EDMSAR, EDMDAR, EDMCRA, EDMCRB, EDMTMD, EDMOMD, EDMINT, EDMAMD, EDMOFR, and EDMRMD registers of EXDMACn while the EDMSTS.ACT flag of the same channel is set to 1 (DMA operating state) or the EDMCNT.DTE bit of the same channel is set to 1 (DMA transfer enabled).

19.10.3 DMA Transfer to Reserved Areas

DMA transfer to the reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on the reserved areas, see section 4, Address Space.

20. Data Transfer Controller (DTCa)

This MCU incorporates a data transfer controller (DTC).
The DTC is triggered by an interrupt request to perform data transfers.

20.1 Overview

Table 20.1 lists the specifications of the DTC, and Figure 20.1 shows a block diagram of the DTC.

Table 20.1 DTC Specifications

Item	Description
Number of transfer channels	• The same number as all interrupt sources that can start the DTC transfer.
Transfer modes	• Normal transfer mode A single transfer request leads to a single data transfer. • Repeat transfer mode A single transfer request leads to a single data transfer. The transfer address is returned to the transfer start address after the number of data transfers corresponding to “repeat size”. The maximum number of repeat transfers is 256, and the maximum data transfer size is 256 × 32 bits, 1024 bytes. • Block transfer mode A single transfer request leads to the transfer of a single block. The maximum block size is 256 × 32 bits = 1024 bytes.
Chain transfer	• Multiple types of data transfers can sequentially be executed in response to a single request. • Either “performed only when the transfer counter becomes 0” or “every time” can be selected.
Transfer space	• In short-address mode: 16 Mbytes (Areas from 0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh except reserved areas) • In full-address mode: 4 Gbytes (Area from 0000 0000h to FFFF FFFFh except reserved areas)
Data transfer units	• Single data: 1 byte (8 bits), 1 word (16 bits), 1 longword (32 bits) • Single block size: 1 to 256 data
CPU interrupt source	• An interrupt request can be generated to the CPU on a request source for a data transfer. • An interrupt request can be generated to the CPU after a single data transfer. • An interrupt request can be generated to the CPU after data transfer of specified volume.
Event link function	An event link request is generated after one data transfer (for block, after one block transfer).
Read skip	Reading of the transfer information can be skipped when the same transfer is repeated.
Write-back skip	Write-back of the transferred data that is not updated can be skipped when the address of the transfer source or destination is fixed.
Low power consumption function	Module stop state can be set.

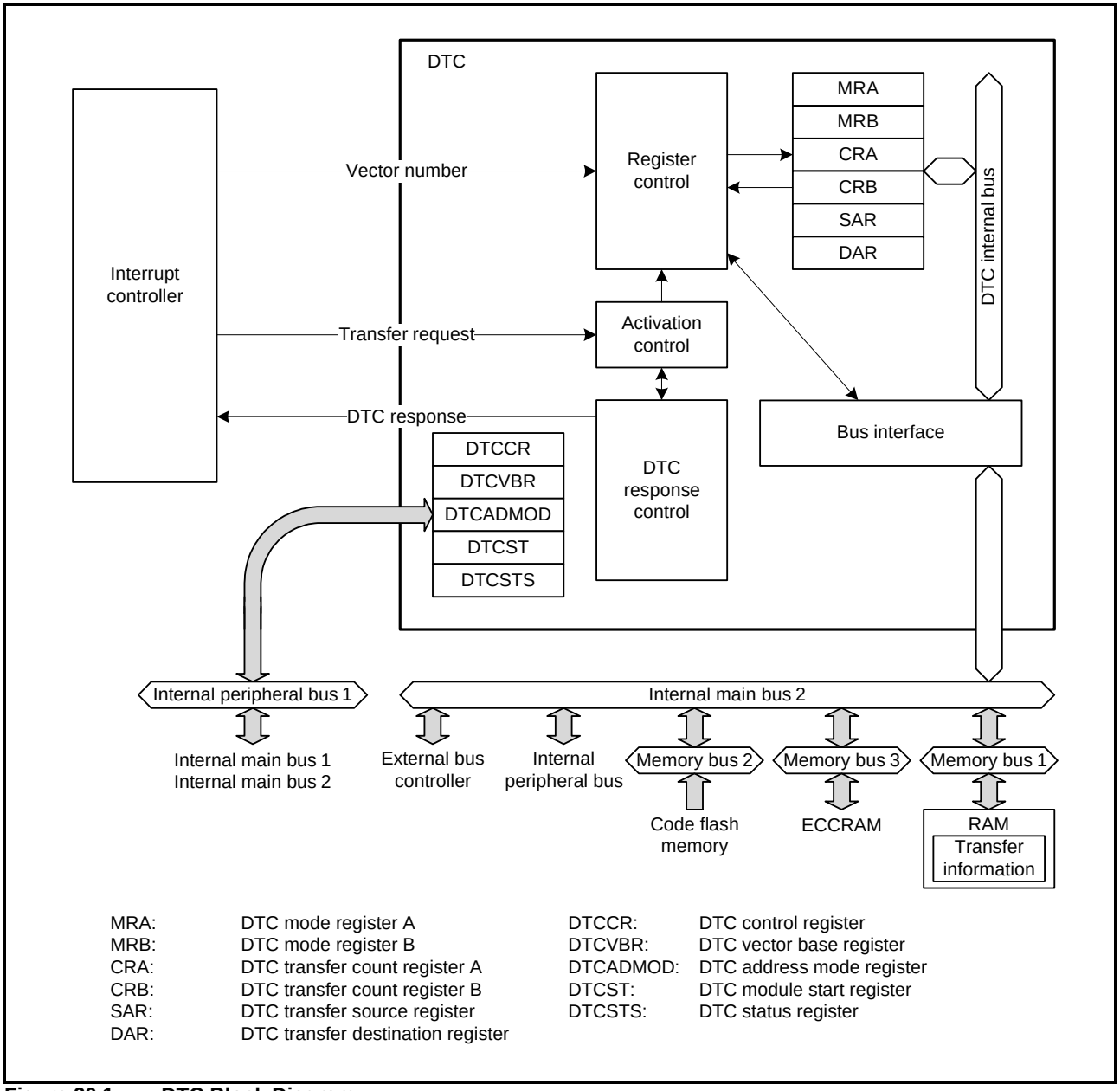


Figure 20.1 DTC Block Diagram

20.2 Register Descriptions

Registers MRA, MRB, SAR, DAR, CRA, and CRB are DTC internal registers, which cannot be directly accessed from the CPU. Values to be set in these DTC internal registers are placed in the RAM area as transfer information. When accepting a transfer request, the DTC reads the transfer information from the RAM area and sets it in the internal registers. After the data transfer ends, the values of the updated internal register are written back to the RAM area as transfer information.

20.2.1 DTC Mode Register A (MRA)

Address(es): (inaccessible directly from the CPU)

	b7	b6	b5	b4	b3	b2	b1	b0
	MD[1:0]		SZ[1:0]		SM[1:0]		—	—
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	Set these bits to 0.	—
b3, b2	SM[1:0]	Transfer Source Address Addressing Mode	b3 b2 0 0: The address in the SAR register is fixed. (write-back to SAR is skipped.) 0 1: The address in the SAR register is fixed. (write-back to SAR is skipped.) 1 0: The SAR value is incremented after a data transfer. (+1 when the SZ[1:0] bits are 00b, +2 when 01b, +4 when 10b) 1 1: The SAR value is decremented after a data transfer. (–1 when the SZ[1:0] bits are 00b, –2 when 01b, –4 when 10b)	—
b5, b4	SZ[1:0]	DTC Data Transfer Size	b5 b4 0 0: Byte (8-bit) transfer 0 1: Word (16-bit) transfer 1 0: Longword (32-bit) transfer 1 1: Setting prohibited	—
b7, b6	MD[1:0]	DTC Transfer Mode Select	b7 b6 0 0: Normal transfer mode 0 1: Repeat transfer mode 1 0: Block transfer mode 1 1: Setting prohibited	—

MRA register is used to select the DTC operating mode and cannot be accessed directly from the CPU.

20.2.2 DTC Mode Register B (MRB)

Address(es): (inaccessible directly from the CPU)

	b7	b6	b5	b4	b3	b2	b1	b0
	CHNE	CHNS	DISEL	DTS	DM[1:0]		—	—
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	Set these bits to 0.	—
b3, b2	DM[1:0]	Transfer Destination Address Addressing Mode	b3 b2 0 0: The address in the DAR register is fixed. (Write-back to DAR is skipped.) 0 1: The address in the DAR register is fixed. (Write-back to DAR is skipped.) 1 0: The DAR value is incremented after data transfer. (+1 when the MRA.SZ[1:0] bits are 00b, +2 when 01b, +4 when 10b) 1 1: The DAR value is decremented after data transfer. (–1 when the SZ[1:0] bits are 00b, –2 when 01b, –4 when 10b)	—
b4	DTS	DTC Transfer Mode Select	0: Transfer destination side is repeat area or block area. 1: Transfer source side is repeat area or block area.	—
b5	DISEL	DTC Interrupt Select	0: An interrupt request to the CPU is generated on completion of the specified number of data transfers. 1: An interrupt request to the CPU is generated for each data transfer.	—
b6	CHNS	DTC Chain Transfer Select	0: Chain transfer is performed on completion of each transfer. 1: Chain transfer is performed only when the transfer counter is changed from 1 to 0 or 1 to CRAH.	—
b7	CHNE	DTC Chain Transfer Enable	0: Chain transfer is disabled. 1: Chain transfer is enabled.	—

MRB register is used to select the DTC operating mode and cannot be accessed directly from the CPU.

DTS Bit (DTC Transfer Mode Select)

The DTS bit specifies the side (transfer source or destination) to be a repeat area or block area in repeat transfer mode or block transfer mode.

CHNS Bit (DTC Chain Transfer Select)

The CHNS bit selects the chain transfer condition.

When the CHNE bit is 0, setting of the CHNS bit is ignored. For details on the conditions to select the chain transfer, refer to Table 20.3, Chain Transfer Conditions.

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the interrupt status flag for the request source is not cleared, and an interrupt request to the CPU is not generated.

CHNE Bit (DTC Chain Transfer Enable)

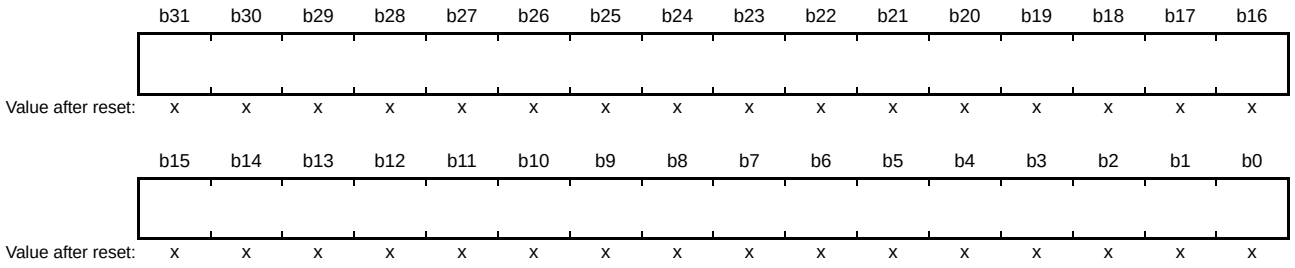
The CHNE bit enables or disables chain transfer.

The chain transfer condition is selected by the CHNS bit.

For details of chain transfer, refer to section 20.4.6, Chain Transfer.

20.2.3 DTC Transfer Source Register (SAR)

Address(es): (inaccessible directly from the CPU)

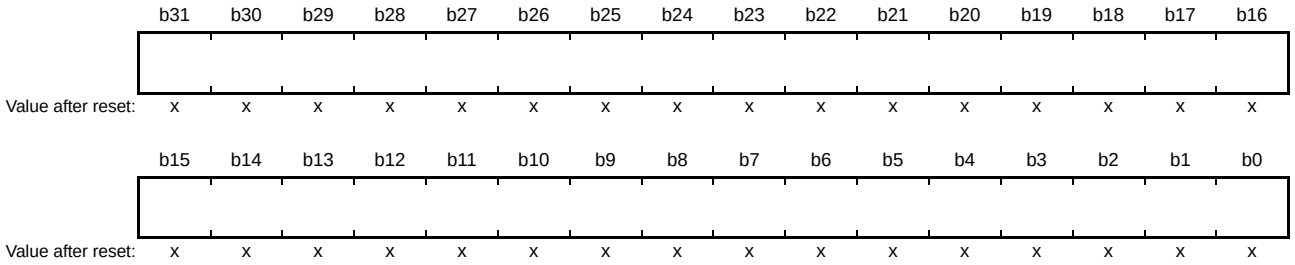


x: Undefined

SAR register is used to set the transfer source start address.
In full-address mode, 32 bits are valid.
In short-address mode, lower 24 bits are valid and upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.
SAR register cannot be accessed directly from the CPU.

20.2.4 DTC Transfer Destination Register (DAR)

Address(es): (inaccessible directly from the CPU)



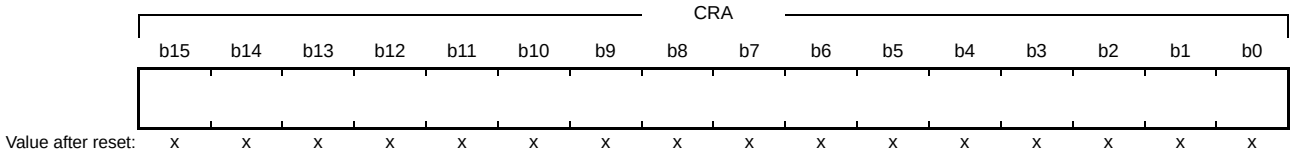
x: Undefined

DAR register is used to set the transfer destination start address.
In full-address mode, 32 bits are valid.
In short-address mode, lower 24 bits are valid and upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.
DAR register cannot be accessed directly from the CPU.

20.2.5 DTC Transfer Count Register A (CRA)

- Normal transfer mode

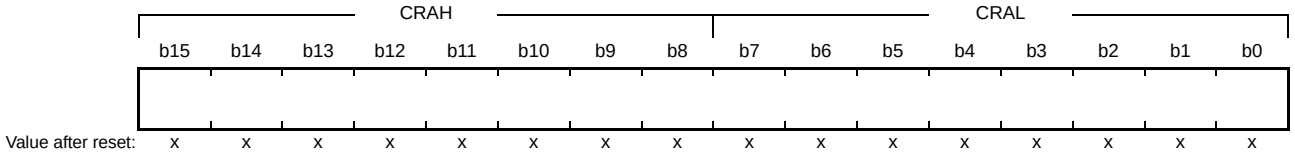
Address(es): (inaccessible directly from the CPU)



x: Undefined

- Repeat transfer mode/block transfer mode

Address(es): (inaccessible directly from the CPU)



x: Undefined

Symbol	Register Name	Description	R/W
CRAL	Transfer Counter A Lower Register	Set transfer count. This register functions as a transfer counter during data transfer.	—
CRAH	Transfer Counter A Upper Register	Set transfer count. This register functions as a reload register during data transfer.	—

Note: The function depends on transfer mode.
Note: Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

This register is for counting the number of transfers and cannot be accessed directly from the CPU.

(1) Normal transfer mode (MRA.MD[1:0] bits = 00b)

CRA register functions as a 16-bit transfer counter in normal transfer mode.
The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively.
The CRA value is decremented (–1) at each data transfer.

(2) Repeat transfer mode (MRA.MD[1:0] bits = 01b)

The CRAH register retains the transfer count and the CRAL register functions as an 8-bit transfer counter.
The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.
The CRAL value is decremented (–1) at each data transfer. When it reaches 00h, the CRAH value is reloaded to the CRAL register.

(3) Block transfer mode (MRA.MD[1:0] bits = 10b)

The CRAH register retains the block size and the CRAL register functions as an 8-bit block size counter.
The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.
The CRAL value is decremented (–1) at each data transfer. When it reaches 00h, the CRAH value is reloaded to the CRAL register.

20.2.6 DTC Transfer Count Register B (CRB)

Address(es): (inaccessible directly from the CPU)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

CRB register is used to set the block transfer count for block transfer mode and cannot be accessed directly from the CPU.
The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively.
The CRB value is decremented (–1) when the final data of a single block size is transferred.
When normal transfer mode or repeat transfer mode is selected, this register is not used and the set value is ignored.

20.2.7 DTC Control Register (DTCCR)

Address(es): DTC.DTCCR 0008 2400h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	RRS	—	—	—	—
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b4	RRS	DTC Transfer Information Read Skip Enable	0: Transfer information read is not skipped. 1: Transfer information read is skipped when vector numbers match.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DTCCR register is used to control the DTC operation.

RRS Bit (DTC Transfer Information Read Skip Enable)

The DTC vector number is compared with the vector number in the previous data transfer.
When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transferred information. However, when the previous transfer was chain transfer, the transferred information is read regardless of the value of the RRS bit.
Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, the transferred information is read regardless of the RRS bit value.

20.2.8 DTC Vector Base Register (DTCVBR)

Address(es): DTC.DTCVBR 0008 2404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DTCVBR register is used to set the base address for calculating the address to which the DTC vector is allocated. Writing to the upper 4 bits (b31 to b28) is ignored, and the address of this register is extended by the value specified by b27. The lower 10 bits are reserved and the values are fixed to 0. Write 0 to the lower 10 bits if necessary.
It can be set in the range of 0000 0000h to 07FF FC00h and F800 0000h to FFFF FC00h in 1-Kbyte units.

20.2.9 DTC Address Mode Register (DTCADMOD)

Address(es): DTC.DTCADMOD 0008 2408h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SHORT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SHORT	Short-Address Mode Set	0: Full-address mode 1: Short-address mode	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DTCADMOD register is used to specify the area accessible by the DTC.

SHORT Bit (Short-Address Mode Set)

This bit is used to select address mode of registers SAR and DAR.
Full-address mode allows the DTC to access to a 4-Gbyte space (0000 0000h to FFFF FFFFh).
Short-address mode allows the DTC to access to a 16-Mbyte space (0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh).

20.2.10 DTC Module Start Register (DTCST)

Address(es): DTC.DTCST 0008 240Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTCST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DTCST	DTC Module Start	0: DTC module stop 1: DTC module start	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DTCST Bit (DTC Module Start)

Set the DTCST bit to 1 to enable the DTC to accept transfer requests. When this bit is set to 0, transfer requests are no longer accepted.

If this bit is set to 0 during data transfer, the accepted transfer request is active until the processing is completed.

Set the DTCST bit to 0 before making a transition to the module stop state, all-module clock stop mode, software standby mode, or deep software standby mode.

Set the DTCST bit to 1 to resume the data transfer after returning from the module stop state, all-module clock stop mode, or software standby mode.

For details on transitions to the module stop state, all-module clock stop mode, software standby mode, and deep software standby mode, refer to section 20.9, Low Power Consumption Function, and section 11, Low Power Consumption.

20.2.11 DTC Status Register (DTCSTS)

Address(es): DTC.DTCSTS 0008 240Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ACT	—	—	—	—	—	—	—	VECN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	VECN[7:0]	DTC Active Vector Number Monitoring Flag	These bits indicate the vector number for the request source when data transfer is in progress. The value is only valid if data transfer is in progress (the value of the ACT flag is 1).	R
b14 to b8	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b15	ACT	DTC Active Flag	0: Data transfer is not in progress. 1: Data transfer is in progress.	R

VECN[7:0] Flags (DTC Active Vector Number Monitoring Flag)

While data transfer is in progress, these bits indicate the vector number corresponding to the request source for the transfer.

When the DTCSTS register is read, the value of the VECN[7:0] flags is valid if the value of the ACT flag was 1 (data transfer is in progress) and invalid if the value of the ACT flag was 0 (data transfer is not in progress).

For the correspondence between the DTC request sources and the vector addresses, refer to Table 15.5, Interrupt Vector Table in section 15, Interrupt Controller (ICUA).

ACT Flag (DTC Active Flag)

This flag indicates the state of data transfer operation.

[Setting condition]

- When the data transfer is started by a transfer request.

[Clearing condition]

- When the data transfer is completed in response to a transfer request.

20.3 Request Sources

The DTC data transfer is triggered by an interrupt request. Setting the ICU.DTCERn.DTCE bit (n = interrupt vector number) to 1 selects the corresponding interrupt request as a request source for the DTC.
For the correspondence between the DTC request sources and the vector addresses, refer to Table 15.5, Interrupt Vector Table in section 15, Interrupt Controller (ICUA). For request by software, refer to section 15.2.5, Software Interrupt Generation Register (SWINTR) and section 15.2.6, Software Interrupt 2 Generation Register (SWINT2R) in section 15, Interrupt Controller (ICUA).
Once the DTC has accepted a transfer request, it does not accept another transfer request until transfer for that single request is completed, regardless of the priority of the requests.
When multiple transfer requests are generated during data transfer by the DMAC/DTC, the request with the highest priority on completion of the current transfer is accepted. When multiple transfer requests are generated while the DTCST.DTCST bit is 0 (DTC module stop), the request with the highest priority at the moment when the bit is subsequently set to 1 (DTC module start) is accepted.

The DTC performs the following operations at the start of a single data transfer (or the last of the consecutive transfers in the case of a chain transfer).

- On completion of a specified number of data transfer, the ICU.DTCERn.DTCE bit is set to 0 and an interrupt is requested to the CPU.
- If the MRB.DISEL bit is 1, an interrupt is requested to the CPU on completion of data transfer.
- For the other transfers, the interrupt status flag of the request source is set to 0 at the start of data transfer.

20.3.1 Allocating Transfer Information and DTC Vector Table

The DTC reads the start address of the transfer information corresponding to each request source from the vector table and reads the transfer information starting at that address.
The vector table should be located so that the lower 10 bits of the base address (start address) are 0. Use the DTC vector base register (DTCVBR) to set the base address of the DTC vector table.
Transfer information is allocated in the RAM area. The start address of the transfer information n with vector number n should be allocated at DTCVBR + 4n.
Transfer information should be aligned on a 4-byte boundary. The size of a transfer information is 12 bytes in short-address mode or 16 bytes in full-address mode. Use the DTCADMOD.SHORT bit to select short-address mode (SHORT bit = 1) or full-address mode (SHORT bit = 0).
Figure 20.2 shows the relationship between the DTC vector table and transfer information.
Figure 20.3 shows the allocation of transfer information in the RAM area. The lower addresses vary according to the endian of the corresponding allocation area. For details, refer to section 20.10.2, Allocating Transfer Information.

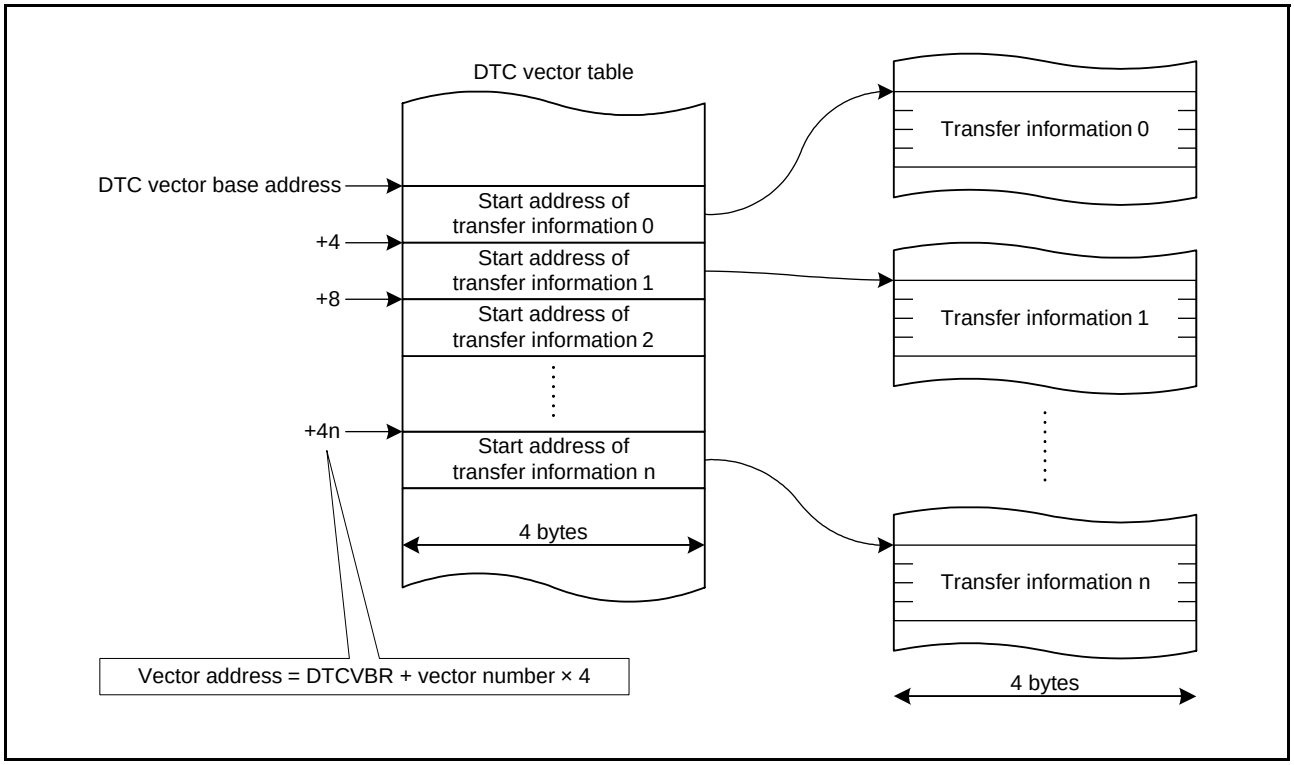


Figure 20.2 DTC Vector Table and Transfer Information

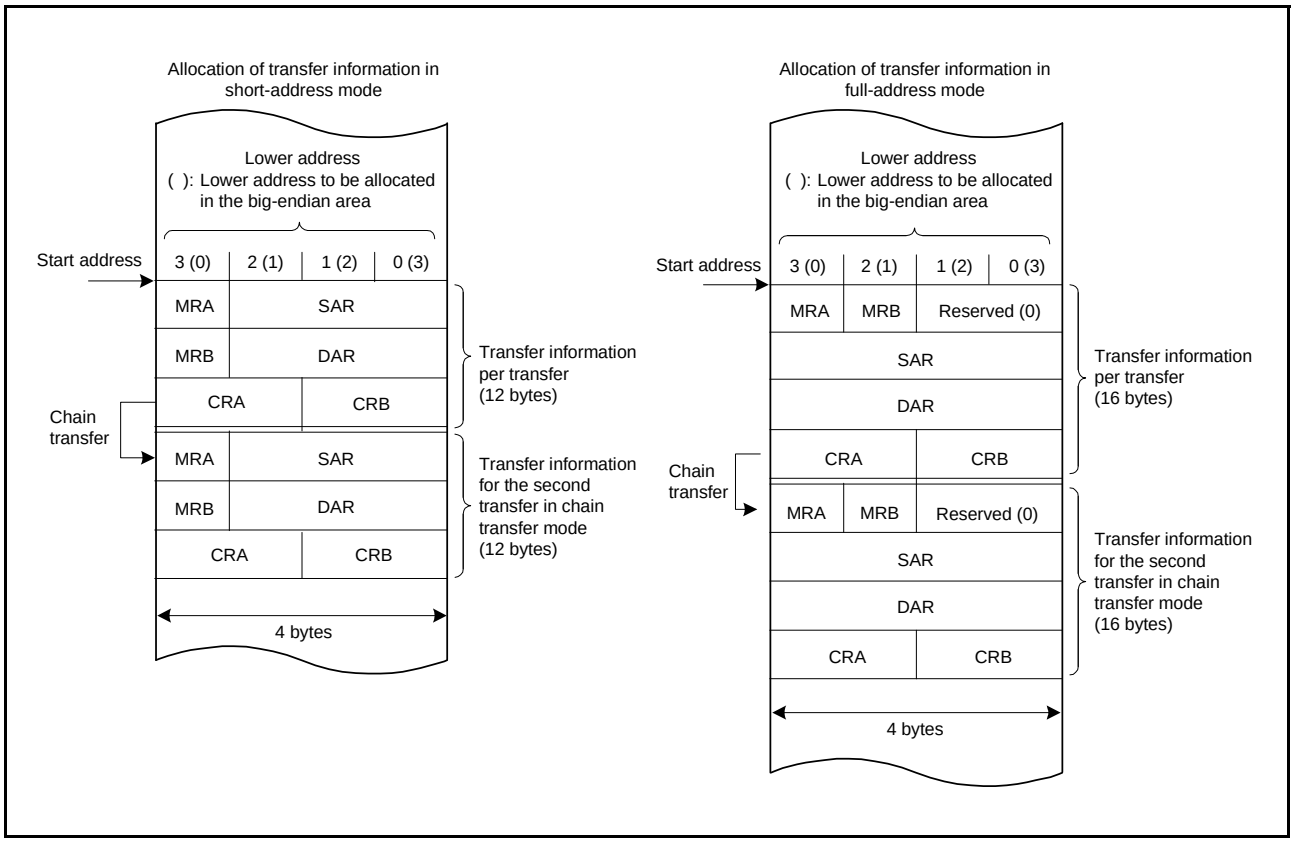


Figure 20.3 Allocation of Transfer Information in the RAM Area

20.4 Operation

The DTC transfers data in accordance with the transfer information. Storage of the transfer information in the RAM area is required before DTC operation.

When the DTC accepts a transfer request, it reads the DTC vector corresponding to the vector number. Next, the DTC reads transfer information from the address pointed by the DTC vector, transfers data, and then writes back the transfer information after the data transfer. Allocating transfer information in the RAM area allows data transfer of arbitrary number of channels.

There are three transfer modes: normal transfer mode, repeat transfer mode, and block transfer mode.

Set a transfer source address in the SAR register and a transfer destination address in the DAR register. The SAR and DAR registers are updated after the transfer according to the respective settings (increment, decrement, or fixed).

Table 20.2 lists transfer modes of the DTC.

Table 20.2 Transfer Modes of the DTC

Transfer Mode	Data Size Transferred on Single Transfer Request	Increment/Decrement of Memory Address	Settable Transfer Count
Normal transfer mode	1 byte/1 word/1 longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 65536
Repeat transfer mode*1	1 byte/1 word/1 longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 256*3
Block transfer mode*2	Block size specified in CRAH (1 to 256 bytes/1 to 256 words/1 to 256 longwords)	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 65536

Note 1. Set transfer source or transfer destination in the repeat area.
Note 2. Set transfer source or transfer destination in the block area.
Note 3. After data transfer of the specified count, the initial state is restored and the operation is continued (repeated).

Setting the MRB.CHNE bit to 1 allows multiple transfers (chain transfer) on a single transfer request. The setting in combination with the MRB.CHNS bit enables a chain transfer when the specified number of data transfers is completed. Figure 20.4 shows the operation flowchart of the DTC. Table 20.3 lists chain transfer conditions.

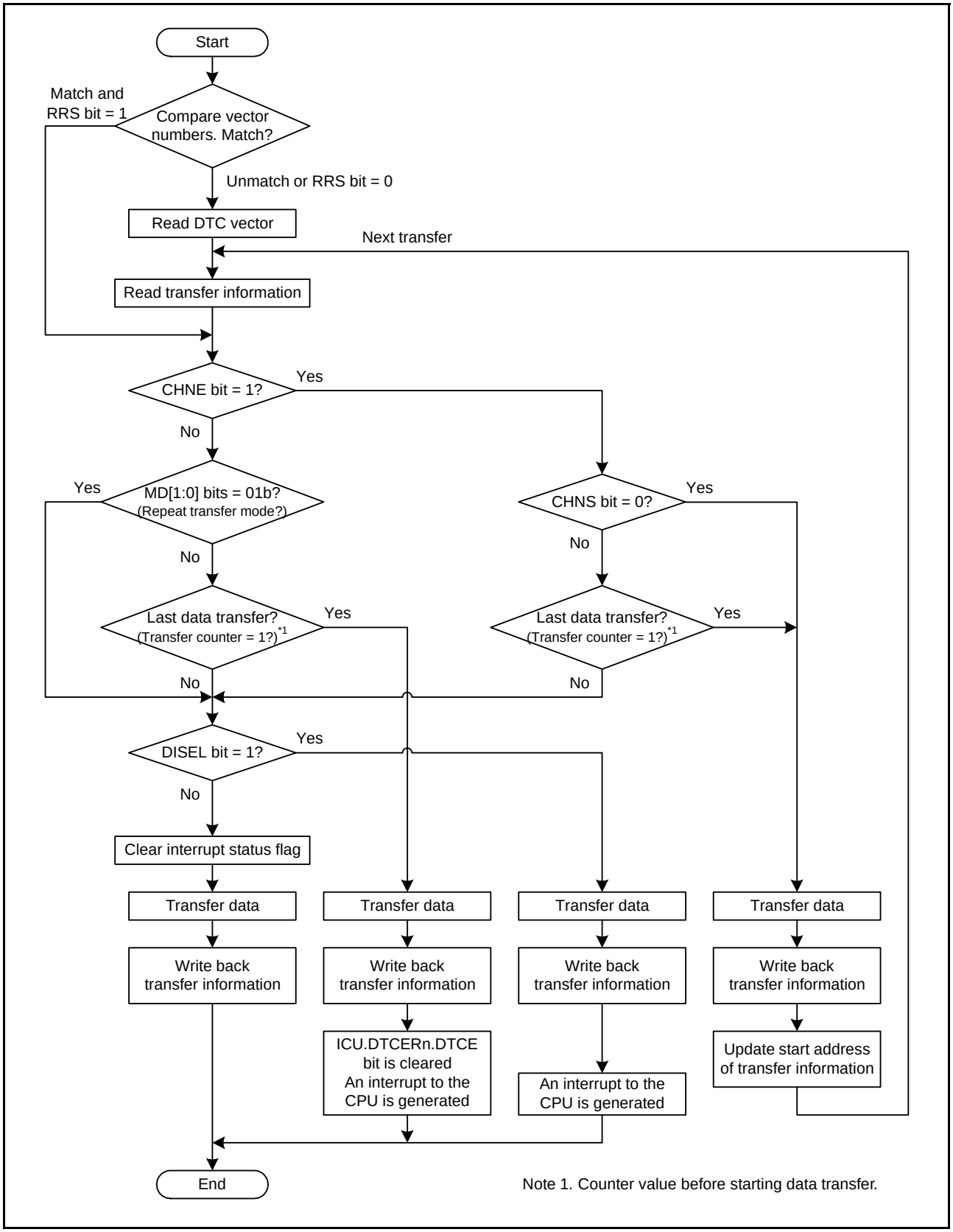


Figure 20.4 Operation Flowchart of the DTC

Table 20.3 Chain Transfer Conditions

First Transfer				Second Transfer*3				
CHNE Bit	CHNS Bit	DISEL Bit	Transfer Counter*1,*2	CHNE Bit	CHNS Bit	DISEL Bit	Transfer Counter*1,*2	Data Transfer
0	—	0	Other than (1 → 0)	—	—	—	—	Ends after the first transfer
0	—	0	(1 → 0)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU
0	—	1	—	—	—	—	—	
1	0	—	—	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	0	Other than (1 → ∗)	—	—	—	—	Ends after the first transfer
1	1	—	(1 → ∗)	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	1	Other than (1 → ∗)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU

Note 1. The transfer counters used depend on transfer modes as follows:
Normal transfer mode: CRA register
Repeat transfer mode: CRAL register
Block transfer mode: CRB register

Note 2. On completion of data transfer, the counters operate as follows:
1 → 0 in normal and block transfer modes
1 → CRAH in repeat transfer mode
(1 → ∗) in the table indicates both of the two operations above.

Note 3. Chain transfer can be selected for the second or subsequent transfers. The condition combination of “second transfer and the CHNE bit is 1” is omitted.

20.4.1 Transfer Information Read Skip Function

Reading of DTC vector and transfer information can be skipped by the setting of the DTCCR.RRS bit.

When a DTC transfer request is accepted, the current DTC vector number is compared with the DTC vector number in the previous data transfer. When these vector numbers match and the RRS bit is 1, the DTC does not read the DTC vector and transfer information, and transfers data according to the transfer information remained in the DTC.

However, when the previous transfer was chain transfer, the DTC vector and transfer information are read. Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, transfer information is read regardless of the value of the RRS bit. Figure 20.13 shows an example of transfer information read skip.

When updating the vector table and transfer information, set the RRS bit to 0, update the vector table and transfer information, and then set the RRS bit to 1. Setting the RRS bit to 0 discards the vector numbers retained in the DTC. The updated DTC vector table and transfer information are read in the next data transfer.

20.4.2 Transfer Information Write-Back Skip Function

When the MRA.SM[1:0] bits or the MRB.DM[1:0] bits are set to “address is fixed” (00b or 01b), a part of transfer information is not written back. This function is performed independently of the setting of short-address mode or full-address mode.

Table 20.4 lists transfer information write-back skip conditions and applicable registers. The CRA and CRB registers are written back independently of the setting of short-address mode or full-address mode.

Furthermore, in full-address mode, write-back of registers MRA and MRB is skipped.

Table 20.4 Transfer Information Write-Back Skip Conditions and Applicable Registers

MRA.SM[1:0] Bits		**MRB.DM[1:0] Bits**			
b3	b2	b3	b2	**SAR Register**	**DAR Register**
0	0	0	0	Skip	Skip
0	0	0	1		
0	1	0	0		
0	1	0	1		
0	0	1	0	Skip	Write-back
0	0	1	1		
0	1	1	0		
0	1	1	1		
1	0	0	0	Write-back	Skip
1	0	0	1		
1	1	0	0		
1	1	0	1		
1	0	1	0	Write-back	Write-back
1	0	1	1		
1	1	1	0		
1	1	1	1		

20.4.3 Normal Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single transfer request. The transfer count can be set to 1 to 65536.

Transfer source addresses and transfer destination addresses can be set to increment, decrement, or fixed independently.

This mode enables an interrupt request to the CPU to be generated at the end of specified-count transfer.

Table 20.5 lists register functions in normal transfer mode, and Figure 20.5 shows the memory map of normal transfer mode.

Table 20.5 Register Functions in Normal Transfer Mode

Register	Description	Value Written Back by Writing Transfer Information
SAR	Transfer source address	Increment/decrement/fixed*1
DAR	Transfer destination address	Increment/decrement/fixed*1
CRA	Transfer counter A	CRA – 1
CRB	Transfer counter B	Not updated

Note 1. Write-back operation is skipped when address is fixed.

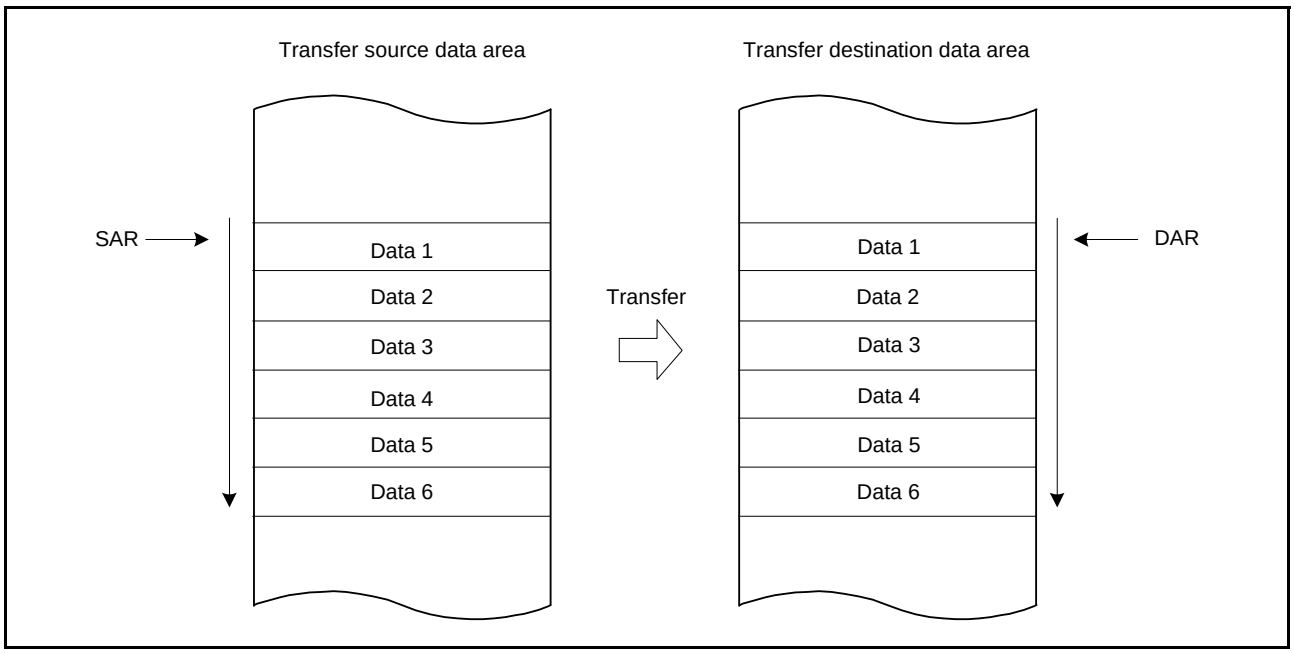


Figure 20.5 Memory Map of Normal Transfer Mode

20.4.4 Repeat Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single transfer request.
Specify either transfer source or transfer destination for the repeat area by the MRB.DTS bit. The transfer count can be set to 1 to 256. When the specified-count transfer is completed, the initial value of the address register specified in the transfer counter and the repeat area is restored and transfer is repeated. The other address register is incremented or decremented continuously or remains unchanged.
When the transfer counter CRAL is decreased to 00h in repeat transfer mode, the CRAL value is updated to the value set in the CRAH register. Thus the transfer counter does not become 00h, which disables an interrupt request to be generated to the CPU when the MRB.DISEL bit is set to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers).
Table 20.6 lists the register functions in repeat transfer mode, and Figure 20.6 shows the memory map of repeat transfer mode.

Table 20.6 Register Functions in Repeat Transfer Mode

		Value Written Back by Writing Transfer Information		
			When CRAL = 1	
Register	**Description**	**When CRAL ≠ 1**	**When the MRB.DTS bit is 0**	**When the MRB.DTS bit is 1**
SAR	Transfer source address	Increment/decrement/fixed*1	Increment/decrement/fixed*1	SAR register initial value
DAR	Transfer destination address	Increment/decrement/fixed*1	DAR register initial value	Increment/decrement/fixed*1
CRAH	Retains initial value of transfer counter	CRAH	CRAH	
CRAL	Transfer counter A	CRAL – 1	CRAH	
CRB	Transfer counter B	Not updated	Not updated	

Note 1. Write-back operation is skipped when address is fixed.

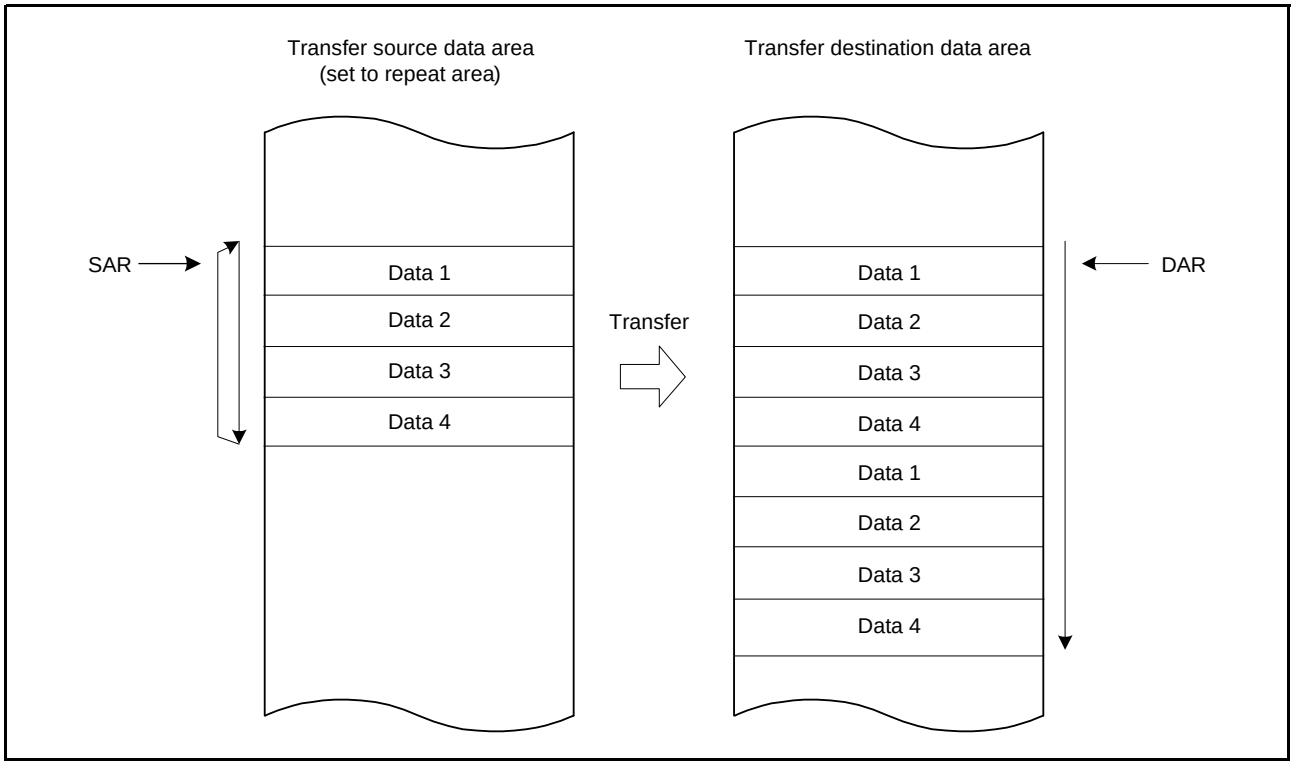


Figure 20.6 Memory Map of Repeat Transfer Mode (Transfer Source: Repeat Area)

20.4.5 Block Transfer Mode

This mode allows single-block data transfer on a single transfer request.
Specify either transfer source or transfer destination for the block area by the MRB.DTS bit. The block size can be set to 1 to 256 bytes, 1 to 256 words, or 1 to 256 longwords.
When transfer of the specified one block is completed, the initial values of the block size counter CRAL and the address register (the SAR register when the MRB.DTS bit is 1 or the DAR register when the DTS bit is 0) specified in the block area are restored. The other address register is incremented or decremented continuously or remains unchanged.
The transfer count (block count) can be set to 1 to 65536. This mode enables an interrupt request to the CPU to be generated at the end of specified-count block transfer.
Table 20.7 lists register functions in block transfer mode, and Figure 20.7 shows the memory map of block transfer mode.

Table 20.7 Register Functions in Block Transfer Mode

		Value Written Back by Writing Transfer Information	
Register	**Description**	**When MRB.DTS bit is 0**	**When MRB.DTS bit is 1**
SAR	Transfer source address	Increment/decrement/fixed*1	SAR register initial value
DAR	Transfer destination address	DAR register initial value	Increment/decrement/fixed*1
CRAH	Retains initial value of block size	CRAH	
CRAL	Block size counter	CRAH	
CRB	Block transfer counter	CRB – 1	

Note 1. Write-back operation is skipped when address is fixed.

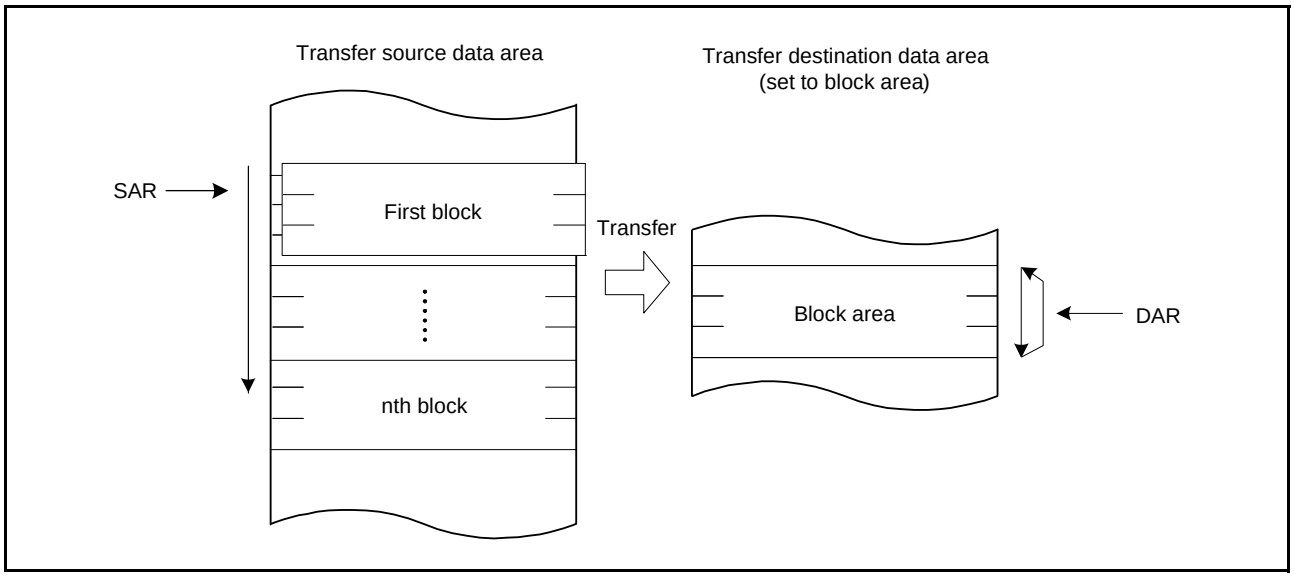


Figure 20.7 Memory Map of Block Transfer Mode (Transfer Destination: Block Area)

20.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single transfer request. If the MRB.CHNE bit is 1 and the MRB.CHNS bit is 0, an interrupt request to the CPU is not generated when the specified number of data transfers is completed, or while the MRB.DISEL bit is 1 (an interrupt request to the CPU is generated for every data transfer). Data transfer has no effect on the interrupt status flag, which is the request source. The transfer information (SAR, DAR, CRA, CRB, MRA, and MRB) that define a data transfer can be specified independently of each other. Figure 20.8 shows chain transfer operation.

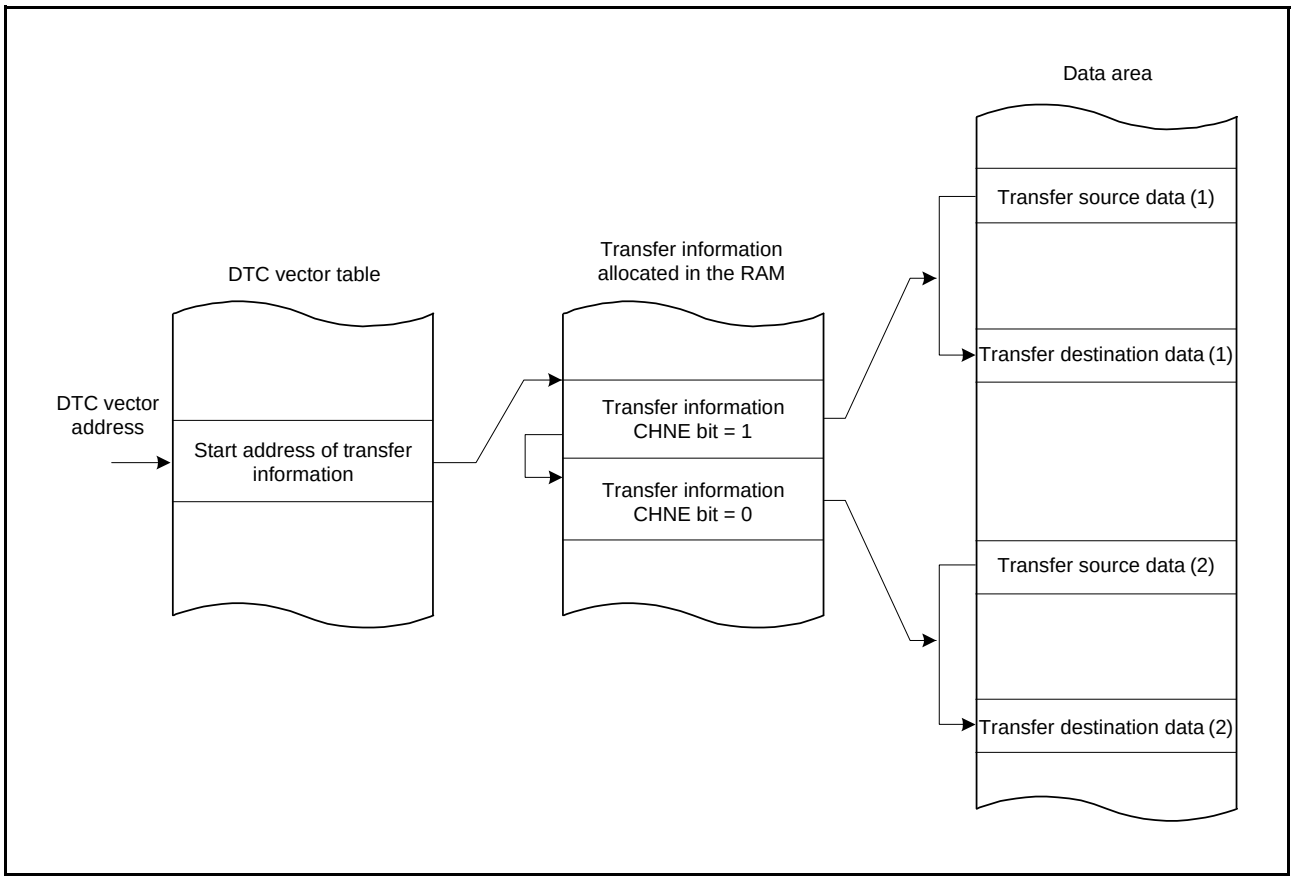


Figure 20.8 Chain Transfer Operation

If the MRB.CHNE bit is 1 and the CHNS bit is 1, chain transfer is performed only after completion of specified number of data transfers. In repeat transfer mode, chain transfer is performed after completion of specified number of data transfers.
For details on chain transfer conditions, refer to Table 20.3, Chain Transfer Conditions.

20.4.7 Operation Timing

Figure 20.9 to Figure 20.13 show examples of DTC operation timing.

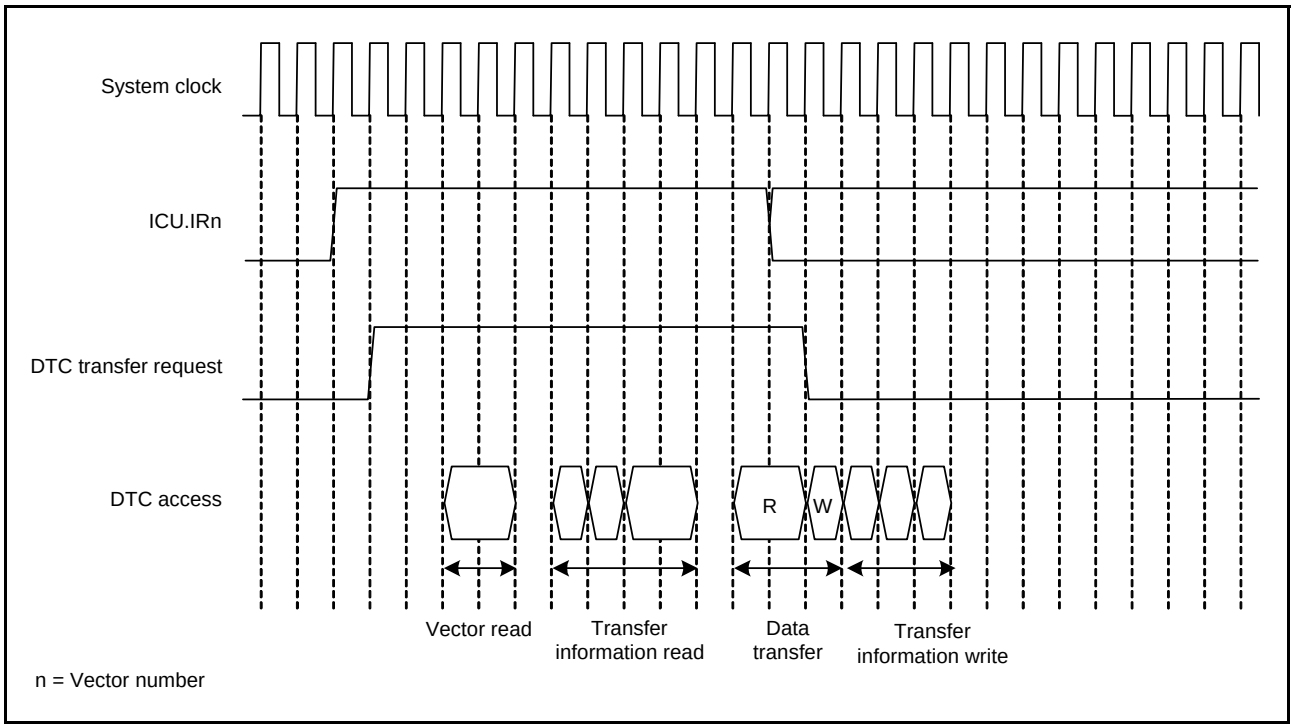


Figure 20.9 Example (1) of DTC Operation Timing (Short-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)

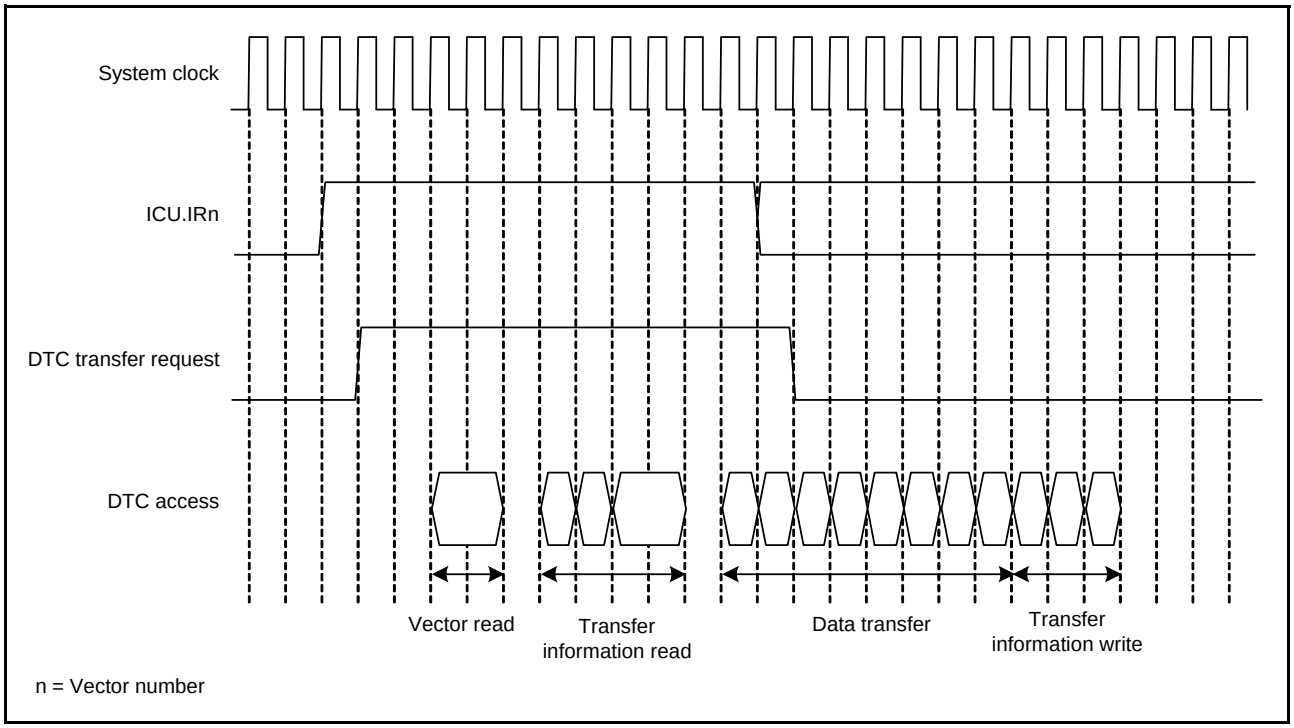


Figure 20.10 Example (2) of DTC Operation Timing (Short-Address Mode, Block Transfer Mode, Block Size = 4)

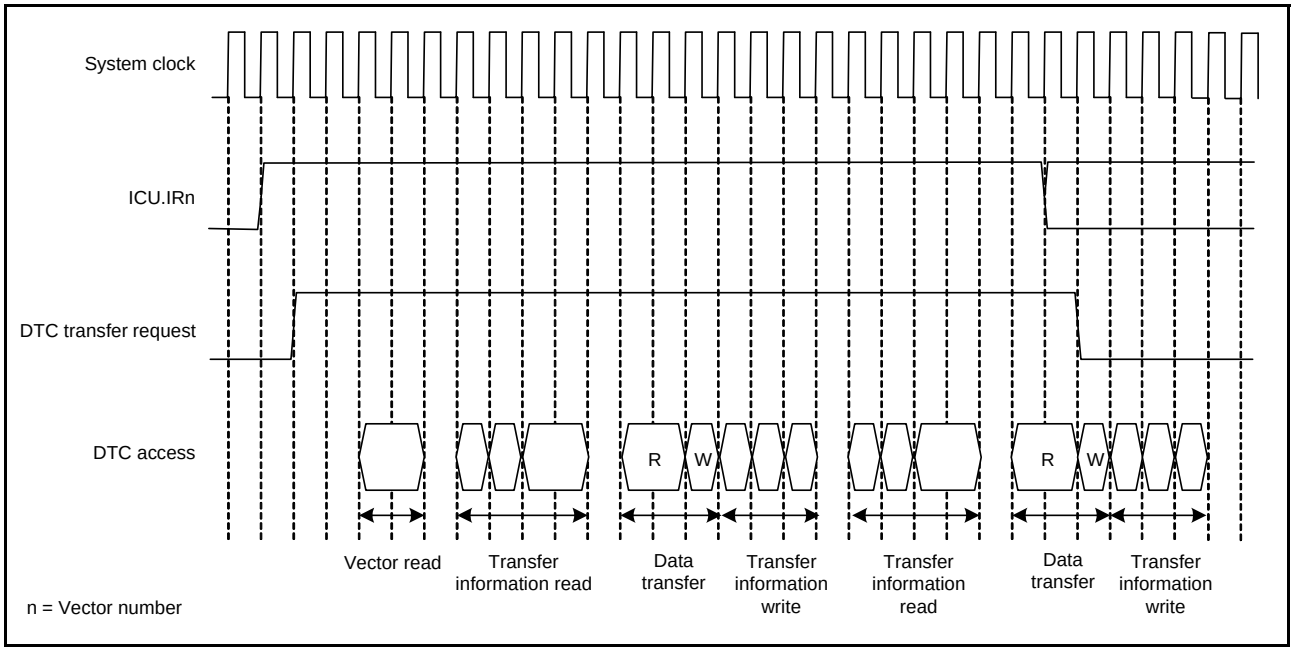


Figure 20.11 Example (3) of DTC Operation Timing (Short-Address Mode, Chain Transfer)

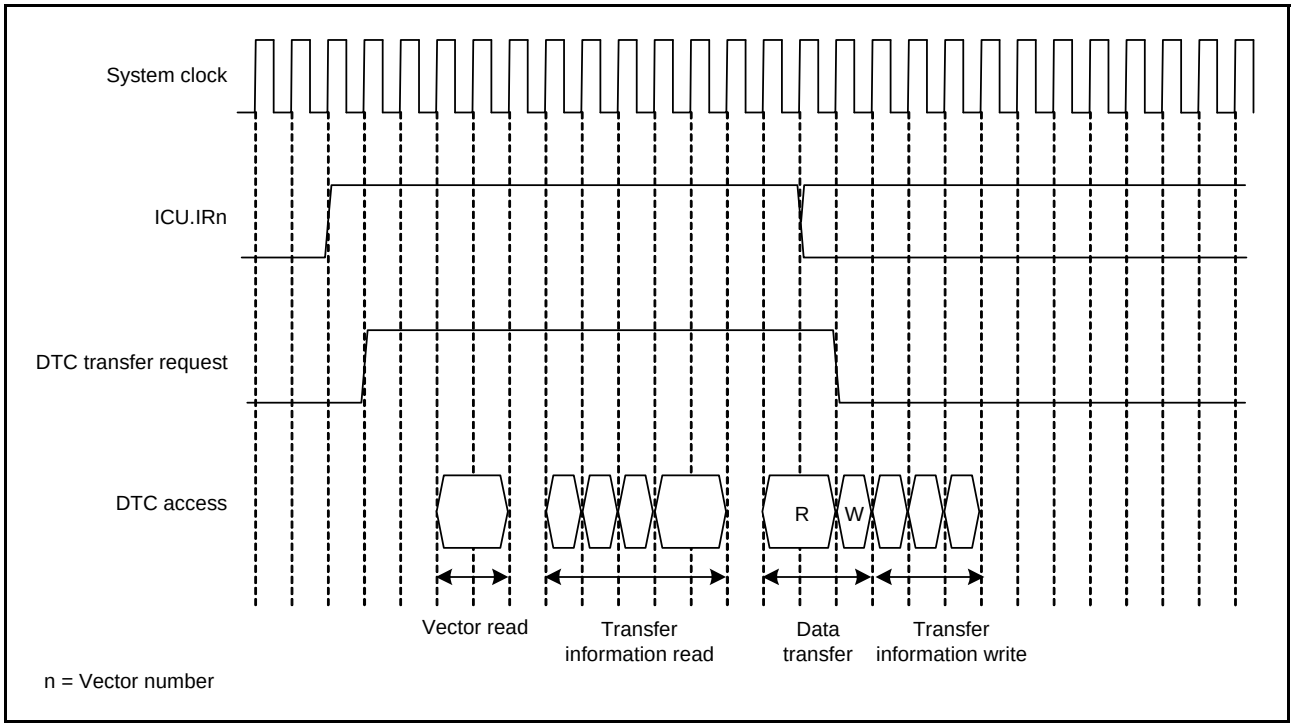
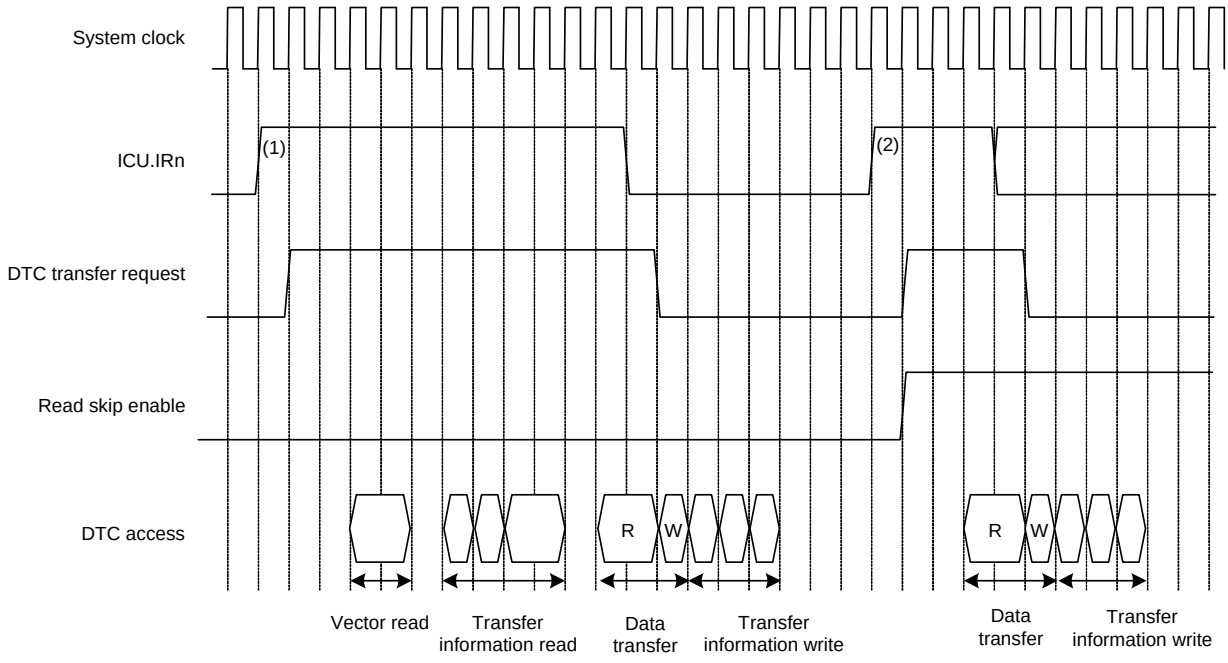


Figure 20.12 Example (4) of DTC Operation Timing (Full-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)



n = Vector number

Note: When request sources (vector numbers) of (1) and (2) are the same and the DTCCR.RRS bit is 1, the transfer information read for request (2) is skipped.

Figure 20.13 Example of Operation When Transfer Information Read Skip is Executed (Vector, Transfer Information, and Transfer Destination Data on the RAM, and Transfer Source Data on the Peripheral Module)

20.4.8 Execution Cycles of the DTC

Table 20.8 lists the execution cycles of single data transfer of the DTC.
For the order of the execution states, refer to section 20.4.7, Operation Timing.

Table 20.8 Execution Cycles of the DTC

Transfer Mode	Vector Read		Transfer Information Read			Transfer Information Write			Data Transfer		Internal Operation	
									Read	Write		
Normal	Cv + 1	0*1	4 × Ci + 1*2	3 × Ci + 1*3	0*1	3 × Ci*4	2 × Ci*5	Ci*6	Cr + 1	Cw	2	0*1
Repeat									Cr + 1	Cw		
Block*7									P × Cr	P × Cw		

Note 1. When transfer information read is skipped
Note 2. In full-address mode
Note 3. In short-address mode
Note 4. When neither SAR nor DAR is set to address-fixed
Note 5. When SAR or DAR is set to address-fixed
Note 6. When SAR and DAR are set to address-fixed
Note 7. When the block size is 2 or more. If the block size is 1, the cycle number for normal transfer is applied.

P: Block size (initial settings of CRAH and CRAL)
Cv: Cycles for access to vector transfer information storage destination
Ci: Cycles for access to transfer information storage destination address
Cr: Cycles for access to data read destination
Cw: Cycles for access to data write destination
(The unit is system clocks (ICLK) for “+ 1” in the Vector Read, Transfer Information Read, and Data Transfer Read columns and “2” in the Internal Operation column.)
(Cv, Ci, Cr, and Cw vary depending on the corresponding access destination. For the number of cycles for respective access destinations, refer to section 61, RAM, section 63, Flash Memory, section 5, I/O Registers, and section 16.2.6, External Bus.)

20.4.9 DTC Bus Mastership Release Timing

The DTC does not release the bus mastership during transfer information read and transfer information write. While transfer information is not read or written, bus arbitration is made according to the priority determined by the bus master arbitrator.
For bus arbitration, refer to section 16, Buses.

20.5 DTC Setting Procedure

Before using the DTC, set the DTC vector base register (DTCVBR).
Figure 20.14 shows the procedure to set the DTC.

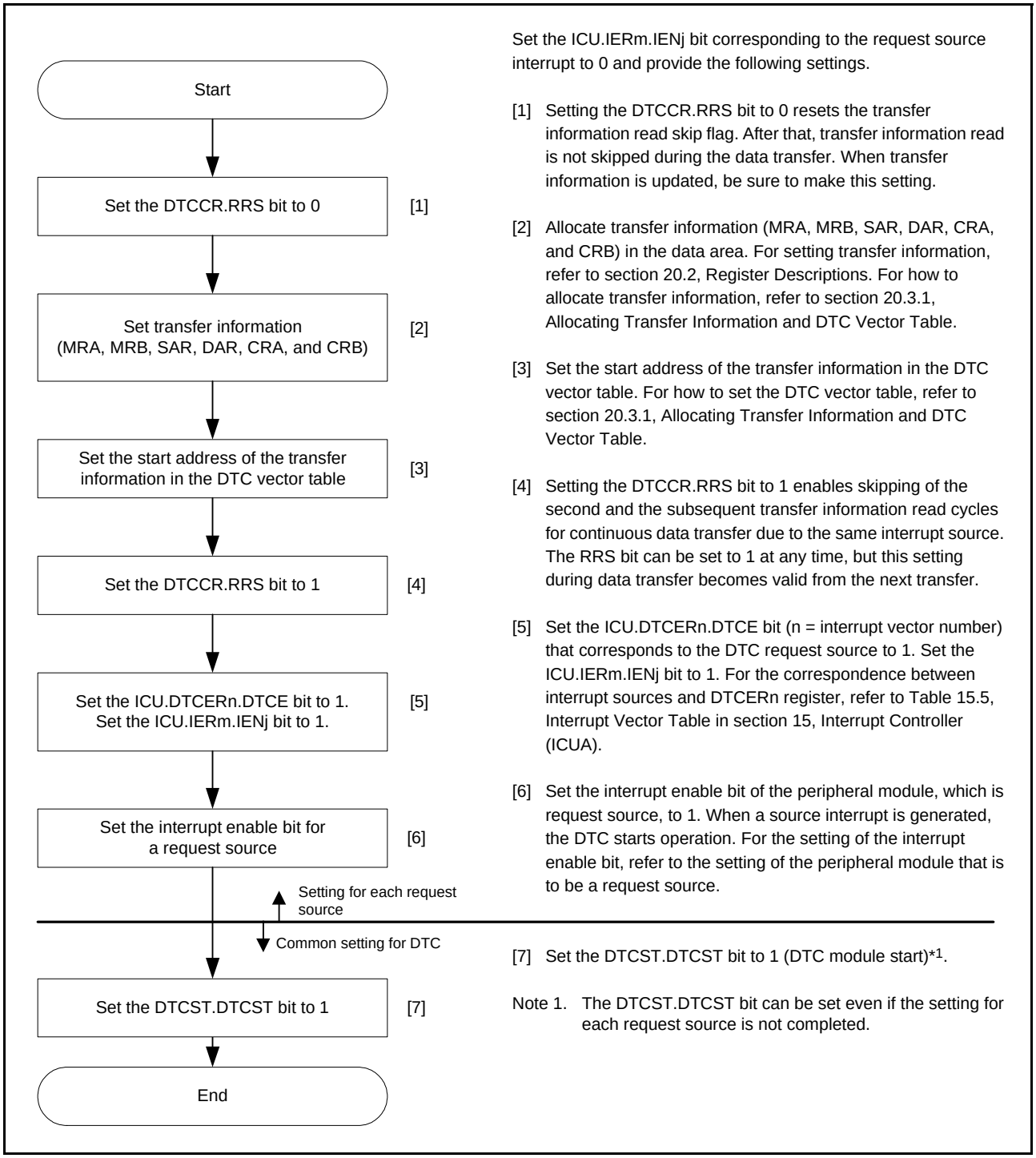


Figure 20.14 Procedure to Set the DTC

20.6 Examples of DTC Usage

20.6.1 Normal Transfer

As an example of DTC usage, its employment in the reception of 128 bytes of data by an SCI is described below.

(1) Transfer Information Setting

Set the MRA.MD[1:0] bits to 00b (normal transfer mode), the MRA.SZ[1:0] bits to 00b (byte transfer), and the MRA.SM[1:0] bits to 00b (source address is fixed). Set the MRB.CHNE bit to 0 (chain transfer is disabled), the MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers), and the MRB.DM[1:0] bits to 10b (DAR is inclemented after data transfer). The MRB.DTS bit can be set to any value. Set the RDR register address of the SCI in the SAR register, the start address of the RAM area for data storage in the DAR register, and 128 (0080h) in the CRA register. The CRB register can be set to any value.

(2) DTC Vector Table Setting

The start address of the transfer information for the RXI interrupt is set in the vector table for the DTC.

(3) ICU Setting and DTC Module Activation

Set the corresponding ICU.DTCERn.DTCE bit to 1 and the ICU.IERm.IENj bit to 1.
Set the DTCST.DTCST bit to 1.

(4) SCI Setting

Enable the RXI interrupt by setting the SCR.RIE bit in the SCI to 1. If a reception error occurs during the SCI receive operation, further reception is not performed. Accordingly, make settings so that the CPU can accept receive error interrupts.

(5) DTC Transfer

Every time the reception of 1 byte by the SCI is completed, an RXI interrupt is generated to start the data transfer. The DTC transfers the received byte from the RDR of the SCI to RAM, after which the DAR register is incremented and the CRA register is decremented.

(6) Interrupt Handling

After 128 times of data transfers have been completed and the value in the CRA register becomes 0, an RXI interrupt request is output to the CPU. Complete the process in the handling routine for this interrupt.

20.6.2 Chain Transfer

As an example of chain transfer by the DTC, its employment in the output of pulses by a PPG is described below. Lower-case letters x, y, and k in this text indicate a unit, channel, and bit number, respectively.
Chain transfer is used to transfer pulse output data and change the period of the output trigger for the PPG. For the first half of the chain transfer, repeat transfer mode for transfer to the PPGx.NDRH and PPGx.NDRL registers is specified. For the second half, normal transfer mode for transfer to the MTUy.TGR registers is specified. This is because clearing of the request source and generation of an interrupt on completion of the specified number of data transfer are restricted to the second half of the chain transfer (transfer while the MRB.CHNE bit is 0).
An example of using the compare match interrupt for an MTUy.TGRA register as a request source for the DTC is provided below.

(1) First Transfer Information Setting

Settings should be made for transfer to the PPGx.NDRH and PPGx.NDRL registers. Set the MRA.MD[1:0] bits to 01b (repeat transfer mode), the MRA.SZ[1:0] bits to 01b (word transfer), and the MRA.SM[1:0] bits to 10b (SAR is inclemented after data transfer). Set the MRB.CHNE bit to 1 (chain transfer is enabled), the MRB.CHNS bit to 0 (chain transfer is performed after each transfer), the MRB.DTS bit to 1 (source is repeat area), and the MRB.DM[1:0] bits to 00b (destination address is fixed). Set the SAR to the start address of the data table, the DAR register to the address of the PPGx.NDRH register, and the CRAH and CRAL registers to the size of the data table. The CRB register can be set to any value.

(2) Second Transfer Information Setting

Settings should be made for transfer to the MTUy.TGRA register. Set the MRA.MD[1:0] bits to 00b (normal transfer mode), the MRA.SZ[1:0] bits to 01b (word transfer), and the MRA.SM[1:0] bits to 10b (SAR is inclemented after data transfer). Set the MRB.CHNE bit to 0 (chain transfer is disabled), the MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers), and the MRB.DM[1:0] bits to 00b (destination address is fixed). The MRB.DTS bit can be set to any value. Set the SAR register to the start address of the data table, the DAR register to the address of the MTUy.TGRA register, and the CRA register to the size of the data table. The CRB register can be set to any value.

(3) Transfer Information Assignment

Place the second transfer information immediately after the first transfer information.

(4) DTC Vector Table

In the DTC vector table, set the address where the first transfer information starts.

(5) ICU Setting and DTC Module Activation

Set the ICU.DTCERn.DTCE bit corresponding to the TGIA interrupt and the ICU.IERm.IENj bit to 1.
Set the DTCST.DTCST bit to 1.

(6) MTU Setting

In the given MTUy, set the TIOR register so that the TGRA register operates as an output compare register (with output disabled) and make the TIER setting to enable TGIAn interrupt requests.

(7) PPG Setting

Set the default output values in the PPGx.PODRH and PPGx.PODRL registers and the next output values in the PPGx.NDRH and PPGx.NDRL registers. Set 1 to the output bits in PORTm.PDR and PPGx.NDRH, and PPGx.NDRL. Also, select a compare match signal of the MTU as the output trigger in the PPGx.PCR register.

(8) MTU Activation

Set the MTU.TSTR.CSTk bits to 1 to start counting operation of the MTUy.TCNT counter.

(9) DTC Transfer

Every time a compare-match with the MTUy.TGRA register is generated, next output values are transferred to the PPGx.NDRH and PPGx.NDRL registers and the setting for the next output trigger period is transferred to the MTUy.TGRA register.

(10) Interrupt Handling

When the specified number of data transfers is completed (i.e. when the value of the CRA register for second transfer becomes 0), a TGIAn interrupt request is issued for the CPU. Complete the process in the handling routine for this interrupt.

20.6.3 Chain Transfer When the Counter is 0

The second data transfer is performed only when the transfer counter is set to 0 in the first data transfer, and the first data transfer information is repeatedly changed in the second data transfer. Repeating this chain transfer enables transfers to be repeated more than 256 times.

The following shows an example of configuring a 128-Kbyte input buffer to addresses 20 0000h to 21 FFFFh (where the input buffer is set so that its lower address starts with 0000h). Figure 20.15 shows a chain transfer when the counter is 0.

(1) Set normal transfer mode for input data for the first data transfer. Set the following:
Transfer source address: Fixed, the CRA register is 0000h (65,536 times), the MRB.CHNE bit is 1 (chain transfer is enabled), the MRB.CHNS bit is 1 (chain transfer is performed only when the transfer counter becomes 0), and the MRB.DISEL bit is 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers).

(2) Prepare the upper 8 bits (in this case, 21h and 20h) of the start address at every 65,536 times of the transfer destination address for the first data transfer in another area (such as ROM).

(3) For the second data transfer, set repeat transfer mode (source is repeat area) for rewriting the transfer destination address of the first data transfer. The transfer destination is the address where the upper 8 bits of the DAR register in the first transfer information is allocated. In this case, set the MRB.CHNE bit to 0 (chain transfer is disabled) and the MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers). In this case, set the transfer counter to 2.

(4) When a transfer request is accepted, the first data transfer is executed. When transfer is executed 65,536 times and the transfer counter of the first data transfer becomes 0, the second data transfer is started and the upper 8 bits of the transfer destination address of the first data transfer is set to 21h. At this time, the lower 16 bits of the transfer destination address and the transfer counter of the first data transfer have become 0000h.

(5) In succession, when another transfer request is accepted, the first data transfer is executed. When transfer is executed 65,536 times and the transfer counter of the first data transfer becomes 0, the second data transfer is started and the upper 8 bits of the transfer destination address of the first data transfer is set to 20h. At this time, the lower 16 bits of the transfer destination address and the transfer counter of the first data transfer have become 0000h.

(6) Steps (4) and (5) above are repeated infinitely. Because the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

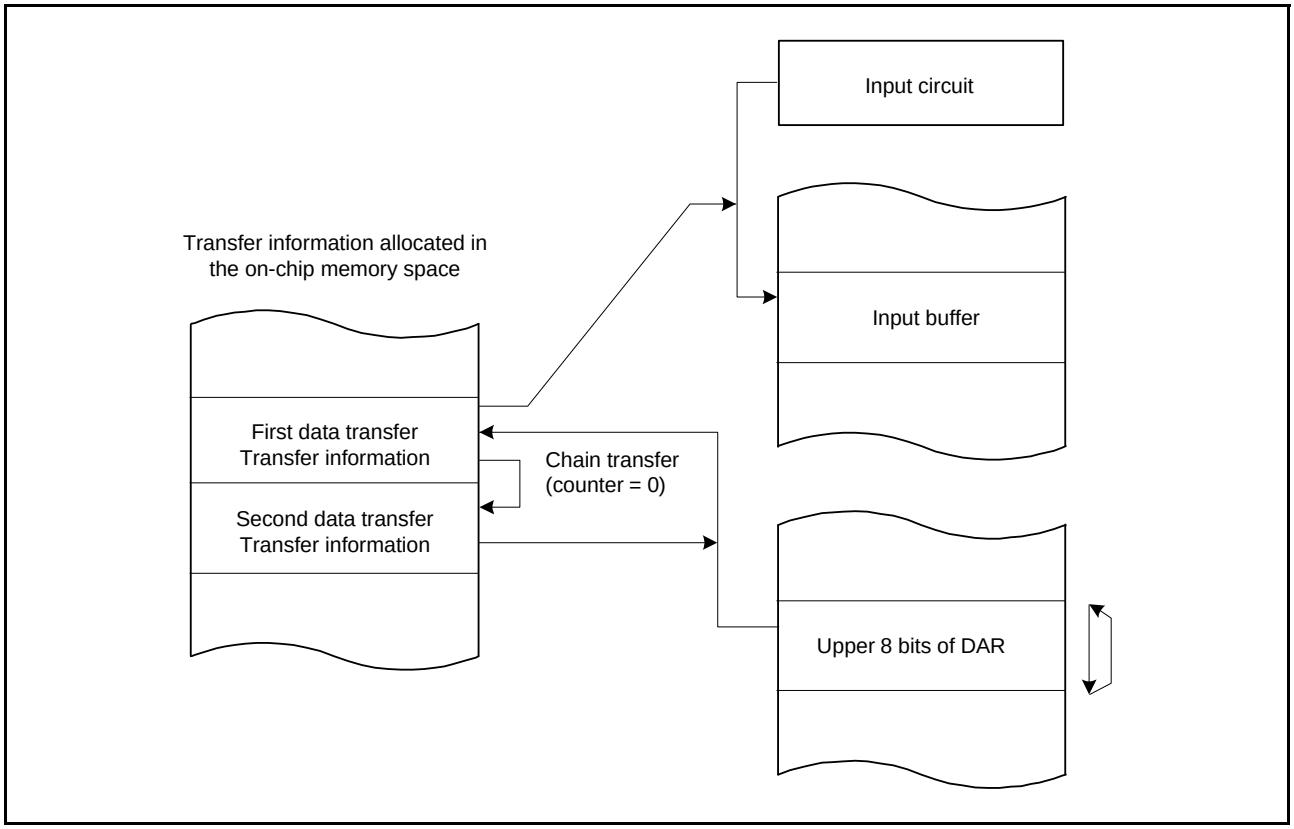


Figure 20.15 Chain Transfer When the Counter is 0

20.7 Interrupt Source

When the DTC has finished data transfer of specified count or when data transfer with the MRB.DISEL bit set to 1 (an interrupt request to the CPU is generated each time the data transfer is performed) has been completed, an interrupt to the CPU is generated by the DTC trigger source. Such interrupts to the CPU are controlled according to the PSW.I bit (interrupt enable) of the CPU, the PSW.IPL[3:0] bits (processor interrupt priority level), and the priority level of the interrupt controller.

20.8 Event Link

The DTC outputs an event signal on completing data transfer in response to one request. When the destination for transfer is an external bus or an internal peripheral bus, however, the event signal will be output after completion of writing to the write buffer rather than after completion of writing to the actual destination for transfer.

20.9 Low Power Consumption Function

Before making a transition to the module stop state, all-module clock stop mode, software standby mode, or deep software standby mode, set the DTCST.DTCST bit to 0 (DTC module stop), and then perform the following.

(1) Module Stop Function

Writing 1 (transition to the module-stop state is made) to the MSTPCRA.MSTPA28 bit enables the module stop function of the DTC. If data transfer is in progress at the time 1 is written to the MSTPCRA.MSTPA28 bit, the transition to the module stop state proceeds after data transfer has ended. While the MSTPCRA.MSTPA28 bit is 1, accessing the DTC registers is prohibited.
Writing 0 (release from the module-stop state) to the MSTPCRA.MSTPA28 bit releases the DTC from the module stop state.

(2) All-Module Clock Stop Mode

Make settings according to the procedure under section 11.6.2.1, Transition to All-Module Clock Stop Mode, in section 11, Low Power Consumption.
If any data transfer is in progress at the time the WAIT instruction is executed, the transition to all-module clock stop mode follows the completion of the data transfer.
The DTC is released from the module stop state by writing 0 to the MSTPCRA.MSTPA28 bit following recovery from all-module clock stop mode.

(3) Software Standby and Deep Software Standby Modes

Make settings according to the procedure under section 11.6.3.1, Transition to Software Standby Mode, or section 11.6.4.1, Transition to Deep Software Standby Mode, in section 11, Low Power Consumption.
If any data transfer is in progress at the time the WAIT instruction is executed, the transition to software standby mode or deep software standby mode follows the completion of the data transfer.

(4) Notes on Low Power Consumption Function

For the WAIT instruction and the register setting procedure, refer to section 11.7.6, Timing of Wait Instructions in section 11, Low Power Consumption.
To perform data transfer after returning from a low power consumption mode, set the DTCST.DTCST bit to 1 again.
To use a request that is generated in all-module clock stop mode or software standby mode as an interrupt request to the CPU but not as a DTC transfer request, specify the CPU as the interrupt request destination according to the description in section 15.7.3.1, Interrupt Request Destination Setting Procedure in section 15, Interrupt Controller (ICUA), and then execute the WAIT instruction.

20.10 Usage Notes

20.10.1 Start Address of Transfer Information

Set multiples of 4 for the start addresses of the transfer information to be specified in the DTC vector table. If any value other than a multiple of 4 is specified, access still proceeds with the lower 2 bits of the address regarded as 00b.

20.10.2 Allocating Transfer Information

Allocate transfer information in the memory area according to the endian of the area as shown in Figure 20.16. For example, when writing CRA and CRB settings in 16-bit units in big endian, write the CRA setting to the address plus 8h (Ch) and the CRB setting to the address plus Ah (Eh). In little endian, write the CRB setting to the address plus 8h (Ch) and the CRA setting to the address plus Ah (Eh). When writing CRA and CRB settings in 32-bit units, allocate the CRA setting at the MSB side of the 32 bits and the CRB setting at the LSB side, and write the settings to the address plus 8h (Ch), regardless of endian.

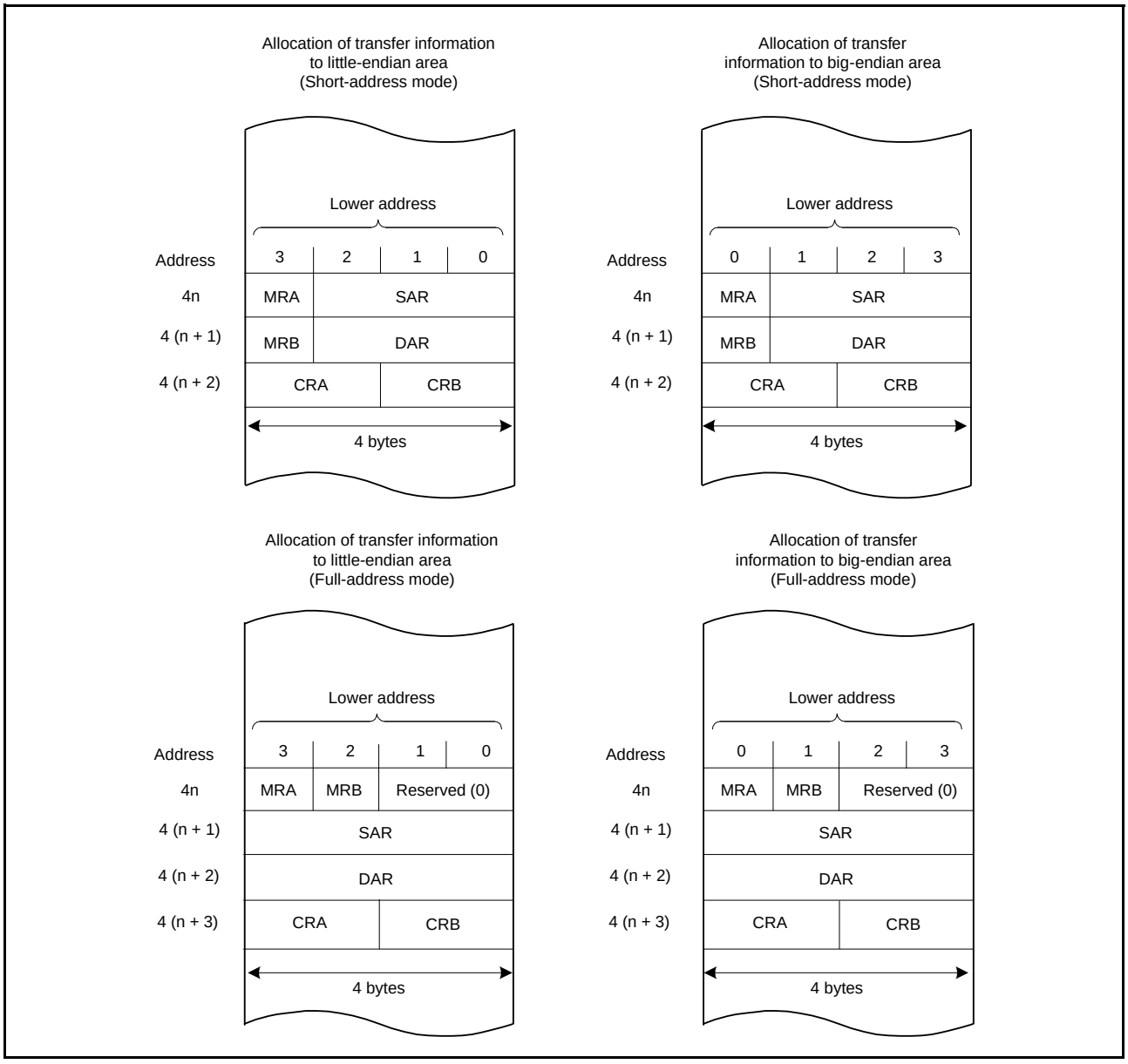


Figure 20.16 Allocation of Transfer Information

20.10.3 Setting the DTC Transfer Request Enable Register in the Interrupt Controller (ICU.DTCERn)

The DMA request should not be issued by setting the DMAC trigger select register (ICU.DMRSRm (m = DMAC channel number)) to the same vector number that has been specified by setting the ICU.DTCERn.DTCE bit to 1 (the corresponding interrupt source is selected as the DTC trigger). For details on the ICU.DTCERn and ICU.DMRSRm registers (m = DMAC channel number), refer to section 15, Interrupt Controller (ICUA).

21. Event Link Controller (ELC)

21.1 Overview

The event link controller (ELC) uses the interrupt requests generated by various peripheral modules as event signals to connect (link) them to different modules, allowing direct cooperation between the modules without CPU intervention. Event signals can be output regardless of the setting of the corresponding interrupt request enable bit.

Table 21.1 lists the specifications of the ELC, and Figure 21.1 shows a block diagram of the ELC.

Table 21.1 ELC Specifications

Item	Description
Event link function	• 119 types of event signals can be directly connected to modules. • The operation of timer modules can be selected when an event is input to the timer module. • Event link operation is possible for port B and port E. Single port*1: An event link can be set for a single bit specified in a port. Port group*1: An event link can be set for a group of single bits specified within eight I/O ports.
Low power consumption function	Module stop state can be set.

Note 1. The single port and port group specified as the input generate an event according to the change in the connected signal value.

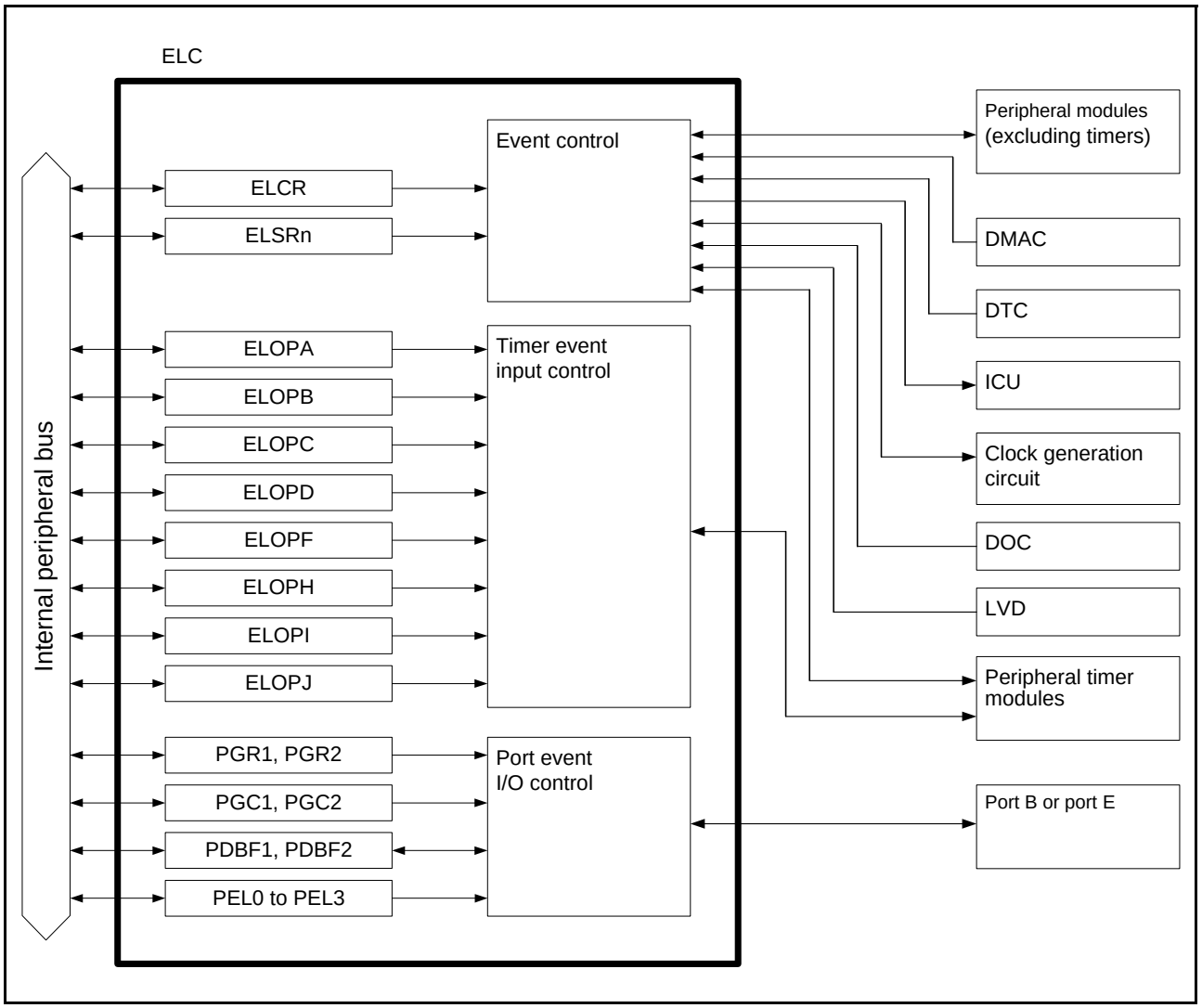


Figure 21.1 ELC Block Diagram (n = 0, 3, 4, 7, 10 to 13, 15, 16, 18 to 28, 33, 35 to 38, 41 to 45)

21.2 Register Descriptions

21.2.1 Event Link Control Register (ELCR)

Address(es): 0008 B100h

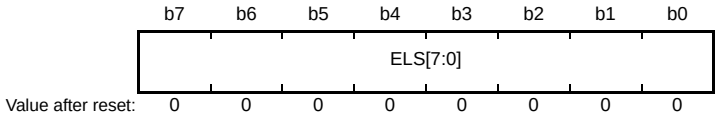
	b7	b6	b5	b4	b3	b2	b1	b0
	ELCON	—	—	—	—	—	—	—
Value after reset:	0	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7	ELCON	All Event Link Enable	0: ELC function is disabled. 1: ELC function is enabled	R/W

The ELCR register controls operation of the ELC.

21.2.2 Event Link Setting Register n (ELSRn)
(n = 0, 3, 4, 7, 10 to 13, 15, 16, 18 to 28, 33, 35 to 38, 41 to 45)

Address(es): ELSR0 0008 B101h, ELSR3 0008 B104h, ELSR4 0008 B105h, ELSR7 0008 B108h, ELSR10 0008 B10Bh, ELSR11 0008 B10Ch, ELSR12 0008 B10Dh, ELSR13 0008 B10Eh, ELSR15 0008 B110h, ELSR16 0008 B111h, ELSR18 0008 B113h, ELSR19 0008 B114h, ELSR20 0008 B115h, ELSR21 0008 B116h, ELSR22 0008 B117h, ELSR23 0008 B118h, ELSR24 0008 B119h, ELSR25 0008 B11Ah, ELSR26 0008 B11Bh, ELSR27 0008 B11Ch, ELSR28 0008 B11Dh, ELSR33 0008 B131h, ELSR35 0008 B133h, ELSR36 0008 B134h, ELSR37 0008 B135h, ELSR38 0008 B136h, ELSR41 0008 B139h, ELSR42 0008 B13Ah, ELSR43 0008 B13Bh, ELSR44 0008 B13Ch, ELSR45 0008 B13Dh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ELS[7:0]	Event Link Select	00h: Event output to the corresponding peripheral module is disabled. 01h to BDh: Set the number for the event signal to be linked. Settings other than above are prohibited.	R/W

The ELSRn register specifies an event signal to be linked to for each peripheral module. Table 21.2 shows the correspondence between the ELSRn register and the peripheral modules. Table 21.3 shows the correspondence between the event signal names set in the ELSRn register and the signal numbers.

Table 21.2 Correspondence between the ELSRn Register and the Peripheral Modules

Register Name	Peripheral Module
ELSR0	MTU0
ELSR3	MTU3
ELSR4	MTU4
ELSR7	CMT1
ELSR10	TMR0
ELSR11	TMR1
ELSR12	TMR2
ELSR13	TMR3
ELSR15	S12AD
ELSR16	DA0
ELSR18	ICU (Interrupt 1)*1
ELSR19	ICU (Interrupt 2)*1
ELSR20	Output port group 1
ELSR21	Output port group 2
ELSR22	Input port group 1
ELSR23	Input port group 2
ELSR24	Single port 0*2
ELSR25	Single port 1*2
ELSR26	Single port 2*2
ELSR27	Single port 3*2
ELSR28	Clock source switching to LOCO
ELSR33	CMTW0
ELSR35	TPU0
ELSR36	TPU1
ELSR37	TPU2
ELSR38	TPU3
ELSR41	GPT0
ELSR42	GPT1
ELSR43	GPT2
ELSR44	GPT3
ELSR45	S12AD1

Note 1. Specify an event number from among 63h to 6Ah. Do not set other settings.
Note 2. Do not set the DOC data operation condition met signal (6Ah) in the ELSR24, ELSR25, ELSR26, or ELSR27 register.

Table 21.3 Correspondence between Event Signal Names Set in ELSRn.ELS[7:0] Bits and Signal Numbers (1/4)

ELS[7:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
01h	Multifunction timer pulse unit 3	MTU0 compare match 0A
02h		MTU0 compare match 0B
03h		MTU0 compare match 0C
04h		MTU0 compare match 0D
05h		MTU0 compare match 0E
06h		MTU0 compare match 0F
07h		MTU0 overflow
10h		MTU3 compare match 3A
11h		MTU3 compare match 3B
12h		MTU3 compare match 3C
13h		MTU3 compare match 3D
14h		MTU3 overflow
15h		MTU4 compare match 4A
16h		MTU4 compare match 4B
17h		MTU4 compare match 4C
18h		MTU4 compare match 4D
19h		MTU4 overflow
1Ah		MTU4 underflow
1Fh	Compare match timer	CMT1 compare match 1
22h	8-bit timers	TMR0 compare match A0
23h		TMR0 compare match B0
24h		TMR0 overflow
25h		TMR1 compare match A1
26h		TMR1 compare match B1
27h		TMR1 overflow
28h		TMR2 compare match A2
29h		TMR2 compare match B2
2Ah		TMR2 overflow
2Bh		TMR3 compare match A3
2Ch		TMR3 compare match B3
2Dh		TMR3 overflow
2Eh	Realtime clock	RTC cycle (select 1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, or 2 seconds)
31h	Independent watchdog timer	IWDT underflow or refresh error
3Ah	Serial communications interfaces	SCI5 error (receive error or error signal detection)
3Bh		SCI5 receive data full
3Ch		SCI5 transmit data empty
3Dh		SCI5 transmit end
4Eh	I^2C bus interface	RIIC0 communication error or event generation
4Fh		RIIC0 receive data full
50h		RIIC0 transmit data empty
51h		RIIC0 transmit end

Table 21.3 Correspondence between Event Signal Names Set in ELSRn.ELS[7:0] Bits and Signal Numbers (2/4)

ELS[7:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
52h	Serial peripheral interface	RSPI0 error (mode fault, overrun, or parity error)
53h		RSPI0 idle
54h		RSPI0 receive data full
55h		RSPI0 transmit data empty
56h	Serial peripheral interface	RSPI0 transmit end
58h	12-bit A/D converter	S12AD A/D conversion end
5Bh	Voltage detection circuit	LVD1 voltage detection
5Ch		LVD2 voltage detection
5Dh	DMA controller	DMAC0 transfer end
5Eh		DMAC1 transfer end
5Fh		DMAC2 transfer end
60h		DMAC3 transfer end
61h	Data transfer controller	DTC transfer end
62h	Clock generation circuit	Oscillation stop detection of clock generation circuit
63h	I/O ports	Input edge detection of input port group 1
64h		Input edge detection of input port group 2
65h		Input edge detection of single input port 0
66h		Input edge detection of single input port 1
67h		Input edge detection of single input port 2
68h		Input edge detection of single input port 3
69h	Event link controller	Software event
6Ah	Data operation circuit	DOC data operation condition met signal
6Ch	12-bit A/D converter	S12AD1 A/D conversion end
7Eh	Compare match timer W	CMTW channel 0 compare match

Table 21.3 Correspondence between Event Signal Names Set in ELSRn.ELS[7:0] Bits and Signal Numbers (3/4)

ELS[7:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
80h	General PWM timer	GPT0 compare match A
81h		GPT0 compare match B
82h		GPT0 compare match C
83h		GPT0 compare match D
86h		GPT0 overflow
87h		GPT0 underflow
88h		GPT1 compare match A
89h		GPT1 compare match B
8Ah		GPT1 compare match C
8Bh		GPT1 compare match D
8Eh		GPT1 overflow
8Fh		GPT1 underflow
90h		GPT2 compare match A
91h		GPT2 compare match B
92h		GPT2 compare match C
93h		GPT2 compare match D
96h		GPT2 overflow
97h		GPT2 underflow
98h		GPT3 compare match A
99h		GPT3 compare match B
9Ah		GPT3 compare match C
9Bh		GPT3 compare match D
9Eh		GPT3 overflow
9Fh		GPT3 underflow
A0h	Ethernet controller	EPTPC STCA timer 0 rising edge detection
A1h		EPTPC STCA timer 1 rising edge detection
A2h		EPTPC STCA timer 2 rising edge detection
A3h		EPTPC STCA timer 3 rising edge detection
A4h		EPTPC STCA timer 4 rising edge detection
A5h		EPTPC STCA timer 5 rising edge detection
A6h		EPTPC STCA timer 0 falling edge detection
A7h		EPTPC STCA timer 1 falling edge detection
A8h		EPTPC STCA timer 2 falling edge detection
A9h		EPTPC STCA timer 3 falling edge detection
AAh		EPTPC STCA timer 4 falling edge detection
ABh		EPTPC STCA timer 5 falling edge detection

Table 21.3 Correspondence between Event Signal Names Set in ELSRn.ELS[7:0] Bits and Signal Numbers (4/4)

ELS[7:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
ACh	16-bit timer pulse unit	TPU0 compare match A
ADh		TPU0 compare match B
AEh		TPU0 compare match C
AFh		TPU0 compare match D
B0h		TPU0 overflow
B1h		TPU1 compare match A
B2h		TPU1 compare match B
B3h		TPU1 overflow
B4h		TPU1 underflow
B5h		TPU2 compare match A
B6h		TPU2 compare match B
B7h		TPU2 overflow
B8h		TPU2 underflow
B9h		TPU3 compare match A
BAh		TPU3 compare match B
BBh		TPU3 compare match C
BCh		TPU3 compare match D
BDh		TPU3 overflow
Settings other than above are prohibited.		

21.2.3 Event Link Option Setting Register A (ELOPA)

Address(es): 0008 B11Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	MTU3MD[1:0]		—	—	—	—	MTU0MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MTU0MD[1:0]	MTU0 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*1 1 1: Event is disabled.	R/W
b5 to b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7, b6	MTU3MD[1:0]	MTU3 Operation Select	b7 b6 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*2 1 1: Event is disabled.	R/W

Note 1. The MTU0.TCNT value is captured into MTU0.TGRA.
Note 2. The MTU3.TCNT value is captured into MTU3.TGRA.

ELOPA determines the operation of MTU0 and MTU3 when an event is input. The event should be disabled when the ELC function is not used.

21.2.4 Event Link Option Setting Register B (ELOPB)

Address(es): 0008 B120h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MTU4MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MTU4MD[1:0]	MTU4 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*1 1 1: Event is disabled.	R/W
b7 to b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. The MTU4.TCNT value is captured into MTU4.TGRA.

ELOPB determines the operation of MTU4 when an event is input. The event should be disabled when the ELC function is not used.

21.2.5 Event Link Option Setting Register C (ELOPC)

Address(es): 0008 B121h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMT1MD[1:0]		—	—
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b3, b2	CMT1MD[1:0]	CMT1 Operation Select	b3 b2 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event is disabled.	R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

ELOPC determines the operation of CMT1 when an event is input. The event should be disabled when the ELC function is not used.

21.2.6 Event Link Option Setting Register D (ELOPD)

Address(es): 0008 B122h

	b7	b6	b5	b4	b3	b2	b1	b0
	TMR3MD[1:0]		TMR2MD[1:0]		TMR1MD[1:0]		TMR0MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TMR0MD[1:0]	TMR0 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event is disabled.	R/W
b3, b2	TMR1MD[1:0]	TMR1 Operation Select	b3 b2 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event is disabled.	R/W
b5, b4	TMR2MD[1:0]	TMR2 Operation Select	b5 b4 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event is disabled.	R/W
b7, b6	TMR3MD[1:0]	TMR3 Operation Select	b7 b6 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event is disabled.	R/W

ELOPD determines the operation of TMR0 to TMR3 when an event is input. The event should be disabled when the ELC function is not used.

21.2.7 Event Link Option Setting Register F (ELOPF)

Address(es): 0008 B13Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	TPU3MD[1:0]		TPU2MD[1:0]		TPU1MD[1:0]		TPU0MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TPU0MD[1:0]	TPU0 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*1 1 1: Event is disabled.	R/W
b3, b2	TPU1MD[1:0]	TPU1 Operation Select	b3 b2 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*2 1 1: Event is disabled.	R/W
b5, b4	TPU2MD[1:0]	TPU2 Operation Select	b5 b4 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*3 1 1: Event is disabled.	R/W
b7, b6	TPU3MD[1:0]	TPU3 Operation Select	b7 b6 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*4 1 1: Event is disabled.	R/W

Note 1. The TPU0.TCNT value is captured into TPU0.TGRA.
Note 2. The TPU1.TCNT value is captured into TPU1.TGRA.
Note 3. The TPU2.TCNT value is captured into TPU2.TGRA.
Note 4. The TPU3.TCNT value is captured into TPU3.TGRA

ELOPF determines the operation of TPU0 to TPU3 when an event is input. The event should be disabled when the ELC function is not used.

21.2.8 Event Link Option Setting Register H (ELOPH)

Address(es): 0008 B141h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMTW0MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CMTW0MD[1:0]	CMTW Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event is disabled.	R/W
b7 to b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

ELOPH determines the operation of CMTW0 when an event is input. The event should be disabled when the ELC function is not used.

21.2.9 Event Link Option Setting Register I (ELOPI)

Address(es): 0008 B142h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	GPT1MD[2:0]			—	GPT0MD[2:0]		
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	GPT0MD[2:0]	GPT0 Operation Select	b2 b0 0 0 0: Counting is started. 0 0 1: Counting is restarted. 0 1 0: Counting is stopped. 0 1 1: Input capture*1 1 1 1: Event is disabled. Settings other than above are prohibited.	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b6 to b4	GPT1MD[2:0]	GPT1 Operation Select	b6 b4 0 0 0: Counting is started. 0 0 1: Counting is restarted. 0 1 0: Counting is stopped. 0 1 1: Input capture*2 1 1 1: Event is disabled. Settings other than above are prohibited.	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

Note 1. The GPT0.GTCNT value is captured into GPT0.GTCCRA.
Note 2. The GPT1.GTCNT value is captured into GPT1.GTCCRA.

ELOPI determines the operation of GPT0 and GPT1 when an event is input. The event should be disabled when the ELC function is not used.

21.2.10 Event Link Option Setting Register J (ELOPJ)

Address(es): 0008 B143h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	GPT3MD[2:0]			—	GPT2MD[2:0]		
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	GPT2MD[2:0]	GPT2 Operation Select	b2 b0 0 0 0: Counting is started. 0 0 1: Counting is restarted. 0 1 0: Counting is stopped. 0 1 1: Input capture*1 1 1 1: Event is disabled. Settings other than above are prohibited.	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b6 to b4	GPT3MD[2:0]	GPT3 Operation Select	b6 b4 0 0 0: Counting is started. 0 0 1: Counting is restarted. 0 1 0: Counting is stopped. 0 1 1: Input capture*2 1 1 1: Event is disabled. Settings other than above are prohibited.	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

Note 1. The GPT2.GTCNT value is captured into GPT2.GTCCRA.
Note 2. The GPT3.GTCNT value is captured into GPT3.GTCCRA.

ELOPJ determines the operation of GPT2 and GPT3 when an event is input. The event should be disabled when the ELC function is not used.

21.2.11 Port Group Setting Register n (PGRn) (n = 1, 2)

Address(es): PGR1 0008 B123h, PGR2 0008 B124h

	b7	b6	b5	b4	b3	b2	b1	b0
	PGR7	PGR6	PGR5	PGR4	PGR3	PGR2	PGR1	PGR0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PGR0	Port Group Setting 0	0: The port bit is not specified as a member of the same group. 1: The port bit is specified as a member of the same group.	R/W
b1	PGR1	Port Group Setting 1		R/W
b2	PGR2	Port Group Setting 2		R/W
b3	PGR3	Port Group Setting 3		R/W
b4	PGR4	Port Group Setting 4		R/W
b5	PGR5	Port Group Setting 5		R/W
b6	PGR6	Port Group Setting 6		R/W
b7	PGR7	Port Group Setting 7		R/W

PGRn specifies a group for I/O port bits. PGRn specifies each port bit in the same eight I/O ports as the member of a group. One to eight port bits can be specified as the members of the same group as required. Table 21.4 shows the PGRn register and corresponding ports.

Table 21.4 Registers Related to Port Groups and Corresponding Port Numbers

Port Number	Port Group Setting Register (PGR)	Port Group Control Register (PGC)	Port Buffer Register (PDBF)
Port B	PGR1 register	PGC1 register	PDBF1 register
Port E	PGR2 register	PGC2 register	PDBF2 register

21.2.12 Port Group Control Register n (PGCn) (n = 1, 2)

Address(es): PGC1 0008 B125h, PGC2 0008 B126h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PGCO[2:0]			—	PGCOVE	PGCI[1:0]	
Value after reset:	1	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PGCI[1:0]	Event Output Edge Select	b1 b0 0 0: Event is generated upon detection of the rising edge of the external input signal. 0 1: Event is generated upon detection of the falling edge of the external input signal. 1 x: Event is generated upon detection of both the rising and falling edges of the external input signal.	R/W
b2	PGCOVE	PDBF Overwrite	0: Overwriting PDBFn register is disabled. 1: Overwriting PDBFn register is enabled.	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b6 to b4	PGCO[2:0]	Port Group Operation Select	b6 b4 0 0 0: Low is output when the event is input. 0 0 1: High is output when the event is input. 0 1 0: The toggled (inverted) value is output when the event is input. 0 1 1: The buffer value is output when the event is input. 1 x x: The bit value is rotated out in the group (from MSB to LSB) when the event is input.	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

x: Don’t care

For the output port group, PGCn specifies the form of outputting the signal externally via the port when the event signal is input. For the input port group, PGCn enables/disables overwriting of PDBF and specifies the conditions of event generation (edge of the externally input signal).
Refer to Table 21.4 for the PGCn register and corresponding ports.

21.2.13 Port Buffer Register n (PDBFn) (n = 1, 2)

Address(es): PDBF1 0008 B127h, PDBF2 0008 B128h

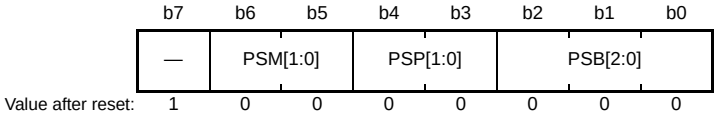
	b7	b6	b5	b4	b3	b2	b1	b0
	PDBF7	PDBF6	PDBF5	PDBF4	PDBF3	PDBF2	PDBF1	PDBF0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PDBF0	Port Buffer 0	Data is transferred between PODR and PDBF when an event is input. Write access to the bit specified as a member of the input port group is disabled. For details, refer to section 21.3, Operation.	R/W
b1	PDBF1	Port Buffer 1		R/W
b2	PDBF2	Port Buffer 2		R/W
b3	PDBF3	Port Buffer 3		R/W
b4	PDBF4	Port Buffer 4		R/W
b5	PDBF5	Port Buffer 5		R/W
b6	PDBF6	Port Buffer 6		R/W
b7	PDBF7	Port Buffer 7		R/W

PDBFn is an 8-bit readable/writable register used in combination with PGRn. Refer to section 21.3.5, I/O Port Operation upon Event Input and Event Generation for PDBFn operations. Refer to Table 21.4 for the PDBFn register and corresponding ports.

21.2.14 Event Link Port Setting Register n (PELn) (n = 0 to 3)

Address(es): PEL0 0008 B129h, PEL1 0008 B12Ah, PEL2 0008 B12Bh, PEL3 0008 B12Ch



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	PSB[2:0]	Bit Number Specification	A bit number in eight I/O ports is specified.	R/W
b4, b3	PSP[1:0]	Port Number Specification	b4 b3 0 0: Setting disabled 0 1: Port B (corresponding to PGR1) 1 0: Port E (corresponding to PGR2) 1 1: Setting prohibited	R/W
b6, b5	PSM[1:0]	Event Link Specification	• For the output port, data to be output from the port is specified. b6 b5 0 0: Low is output when the event is input. 0 1: High is output when the event is input. 1 x: The toggled (inverted) value is output when the event is input. • For the input port, the edge on which the event is to be output is specified. b6 b5 0 0: Event is output upon detection of the rising edge. 0 1: Event is output upon detection of the falling edge. 1 x: Event is output upon detection of both the rising and falling edges.	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

x: Don’t care

PELn specifies the single port to which an event is to be linked, the port operation upon the event signal input, and the conditions of event generation. In this MCU, a total of 4 bits in port B or port E can be specified as single ports.

21.2.15 Event Link Software Event Generation Register (ELSEGR)

Address(es): 0008 B12Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	WI	WE	—	—	—	—	—	SEG
Value after reset:	1	0	1	1	1	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	SEG	Software Event Generation	0: Normal operation 1: Software event is generated.	W
b5 to b1	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b6	WE	SEG Bit Write Enable	0: Write to SEG bit is disabled. 1: Write to SEG bit is enabled.	R/W
b7	WI	ELSEGR Register Write Disable	0: Write to ELSEGR register is enabled. 1: Write to ELSEGR register is disabled.	W

The MOV instruction must be used to write to this register.

SEG Bit (Software Event Generation)

When 1 is written to this bit while the WE bit is 1, a software event is generated.
This bit is read as 0. Even if 1 is written to this bit, this bit does not become 1.

WE Bit (SEG Bit Write Enable)

The SEG bit can be written to only when the WE bit is 1.
[Setting condition]
If 1 is written to this bit while the WI bit is 0, this bit becomes 1.
[Clearing condition]
If 0 is written to this bit while the WI bit is 0, this bit becomes 0.

WI Bit (ELSEGR Register Write Disable)

The ELSEGR register can be written to only when the value to be written to the WI bit is 0.
This bit is read as 1.

21.3 Operation

21.3.1 Relation between Interrupt Handling and Event Linking

The peripheral modules incorporated in the MCU are provided with the interrupt request status flags and the bits to enable/disable these interrupt requests. When an interrupt request is generated in a module, the corresponding interrupt request status flag is set. If the corresponding interrupt request is enabled then, the interrupt requested is issued to the CPU.

In contrast, the event link controller (ELC) uses the interrupt requests generated by various peripheral modules as event signals to connect (link) them to different modules, allowing direct cooperation between the modules without CPU intervention. Event signals can be output regardless of the setting of the corresponding interrupt request enable bit.

Figure 21.2 shows the relation between the interrupt handling and ELC.

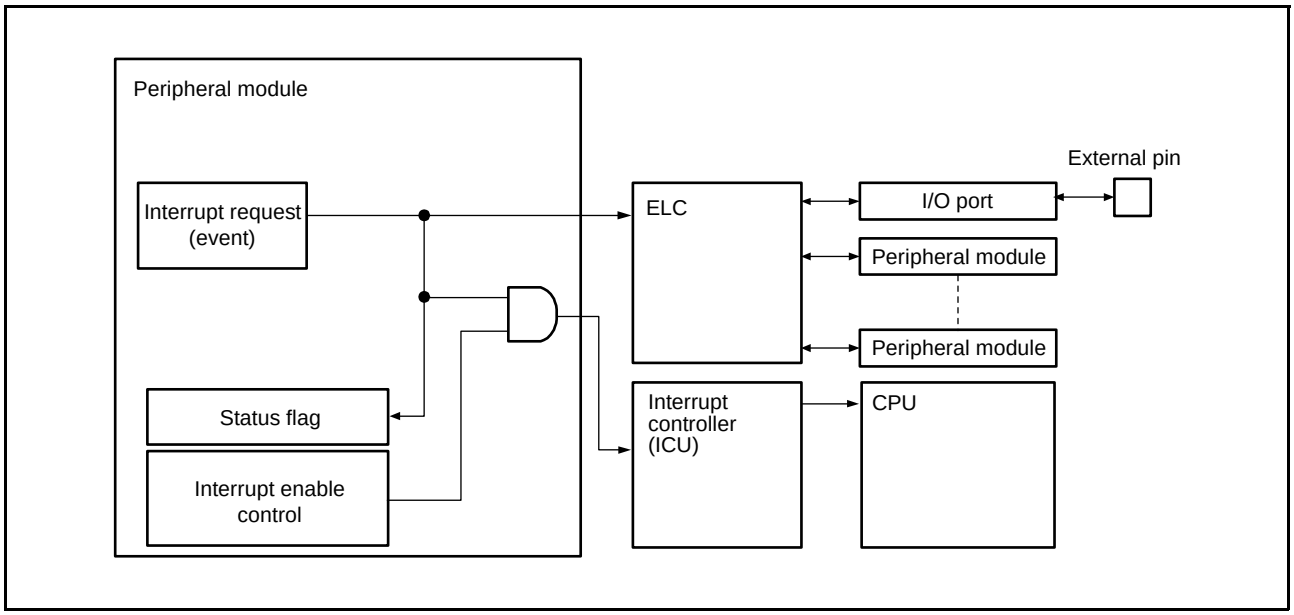


Figure 21.2 Relation between Interrupt Handling and ELC

21.3.2 Event Linkage

When an event has been set as a trigger in an event link setting register (ELSRn) and then occurs, the corresponding module is activated. Only one type of event can be connected with one module. When a module is to be activated by the ELC, the operation of the module must be set up in advance. Table 21.5 lists the operations of modules when an event is input.

Table 21.5 Operations of Modules When Event is Input

Module	Operations When Event is Input		
MTU CMT CMTW TMR TPU GPT	Each timer operates differently depending on the ELOPA to ELOPD, ELOPF, ELOPH, ELOPI, ELOPJ registers as below. • Starts counting when an event signal is input (MTU, CMT, CMTW, TMR, TPU, GPT). • Restarts counting when an event signal is input (MTU, CMT, CMTW, TMR, TPU, GPT). • Counts the input events (CMT, CMTW, TMR). • Performs input-capture operation when an event is input (MTU, TPU, GPT). • Stops counting when an event signal is input (GPT).		
A/D converter	Starts A/D conversion when an event signal is input.		
D/A converter	Starts D/A conversion when an event signal is input		
I/O ports (output)	The value of PODR (port output data register) changes when an event signal is input. (The value output from the relevant external pin changes.)	Port group	• Changes the PODR value to the specified value. • Transfers the PDBFn value to the PODR register. • Rotates out the bit value.
		Single port	Changes the PODR value to the specified value.
I/O ports (input)	When the signal value of the input pin changes	Port group	Generates an event.
		Single port	
	When an event is input	Port group	Transfers the signal value of the external pin to the PDBFn register.
		Single port	Event connection is not possible.
Clock generation circuit	Switches the clock source to the low-speed on-chip oscillator when an event signal is input.*1		
Interrupt controller	Issues an event to the CPU, starts DMAC data transfer, and starts DTC data transfer when an event signal is input.		

Note 1. The SCKCR3.CKSEL[2:0] bits are modified to 000b (LOCO) regardless of the value of the protect register (PRCR.PRC0).

21.3.3 Operation of Peripheral Timer Modules When Event is Input

The operations are performed depending on the ELOPA to ELOPD, ELOPF, ELOPH, ELOPI, and ELOPJ registers when an event is input.

(1) Count Start Operation

When an event is input, the timer starts counting, which sets the count start bit*1 in each timer control register to 1. An event that is input while the count start bit is 1 is invalid.

(2) Count Restart Operation

When an event is input, the timer counter*1 is initialized. Since the count start bit*1 in each timer control register is retained, counting is restarted when an event is input while the count start bit is set to 1.

(3) Event Counter Operation

Event input is selected as the timer clock source and the timer counts events.

(4) Input Capture Operation

When an event is input, the timer performs input-capture operation.

(5) Count Stop Operation

The counter stops counting in response to the input of an event signal.

Note 1. Refer to the register descriptions on starting the timer in the relevant peripheral timer module section.

21.3.4 Operation of A/D and D/A Converters When Event is Input

The A/D and D/A converter start A/D and D/A conversion, respectively, when the ADCSR.ADST bit and the DACR.DAOE0 bit*1 are set to 1.

Note 1. Refer to the bit descriptions in the A/D converter and D/A converter sections.

21.3.5 I/O Port Operation upon Event Input and Event Generation

The I/O port operation to be performed upon event input and the operation to generate an event can be set.

(1) Single Ports and Port Groups

There are two event link modes: event link to single ports and event link to port groups. In the former mode, events can be connected to eight I/O ports. In the latter mode, events can be connected to port groups consisting of any two or more bits in the same eight I/O ports.
A single port can be set by specifying any bit in the I/O port*1 to which an event can be connected using the PEL0 to PEL3 registers. A port group can be set by specifying any 2 or more bits in the I/O port*1 to which an event can be connected using the PGCn register. One input port group and one output port group can be set in the same I/O port.
If the I/O port bit is specified as both a single port and a member of a port group, both functions are enabled when the relevant port is input, whereas only the port group function is enabled when the relevant port is output.
Set the PDR register to select the direction of the I/O ports.

Note 1. Port B and port E

(2) Event Generation by Single Input Ports

An single input port generates an event when the signal value of the external pin connected to the relevant port changes. The event generation condition is specified using the PEL0 to PEL3 registers. An example of operation is shown in Figure 21.3.

(3) Single Output Ports Operation upon Event Input

When an event is input to a single output port, the signal of the external pin connected to the relevant port changes according to the settings of the PEL0 to PEL3 registers. This changes the signal value of the external pin connected to the relevant port. An example of operation is shown in Figure 21.3.

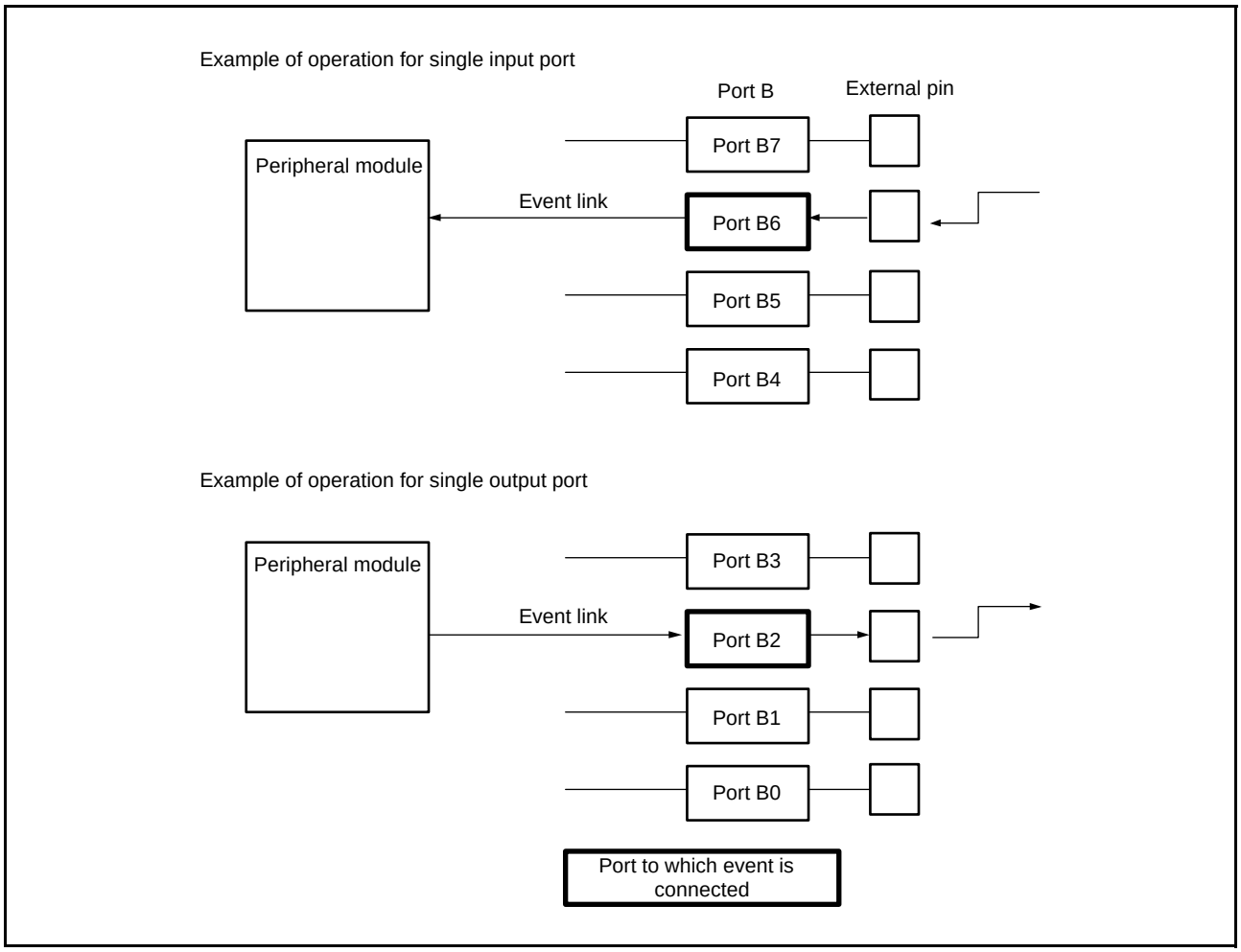


Figure 21.3 Event Linkage Related to Single Ports (Port B)

(4) Input Port Group Operation upon Event Input and Event Generation

An input port group generates an event when the signal value of any one of the external pins connected to the relevant port group changes. The event generation condition is specified using the PGCn register. When an event is input to an input port group, the signal value of the external pin upon event input is transferred to the PDBFn register. In this case, only the values of the bits specified as members of the input port group are transferred. An example of operation is shown in Figure 21.4.

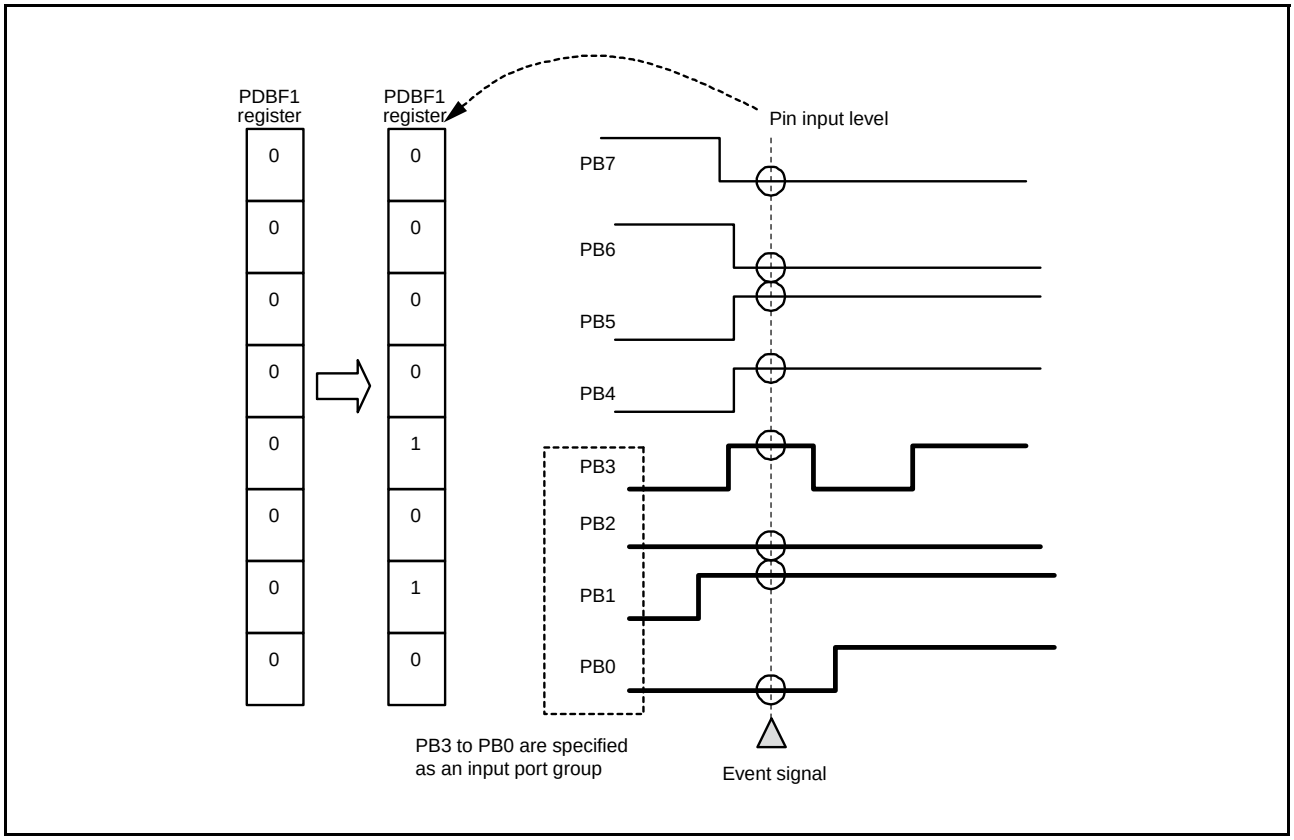


Figure 21.4 Event Linkage Related to Input Port Groups (Port B)

(5) Output Port Group Operation upon Event Input

When an event is input to an output port group, the PODR values change to the values according to the PGCn settings. An example of operation is shown in Figure 21.5.

(6) Operation of Port Buffer Registers

(a) Input Port Groups

When an event is input to an input port group, the signal value of the external pin of the bit specified as a member of the input port group is transferred to the PDBFn register. If another event is input to the input port group in this state, operations are performed depending on the PGCn.PGCOVE bit setting as described below.

- PGCn.PGCOVE = 0 (overwriting is disabled)
  If the PDBFn value that has been transferred upon the latest event input has already been read by the CPU (or transferred by the DTC), the signal value of the external pin is transferred to the PDBFn register. If not read, the signal value of the external pin is not transferred and the input event is invalid.
- PGCn.PGCOVE = 1 (overwriting is enabled)
  When another event is input to an input port group, the signal value of the external pin is transferred to the PDBFn register.

(b) Output Port Groups

If an output port group is specified so that it should output the PDBFn value, the PDBFn value is transferred to the PODR register when an event is input to the output port group. In this case, only the values of the bits specified as members of the output port group are transferred.
If an output port group is specified so that it should rotate out the bit values in the group (PGCn.PGCO[2:0] bits = 1xxb), the PDBFn data is transferred to the PODR register, and then the PODR value is rotated bit by bit from MSB to LSB. The

initial value to be output to the port group should be provided in the PDBFn register.
Examples of operation are shown in Figure 21.5 and Figure 21.6.

(7) Restrictions on Writing to PODR and PDBF Registers

When the ELCR.ELCON bit is set to 1, write access to the following registers is disabled.

- If bits are specified as members of the input port group and the event linkage is set for the port group, write access to the relevant bits in the PDBFn register is disabled.
- If port bits are specified as members of the output port group, write access to the relevant bits in the PODR register is disabled.
- If a port bit is specified as a single output port and the event linkage is set (by the ELSRn register) for the port, write access to the relevant bit in the PODR register is disabled.

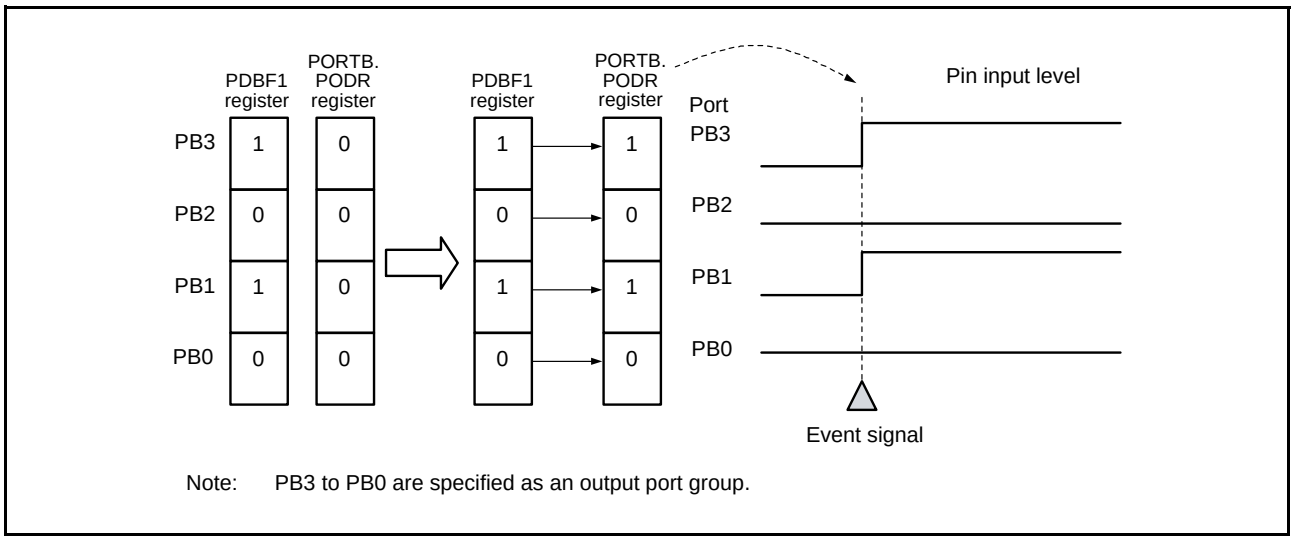


Figure 21.5 Event Linkage Related to Output Port Groups (Port B)

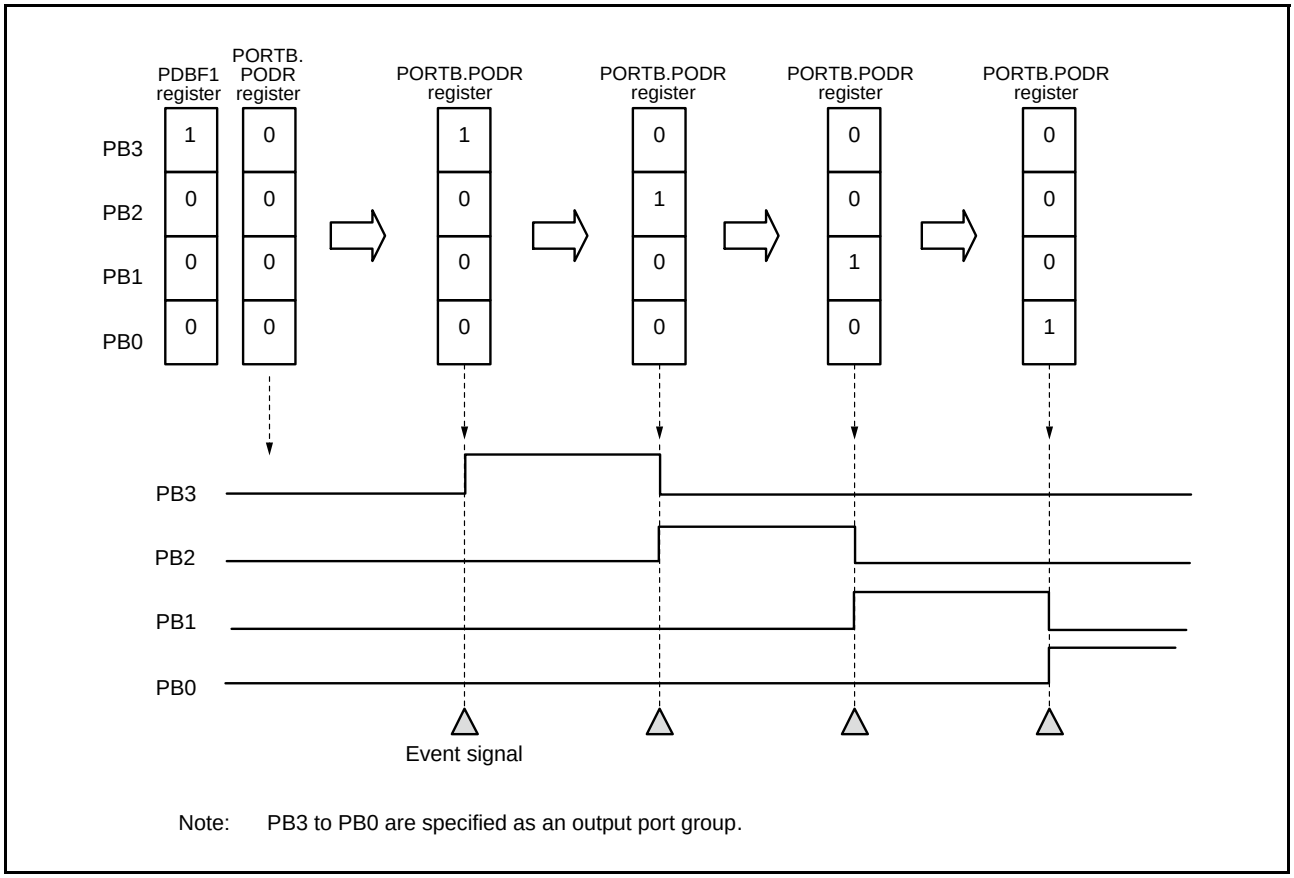


Figure 21.6 Bit-Rotating Operation of Output Port Groups (Port B)

21.3.6 Example of Procedure for Linking Events

The following describes the procedure for linking events.

1. Set the operation of the module to which an event is to be linked.
2. If events are linked to ports, set the registers corresponding to the ports as below.
   PODR: Set the initial values of the output ports.
   PDR: Set the I/O direction of the ports.
   PGRn: If ports are used as a port group, set the ports (in bit units) to be grouped.
   PGCn: Set the operation of the port group.
   PELn: If ports are used as single ports, set the ports, the operation of the ports when an event is input, and the condition when an event is generated.
3. To the ELSRn register corresponding to the module to which an event signal is to be linked, set the number of the event signal.
4. If events are to be linked to timer modules, set the ELOPA to ELOPD, ELOPF, ELOPH, ELOPI, and ELOPJ registers corresponding to the timers as required.
5. Set the ELCR.ELCON bit to 1, which enables linkage of all the events.
6. Set the operation of the module from which an event is output, and activate the module. This allows the event output from the module to start the module to which an event is linked as preset.
7. To stop event linkage of independent modules, set 00h to the ELSRn.ELS[7:0] bits corresponding to the modules. To stop linkage of all the events, set the ELCR.ELCON bit to 0.

Note: If event output from the RTC is to be used, make the ELC settings after the RTC settings (initialization, time setting, etc.). Unintended events may be generated if RTC settings are made after the ELC settings.

Note: When using event output from the LVD, set the LVD and then the ELC. Set the corresponding ELSRn register to 00h and then disable the LVD.

21.4 Usage Notes

21.4.1 Setting ELSRn Register

(1) Setting ELSR18 and ELSR19 Registers

Specify an event number from among 63h to 6Ah. Do not set other settings.

(2) Setting ELSR24, ELSR25, ELSR26, and ELSR27 Registers

Do not set the DOC data operation condition met signal (6Ah).

21.4.2 Setting Bit-Rotating Operation of Output Port Groups

When the values of the PDBFn register are changed in the bit-rotating operation mode of the output port group, set the ELSRn register again. When events are used during bit-rotating operation, generate an event after an interval of one PCLKB cycle. If not, the normal operation cannot be provided.

21.4.3 Linking DMAC/DTC Transfer End Signals as Events

When linking the DMAC/DTC transfer end signals as events, do not set the same peripheral module as the DMAC/DTC transfer destination and event link destination. If set, the peripheral module might be started before DMAC/DTC transfer to the peripheral module is completed.

21.4.4 Setting Clocks

To link events, it is necessary for the ELC and the related modules to be enabled. The modules cannot operate if the related modules are in the module stop state or in the specific low power consumption mode in which the module is stopped (all-module clock stop mode, software standby mode, or deep software standby mode).

21.4.5 Module Stop Function Setting

ELC operation can be disabled or enabled using module stop control register B (MSTPCRB). After reset is released, the ELC function is disabled. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

22. I/O Ports

22.1 Overview

The pins of an I/O port function as general I/O port pins, I/O pins for peripheral modules, interrupt input pins, or bus control pins.

Each pin is also configurable as an I/O pin of a peripheral module or an input pin for an interrupt. All pins function as input pins immediately after a reset, and pin functions are switched by register settings. The setting of each pin is specified by the registers for the corresponding I/O port and peripheral modules.

Each port has the port direction register (PDR) that selects input or output direction, the port output data register (PODR) that holds data for output, the port input register (PIDR) that indicates the pin states, the open-drain control register y (ODRy, y = 0, 1) that selects the output type of each pin, the pull-up resistor control register (PCR) that controls on/off of the input pull-up MOS, the drive capacity control register (DSCR) that selects the drive capacity, and the port mode register (PMR) that specifies the pin function of each port.

For details on PMR, refer to section 23, Multi-Function Pin Controller (MPC).

The configuration of the I/O ports differs depending on the package. Table 22.1 shows the specifications of I/O ports, and Table 22.2 lists the port functions.

Table 22.1 Specifications of I/O Ports

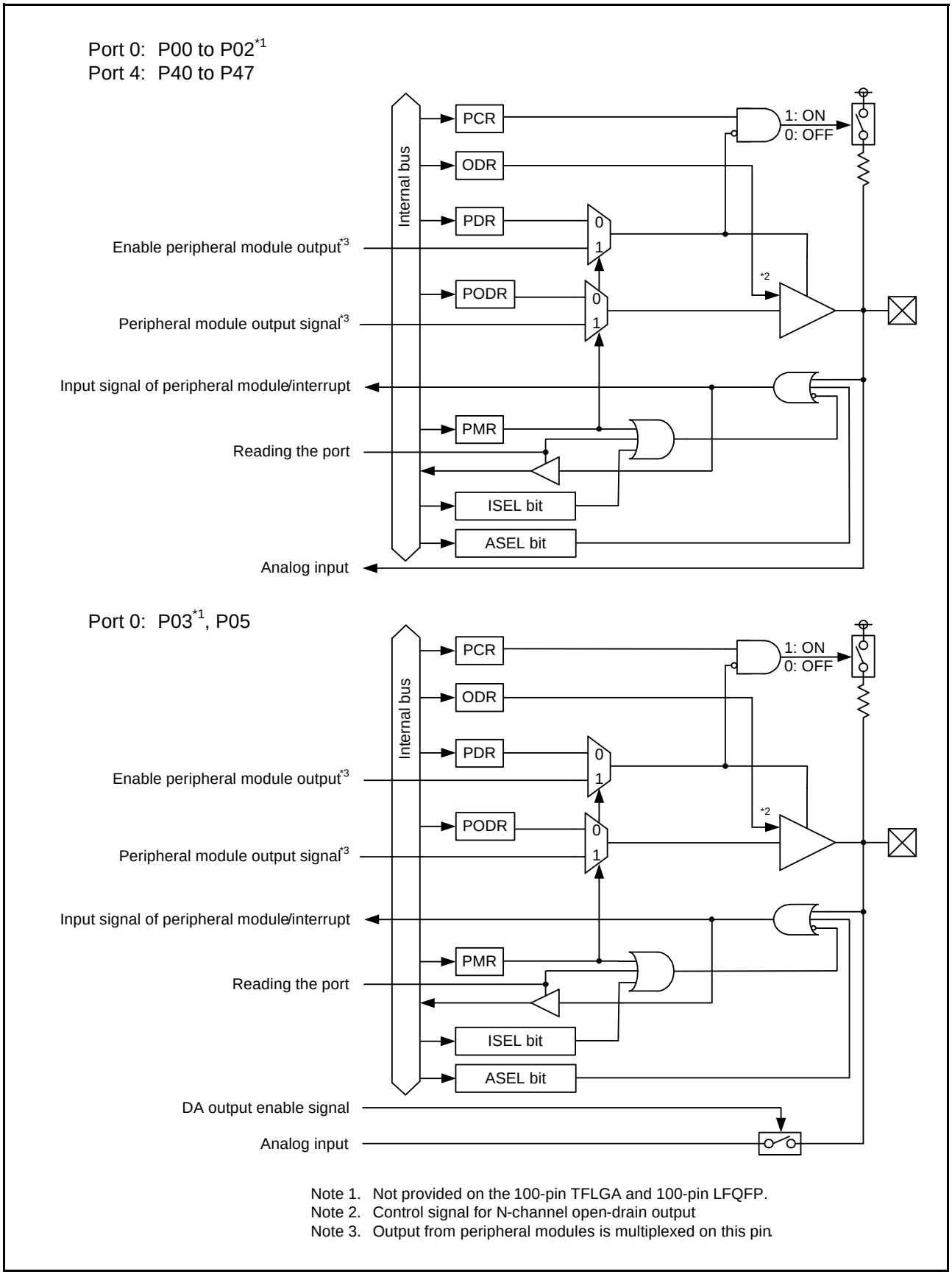
Port	Package		Package		Package	
	177 or 176 Pins	Number of Pin	145 or 144 Pins	Number of Pin	100 Pins	Number of Pin
PORT0	P00 to P03, P05, P07	6	P00 to P03, P05, P07	6	P05, P07	2
PORT1	P10 to P17	8	P12 to P17	6	P12 to P17	6
PORT2	P20 to P27	8	P20 to P27	8	P20 to P27	8
PORT3	P30 to P37	8	P30 to P37	8	P30 to P37	8
PORT4	P40 to P47	8	P40 to P47	8	P40 to P47	8
PORT5	P50 to P53	4	P50 to P56	7	P50 to P55	6
PORT6	P60 to P67	8	P60 to P67	8	Not provided	0
PORT7	P70 to P77	8	P70 to P77	8	Not provided	0
PORT8	P80 to P83, P86, P87	6	P80 to P83, P86, P87	6	Not provided	0
PORT9	P90 to P97	8	P90 to P93	4	Not provided	0
PORTA	PA0 to PA7	8	PA0 to PA7	8	PA0 to PA7	8
PORTB	PB0 to PB7	8	PB0 to PB7	8	PB0 to PB7	8
PORTC	PC0 to PC7	8	PC0 to PC7	8	PC0 to PC7	8
PORTD	PD0 to PD7	8	PD0 to PD7	8	PD0 to PD7	8
PORTE	PE0 to PE7	8	PE0 to PE7	8	PE0 to PE7	8
PORTF	PF0 to PF5	6	PF5	1	Not provided	0
PORTG	PG0 to PG7	8	Not provided	0	Not provided	0
PORTJ	PJ3, PJ5	2	PJ3, PJ5	2	PJ3	1
	Total of pins	128	Total of pins	112	Total of pins	79

Table 22.2 Port Functions

Port	Pin	Input Pull-up	Open-Drain Output	Driving Ability Switching	5-V Tolerant
PORT0	P00 to P02	✓	✓	✓	—
	P03, P05	✓	✓	Fixed to high driving ability output	—
	P07	✓	✓	Fixed to high driving ability output	✓
PORT1	P10	✓	✓	Fixed to high driving ability output	—
	P11 to P17	✓	✓	Fixed to high driving ability output	✓
PORT2	P20, P21	✓	✓	Fixed to high driving ability output	✓
	P22 to P26	✓	✓	Fixed to high driving ability output	—
	P27	✓	✓	✓	—
PORT3	P30 to P33	✓	✓	Fixed to high driving ability output	✓
	P34, P37	✓	✓	Fixed to high driving ability output	—
	P35	—	—	—	—
	P36	✓	✓	Fixed to normal output	—
PORT4	P40 to P47	✓	✓	Fixed to normal output	—
PORT5	P50 to P52, P56	✓	✓	✓	—
	P53 to P55	✓	✓	Fixed to high driving ability output	—
PORT6	P60 to P66	✓	✓	Fixed to high driving ability output	—
	P67	✓	✓	Fixed to high driving ability output	✓
PORT7	P70 to P77	✓	✓	Fixed to high driving ability output	—
PORT8	P80 to P83, P86, P87	✓	✓	Fixed to high driving ability output	—
PORT9	P90 to P97	✓	✓	✓	—
PORTA	PA0 to PA7	✓	✓	✓	—
PORTB	PB0 to PB7	✓	✓	✓	—
PORTC	PC0 to PC3	✓	✓	✓	✓
	PC4 to PC7	✓	✓	✓	—
PORTD	PD0 to PD7	✓	✓	✓	—
PORTE	PE0 to PE7	✓	✓	✓	—
PORTF	PF0 to PF5	✓	✓	Fixed to high driving ability output	—
PORTG	PG0, PG1	✓	✓	✓	—
	PG2 to PG7	✓	✓	Fixed to high driving ability output	—
PORTJ	PJ3, PJ5	✓	✓	Fixed to high driving ability output	—

Specifying input pull-up, open-drain output, switching of driving ability, or 5-V tolerance is available for other signals on pins that also function as general I/O pins.

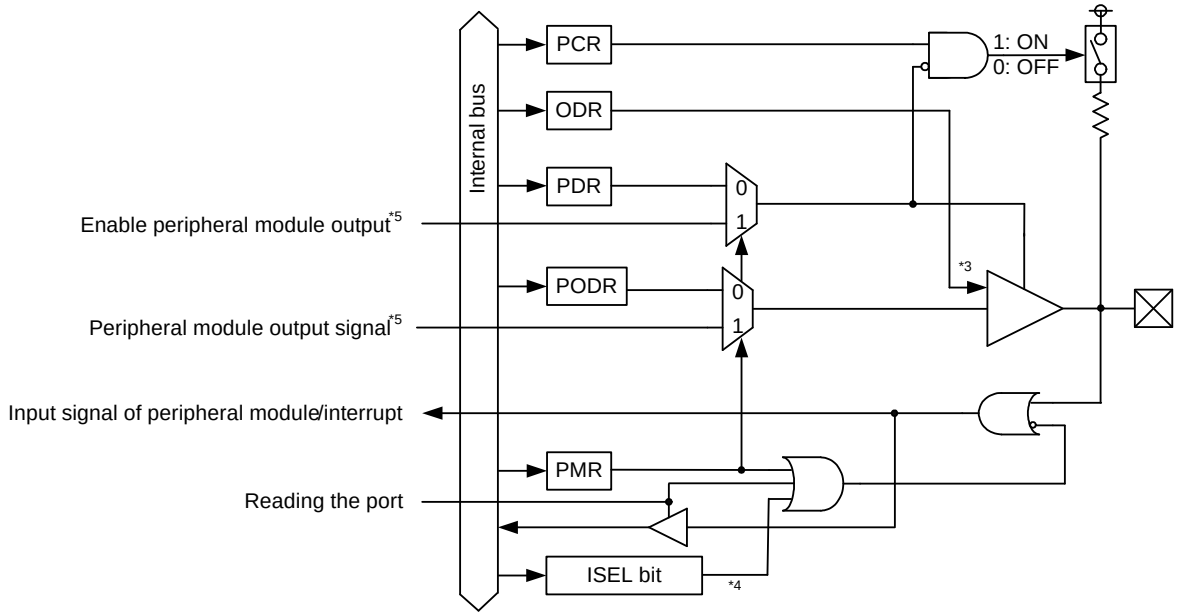
22.2 I/O Port Configuration



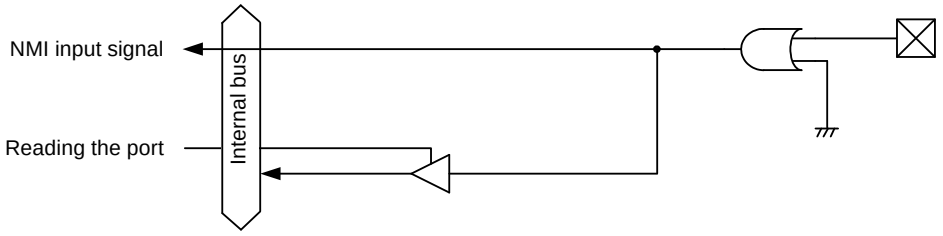
Note 1. Not provided on the 100-pin TFLGA and 100-pin LFQFP.
Note 2. Control signal for N-channel open-drain output
Note 3. Output from peripheral modules is multiplexed on this pin.

Figure 22.1 I/O Port Configuration (1)

Port 0: P07
Port 1: P11[*1, *2], P14 to P17
Port 2: P20 to P23
Port 3: P30 to P34, P36, P37
Port 7: P70[*2]
Port 8: P80 to P83[*2], P86[*2], P87[*2]
Port F: PF0 to PF4[*1, *2], PF5[*2]
Port J: PJ3, PJ5[*2]



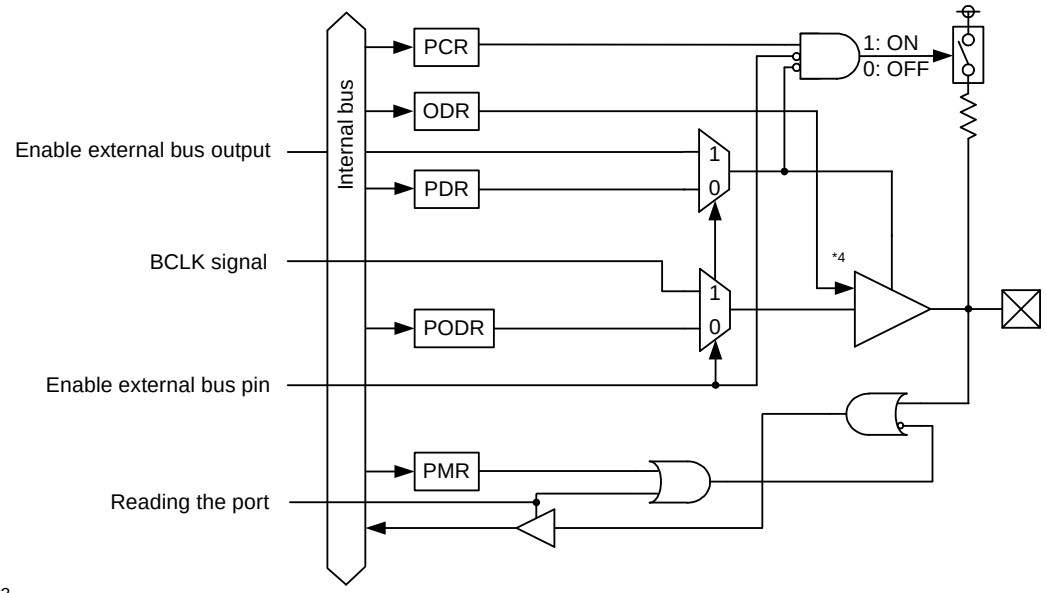
Port 3: P35



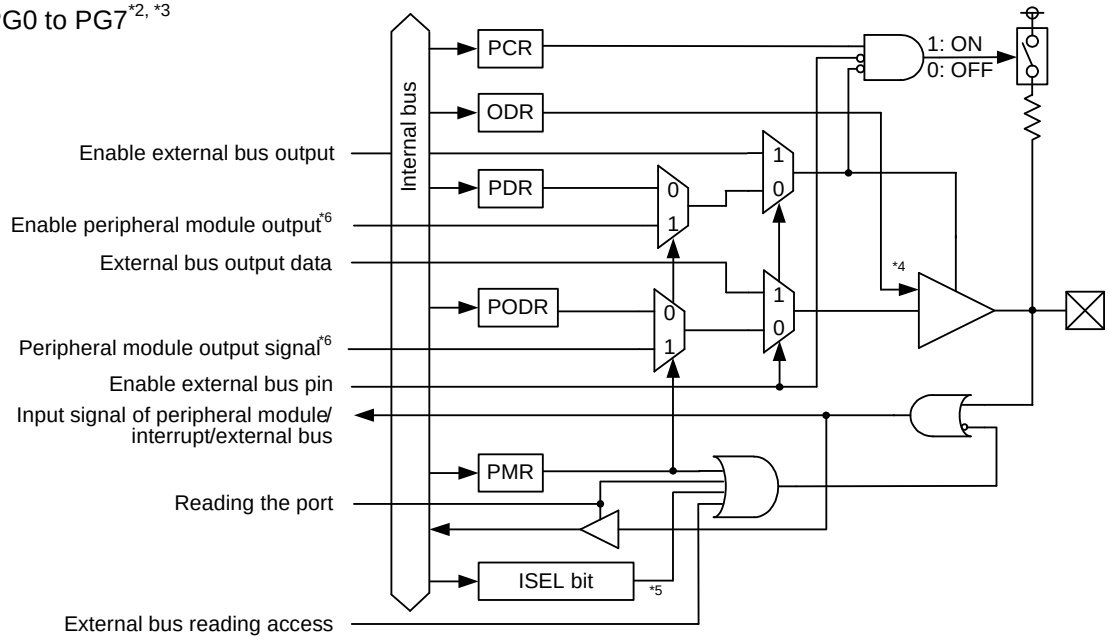
Note 1. Not provided on the 145-pin TFLGA and 144-pin LFQFP.
Note 2. Not provided on the 100-pin TFLGA and 100-pin LFQFP.
Note 3. Control signal for N-channel open-drain output
Note 4. An external interrupt function is multiplexed on this pin.
Note 5. Output from peripheral modules is multiplexed on this pin.

Figure 22.2 I/O Port Configuration (2)

Port 5: P53



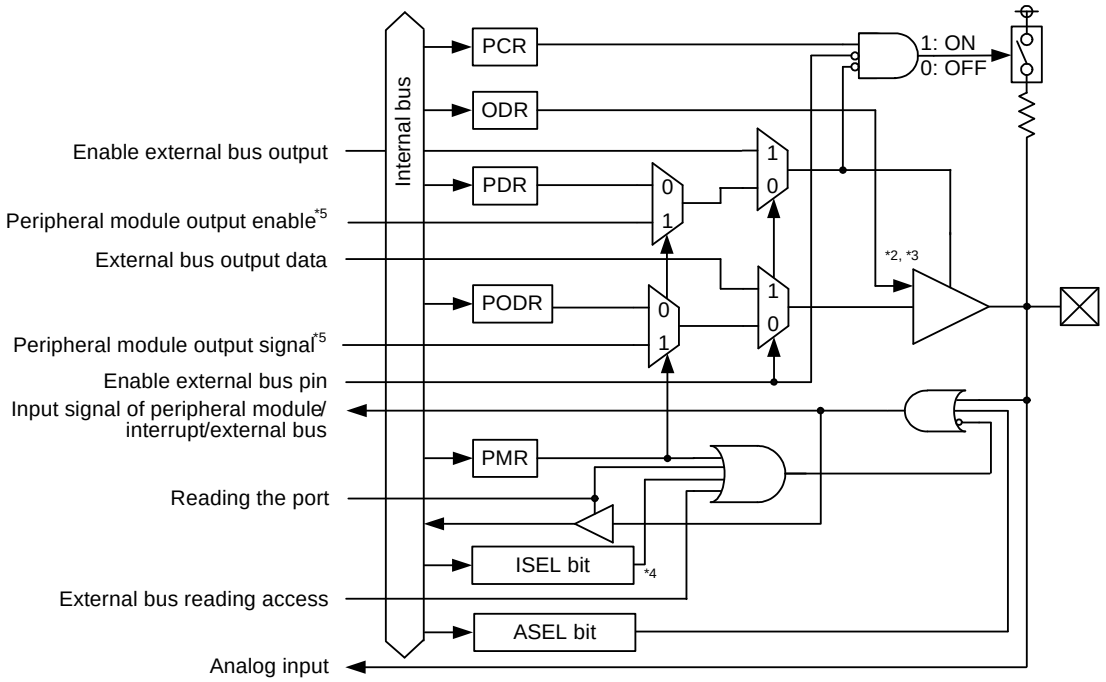
Port 1: P10*2, *3, P12, P13
Port 2: P24 to P27
Port 5: P50 to P52, P54*1, P55*1, P56*1, *3
Port 6: P60 to P67*3
Port 7: P71 to P77*3
Port 9: P94 to P97*2, *3
Port A: PA0 to PA7
Port B: PB0 to PB7
Port C: PC0 to PC7
Port G: PG0 to PG7*2, *3



Note 1. Not provided on the 177-pin TFLGA, 176-pin LFBGA, and 176-pin LFQFP.
Note 2. Not provided on the 145-pin TFLGA and 144-pin LFQFP.
Note 3. Not provided on the 100-pin TFLGA and 100-pin LFQFP.
Note 4. Control signal for N-channel open-drain output
Note 5. An external interrupt function is multiplexed on this pin.
Note 6. Output from peripheral modules is multiplexed on this pin.

Figure 22.3 I/O Port Configuration (3)

Port 9: P90 to P93[*1]
Port D: PD0 to PD7
Port E: PE0 to PE7



Note 1. Not provided on the 100-pin TFLGA and 100-pin LFQFP.
Note 2. Control signal for N-channel open-drain output
Note 3. Control signal for P-channel open-drain output (only for PE1)
Note 4. An external interrupt function is multiplexed on this pin.
Note 5. Output from peripheral modules is multiplexed on this pin.

Figure 22.4 I/O Port Configuration (4)

22.3 Register Descriptions

22.3.1 Port Direction Register (PDR)

Address(es): PORT0.PDR 0008 C000h, PORT1.PDR 0008 C001h, PORT2.PDR 0008 C002h, PORT3.PDR 0008 C003h, PORT4.PDR 0008 C004h, PORT5.PDR 0008 C005h, PORT6.PDR 0008 C006h, PORT7.PDR 0008 C007h, PORT8.PDR 0008 C008h, PORT9.PDR 0008 C009h, PORTA.PDR 0008 C00Ah, PORTB.PDR 0008 C00Bh, PORTC.PDR 0008 C00Ch, PORTD.PDR 0008 C00Dh, PORTE.PDR 0008 C00Eh, PORTF.PDR 0008 C00Fh, PORTG.PDR 0008 C010h, PORTJ.PDR 0008 C012h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 I/O Select	0: Input (Functions as an input pin.) 1: Output (Functions as an output pin.)	R/W
b1	B1	Pm1 I/O Select		R/W
b2	B2	Pm2 I/O Select		R/W
b3	B3	Pm3 I/O Select		R/W
b4	B4	Pm4 I/O Select		R/W
b5	B5	Pm5 I/O Select		R/W
b6	B6	Pm6 I/O Select		R/W
b7	B7	Pm7 I/O Select		R/W

m = 0 to 9, A to G, and J

PDR is a register which is used to select the input or output direction for individual pins of the corresponding port when the pins are configured as the general I/O pins.
Each bit of PORTm.PDR corresponds to each pin of port m; I/O direction can be specified in 1-bit units.
However, the bits that correspond to port m on the 176-pin product but do not exist on a product with fewer pins (excluding port 54 to port 56) are reserved. When writing, write 1 (output) to these bits. Also write 1 (output) to the bits of port 54 to port 56 on the 176-pin product and write 1 (output) to the bit of port 56 on the 100-pin product.
The B5 bit in PORT3.PDR is reserved, because the P35 pin is input only.
The bit corresponding to a pin that does not exist is also reserved. A reserved bit is always read as 0. The write value should always be 0.

22.3.2 Port Output Data Register (PODR)

Address(es): PORT0.PODR 0008 C020h, PORT1.PODR 0008 C021h, PORT2.PODR 0008 C022h, PORT3.PODR 0008 C023h, PORT4.PODR 0008 C024h, PORT5.PODR 0008 C025h, PORT6.PODR 0008 C026h, PORT7.PODR 0008 C027h, PORT8.PODR 0008 C028h, PORT9.PODR 0008 C029h, PORTA.PODR 0008 C02Ah, PORTB.PODR 0008 C02Bh, PORTC.PODR 0008 C02Ch, PORTD.PODR 0008 C02Dh, PORTE.PODR 0008 C02Eh, PORTF.PODR 0008 C02Fh, PORTG.PODR 0008 C030h, PORTJ.PODR 0008 C032h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Output Data Store	0: Low output 1: High output	R/W
b1	B1	Pm1 Output Data Store		R/W
b2	B2	Pm2 Output Data Store		R/W
b3	B3	Pm3 Output Data Store		R/W
b4	B4	Pm4 Output Data Store		R/W
b5	B5	Pm5 Output Data Store		R/W
b6	B6	Pm6 Output Data Store		R/W
b7	B7	Pm7 Output Data Store		R/W

m = 0 to 9, A to G, and J

PODR is a register which holds the data to be output from the pins used for general I/O.
Bits that correspond to port m on the 176-pin product but do not exist on a product with fewer pins are reserved. When writing, write 0 (low output) to these bits.
The B5 bit in PORT3.PDR is reserved, because the P35 pin is input only. Data is not output from the corresponding pins even if these bits are set.
The bit corresponding to a pin that does not exist is reserved. A reserved bit is always read as 0. The write value should always be 0.

22.3.3 Port Input Register (PIDR)

Address(es): PORT0.PIDR 0008 C040h, PORT1.PIDR 0008 C041h, PORT2.PIDR 0008 C042h, PORT3.PIDR 0008 C043h, PORT4.PIDR 0008 C044h, PORT5.PIDR 0008 C045h, PORT6.PIDR 0008 C046h, PORT7.PIDR 0008 C047h, PORT8.PIDR 0008 C048h, PORT9.PIDR 0008 C049h, PORTA.PIDR 0008 C04Ah, PORTB.PIDR 0008 C04Bh, PORTC.PIDR 0008 C04Ch, PORTD.PIDR 0008 C04Dh, PORTE.PIDR 0008 C04Eh, PORTF.PIDR 0008 C04Fh, PORTG.PIDR 0008 C050h, PORTJ.PIDR 0008 C052h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0	0: Low input 1: High input	R
b1	B1	Pm1		R
b2	B2	Pm2		R
b3	B3	Pm3		R
b4	B4	Pm4		R
b5	B5	Pm5		R
b6	B6	Pm6		R
b7	B7	Pm7		R

m = 0 to 9, A to G, J

PIDR is a register which reflects individual pin states of the port.

The pin states of port m can be read with the PORTm.PIDR, regardless of the values of PORTm.PDR and PORTm.PMR.

The NMI pin state is reflected in the P35 bit. However, the states of pins when the PmnPFS.ASEL bit is set to 1 cannot be read.

The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as undefined, and cannot be modified.

22.3.4 Port Mode Register (PMR)

Address(es): PORT0.PMR 0008 C060h, PORT1.PMR 0008 C061h, PORT2.PMR 0008 C062h, PORT3.PMR 0008 C063h, PORT4.PMR 0008 C064h, PORT5.PMR 0008 C065h, PORT6.PMR 0008 C066h, PORT7.PMR 0008 C067h, PORT8.PMR 0008 C068h, PORT9.PMR 0008 C069h, PORTA.PMR 0008 C06Ah, PORTB.PMR 0008 C06Bh, PORTC.PMR 0008 C06Ch, PORTD.PMR 0008 C06Dh, PORTE.PMR 0008 C06Eh, PORTF.PMR 0008 C06Fh, PORTG.PMR 0008 C070h, PORTJ.PMR 0008 C072h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Pin Mode Control	0: Uses the pin as a general I/O pin. 1: Uses the pin as an I/O port for peripheral modules.	R/W
b1	B1	Pm1 Pin Mode Control		R/W
b2	B2	Pm2 Pin Mode Control		R/W
b3	B3	Pm3 Pin Mode Control		R/W
b4	B4	Pm4 Pin Mode Control		R/W
b5	B5	Pm5 Pin Mode Control		R/W
b6	B6	Pm6 Pin Mode Control		R/W
b7	B7	Pm7 Pin Mode Control		R/W

m = 0 to 9, A to G, and J

PMR is a register which specifies the function of the pins of the port.
Each bit of PORTm.PMR corresponds to each pin of port m; pin function can be specified in 1-bit units.
However, bits that correspond to port m on the 176-pin product but do not exist on a product with fewer pins (except for ports 54 to 56) are reserved. When writing, write 0 (general I/O port) to these bits. For ports 54 to 56, write 0 (general I/O port) to the bits corresponding to ports 54 to 56 on the 176-pin product; write 0 (general I/O port) to the bit corresponding to port 56 on the 100-pin product.
The bit corresponding to a pin that does not exist is reserved. A reserved bit is always read as 0. The write value should always be 0.

22.3.5 Open-Drain Control Register 0 (ODR0)

Address(es): PORT0.ODR0 0008 C080h, PORT1.ODR0 0008 C082h, PORT2.ODR0 0008 C084h, PORT3.ODR0 0008 C086h, PORT4.ODR0 0008 C088h, PORT5.ODR0 0008 C08Ah, PORT6.ODR0 0008 C08Ch, PORT7.ODR0 0008 C08Eh, PORT8.ODR0 0008 C090h, PORT9.ODR0 0008 C092h, PORTA.ODR0 0008 C094h, PORTB.ODR0 0008 C096h, PORTC.ODR0 0008 C098h, PORTD.ODR0 0008 C09Ah, PORTE.ODR0 0008 C09Ch, PORTF.ODR0 0008 C09Eh, PORTG.ODR0 0008 C0A0h, PORTJ.ODR0 0008 C0A4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	B6	—	B4	B3	B2	—	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Output Type Select	For pins other than the port PE1 pin	R/W
b1	—	Reserved	Odd bit / Even bit	R/W
b2	B2	Pm1 Output Type Select	X 0: CMOS output	R/W
b3	B3*1	PE1 Output Type Select	X 1: NMOS open-drain output	R/W
b4	B4	Pm2 Output Type Select	(b1, b3, b5, b7: Reserved)	R/W
b5	—	Reserved	For port PE1 pin	R/W
b6	B6	Pm3 Output Type Select	b3 b2	R/W
b7	—	Reserved	0 0: CMOS output	R/W
			0 1: NMOS open-drain output	
			1 0: PMOS open-drain output	
			1 1: Setting prohibited	

m = 0 to 9, A to G, and J

Note 1. The bit for a pin other than PE1 is reserved.

ODR0 is a register which is used to select an output type for the pins of the port.

In the registers other than PORTE.ODR0, the odd bits (b1, b3, b5, and b7) are reserved.

However, the output type of the port PE1 pin is specified by the combination of b3 and b2.

Bits that correspond to port m on the 176-pin product but do not exist on a product with fewer pins are reserved. When writing, write 0 (CMOS output) to these bits.

The bit corresponding to a pin that does not exist is also reserved. A reserved bit is always read as 0. The write value should always be 0.

22.3.6 Open-Drain Control Register 1 (ODR1)

Address(es): PORT0.ODR1 0008 C081h, PORT1.ODR1 0008 C083h, PORT2.ODR1 0008 C085h, PORT3.ODR1 0008 C087h, PORT4.ODR1 0008 C089h, PORT5.ODR1 0008 C08Bh, PORT6.ODR1 0008 C08Dh, PORT7.ODR1 0008 C08Fh, PORT8.ODR1 0008 C091h, PORT9.ODR1 0008 C093h, PORTA.ODR1 0008 C095h, PORTB.ODR1 0008 C097h, PORTC.ODR1 0008 C099h, PORTD.ODR1 0008 C09Bh, PORTE.ODR1 0008 C09Dh, PORTF.ODR1 0008 C09Fh, PORTG.ODR1 0008 C0A1h, PORTJ.ODR1 0008 C0A5h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	B6	—	B4	—	B2	—	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm4 Output Type Select	0: CMOS output 1: NMOS open-drain output	R/W
b1	—	Reserved		R/W
b2	B2	Pm5 Output Type Select		R/W
b3	—	Reserved		R/W
b4	B4	Pm6 Output Type Select		R/W
b5	—	Reserved		R/W
b6	B6	Pm7 Output Type Select		R/W
b7	—	Reserved		R/W

m = 0 to 9, A to G, and J

ODR1 is used to select an output type for each pin of the port.
The odd bits (b1, b3, b5, and b7) in the ODR1 register are reserved.
Bits that correspond to port m on the 176-pin product but do not exist on a product with fewer pins are reserved. When writing, write 0 (CMOS output) to these bits.
The bit corresponding to a pin that does not exist is also reserved. A reserved bit is always read as 0. The write value should always be 0.

22.3.7 Pull-Up Resistor Control Register (PCR)

Address(es): PORT0.PCR 0008 C0C0h, PORT1.PCR 0008 C0C1h, PORT2.PCR 0008 C0C2h, PORT3.PCR 0008 C0C3h, PORT4.PCR 0008 C0C4h, PORT5.PCR 0008 C0C5h, PORT6.PCR 0008 C0C6h, PORT7.PCR 0008 C0C7h, PORT8.PCR 0008 C0C8h, PORT9.PCR 0008 C0C9h, PORTA.PCR 0008 C0CAh, PORTB.PCR 0008 C0CBh, PORTC.PCR 0008 C0CCh, PORTD.PCR 0008 C0CDh, PORTE.PCR 0008 C0CEh, PORTF.PCR 0008 C0CFh, PORTG.PCR 0008 C0D0h, PORTJ.PCR 0008 C0D2h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Input Pull-Up Resistor Control	0: Disables an input pull-up resistor. 1: Enables an input pull-up resistor.	R/W
b1	B1	Pm1 Input Pull-Up Resistor Control		R/W
b2	B2	Pm2 Input Pull-Up Resistor Control		R/W
b3	B3	Pm3 Input Pull-Up Resistor Control		R/W
b4	B4	Pm4 Input Pull-Up Resistor Control		R/W
b5	B5	Pm5 Input Pull-Up Resistor Control		R/W
b6	B6	Pm6 Input Pull-Up Resistor Control		R/W
b7	B7	Pm7 Input Pull-Up Resistor Control		R/W

m = 0 to 9, A to G, J

PCR is a register which enables or disables an input pull-up resistor for individual pins of the port.
While a pin is in the input state with the corresponding bit in PORTm.PCR set to 1, the pull-up resistor connected to the pin is enabled.
When a pin is set as an external bus pin (except for the WAIT pin), a general port output pin, or a peripheral module output pin, the pull-up resistor for the pin is disabled regardless of the settings of PCR.
The pull-up resistor is also disabled in the reset state.
The other bits are also reserved because they correspond to pins that do not exist is reserved. A reserved bit is always read as 0. The write value should always be 0.

22.3.8 Drive Capacity Control Register (DSCR)

Address(es): PORT0.DSCR 0008 C0E0h, PORT2.DSCR 0008 C0E2h, PORT5.DSCR 0008 C0E5h, PORT9.DSCR 0008 C0E9h, PORTA.DSCR 0008 C0EAh, PORTB.DSCR 0008 C0EBh, PORTC.DSCR 0008 C0ECh, PORTD.DSCR 0008 C0EDh, PORTE.DSCR 0008 C0EEh, PORTG.DSCR 0008 C0F0h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Drive Capacity Control	0: Normal drive output 1: High-drive output	R/W
b1	B1	Pm1 Drive Capacity Control		R/W
b2	B2	Pm2 Drive Capacity Control		R/W
b3	B3	Pm3 Drive Capacity Control		R/W
b4	B4	Pm4 Drive Capacity Control		R/W
b5	B5	Pm5 Drive Capacity Control		R/W
b6	B6	Pm6 Drive Capacity Control		R/W
b7	B7	Pm7 Drive Capacity Control		R/W

m = 0, 2, 5, 9, A to E, and G

DSCR is a register which is used to switch the drive capacity of the port.

The bit corresponding to a pin whose drive capacity has been fixed is readable and writable, but the drive capacity cannot be changed.

The bit corresponding to a pin that does not exist is reserved. A reserved bit is always read as 0. The write value should always be 0.

22.4 Handling of Unused Pins

Details on the handling of unused pins are given in Table 22.3.

Table 22.3 Handling of Unused Pins

Pin Name	Handling
EMLE	Connect this pin to VSS via a resistor (pulling down).
BSCANP	Connect this pin to VSS via a resistor (pulling down).
MD	Use this as a mode pin.
RES#	Connect this pin to VCC via a resistor (pulling up).
VCC_USB	Connect this pin to VCC
VSS_USB	Connect this pin to VSS
USB0_DP	Keep these pins open.
USB0_DM	
P35/NMI	Connect this pin to VCC via a resistor (pulling up).
EXTAL	Connect this pin to VSS via a resistor (pulling down).
XTAL	Keep this pin open.
XCIN	Connect this pin to VSS via a resistor (pulling down).
XCOUT	Keep this pin open.
Port 0 to Port 9, Port A to Port G, Port J	• If the direction setting is for input (PORTn.PDR = 0), the corresponding pin is connected to VCC (pulled up) via a resistor or to VSS (pulled down) via a resistor.*1 • If the direction setting is for output (PORTn.PDR = 1), the pin is released.*1, *2
VREFH0	Connect this pin to AVCC0.
VREFL0	Connect this pin to AVSS0.
USBA_DP USBA_DM USBA_RREF	Connect AVCC_USBA = VCC_USBA to VCC. After connecting AVSS_USBA = PVSS_USBA = VSS1_USBA = VSS2_USBA to VSS, set the module-stop state for USBA (MSTPCRB.MSTPB12 = 1) and keep USBA_DP, USBA_DM, and USBA_RREF open.

Note 1. Clear the PORTn.PMR bit, the PmnPFS.ISEL bit and the PmnPFS.ASEL bit to 0.
Note 2. In the case of release when the setting is for output, the port is an input over the period from release from the reset state to the pin becoming an output. Since the voltage on the pin is undefined while it is an input, this may lead to an increase in the current drawn.

22.5 Usage Notes

22.5.1 Note on the Port Direction Register (PDR)

Bits in the port direction register (PDR) that correspond to port m on the 176-pin product but do not exist on a product with fewer pins are reserved. When writing, write 1 (output) to these bits.
Write 1 (output) to bits that correspond to ports 54 to 56 on the 176-pin product.

23. Multi-Function Pin Controller (MPC)

23.1 Overview

The multi-function pin controller (MPC) selects and assigns input/output of peripheral functions and interrupt input signals from multiple ports. The MPC also assigns the port of external bus related signals.

Table 23.1 lists the functions assigned to each multiplexed pin. The symbols ✓ and × in the table indicate whether the pin is available or unavailable for the package. Selecting a single function for multiple pins is prohibited.

Table 23.1 Functions Assigned to Each Multiplexed Pin (1/17)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
Interrupt		NMI (input)	P35	✓	✓	✓
EXDMA controller	EXDMAC0	EDREQ0 (input)	P22	✓	✓	✓
			P55	×	✓	✓
			P80	✓	✓	×
		EDACK0 (output)	P23	✓	✓	✓
			P54	×	✓	✓
			P81	✓	✓	×
	EXDMAC1	EDREQ1 (input)	P24	✓	✓	✓
			P33	✓	✓	✓
			P82	✓	✓	×
		EDACK1 (output)	P25	✓	✓	✓
			P56	×	✓	×
			P83	✓	✓	×
			PJ3	✓	✓	✓
Interrupt	IRQ0	IRQ0-DS (input)	P30	✓	✓	✓
		IRQ0 (input)	P10	✓	×	×
			PD0	✓	✓	✓
	IRQ1	IRQ1-DS (input)	P31	✓	✓	✓
		IRQ1 (input)	P11	✓	×	×
			PD1	✓	✓	✓
	IRQ2	IRQ2-DS (input)	P32	✓	✓	✓
		IRQ2 (input)	P12	✓	✓	✓
			PD2	✓	✓	✓
	IRQ3	IRQ3-DS (input)	P33	✓	✓	✓
		IRQ3 (input)	P13	✓	✓	✓
			PD3	✓	✓	✓
	IRQ4	IRQ4-DS (input)	PB1	✓	✓	✓
		IRQ4 (input)	P14	✓	✓	✓
			P34	✓	✓	✓
			PD4	✓	✓	✓
			PF5	✓	✓	×
	IRQ5	IRQ5-DS (input)	PA4	✓	✓	✓
		IRQ5 (input)	P15	✓	✓	✓
			PD5	✓	✓	✓
			PE5	✓	✓	✓
	IRQ6	IRQ6-DS (input)	PA3	✓	✓	✓
		IRQ6 (input)	P16	✓	✓	✓
			PD6	✓	✓	✓
			PE6	✓	✓	✓
	IRQ7	IRQ7-DS (input)	PE2	✓	✓	✓
		IRQ7 (input)	P17	✓	✓	✓
			PD7	✓	✓	✓
			PE7	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (2/17)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
Interrupt	IRQ8	IRQ8-DS (input)	P40	✓	✓	✓
		IRQ8 (input)	P00	✓	✓	×
			P20	✓	✓	✓
	IRQ9	IRQ9-DS (input)	P41	✓	✓	✓
		IRQ9 (input)	P01	✓	✓	×
			P21	✓	✓	✓
	IRQ10	IRQ10-DS (input)	P42	✓	✓	✓
		IRQ10 (input)	P02	✓	✓	×
			P55	×	✓	✓
	IRQ11	IRQ11-DS (input)	P43	✓	✓	✓
		IRQ11 (input)	P03	✓	✓	×
			PA1	✓	✓	✓
	IRQ12	IRQ12-DS (input)	P44	✓	✓	✓
		IRQ12 (input)	PB0	✓	✓	✓
			PC1	✓	✓	✓
	IRQ13	IRQ13-DS (input)	P45	✓	✓	✓
		IRQ13 (input)	P05	✓	✓	✓
			PC6	✓	✓	✓
	IRQ14	IRQ14-DS (input)	P46	✓	✓	✓
		IRQ14 (input)	PC0	✓	✓	✓
			PC7	✓	✓	✓
	IRQ15	IRQ15-DS (input)	P47	✓	✓	✓
		IRQ15 (input)	P07	✓	✓	✓
			P67	✓	✓	×
Multi-function timer unit 3	MTU0	MTIOC0A (input/output)	P34	✓	✓	✓
			PB3	✓	✓	✓
		MTIOC0B (input/output)	P13	✓	✓	✓
			P15	✓	✓	✓
			PA1	✓	✓	✓
		MTIOC0C (input/output)	P32	✓	✓	✓
			PB1	✓	✓	✓
		MTIOC0D (input/output)	P33	✓	✓	✓
			PA3	✓	✓	✓
	MTU1	MTIOC1A (input/output)	P20	✓	✓	✓
			PE4	✓	✓	✓
		MTIOC1B (input/output)	P21	✓	✓	✓
			PB5	✓	✓	✓
	MTU2	MTIOC2A (input/output)	P26	✓	✓	✓
			PB5	✓	✓	✓
		MTIOC2B (input/output)	P27	✓	✓	✓
			PE5	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (3/17)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
Multi-function timer unit 3	MTU3	MTIOC3A (input/output)	P14	✓	✓	✓
			P17	✓	✓	✓
			PC1	✓	✓	✓
			PC7	✓	✓	✓
		MTIOC3B (input/output)	P17	✓	✓	✓
			P22	✓	✓	✓
			P80	✓	✓	×
			PB7	✓	✓	✓
			PC5	✓	✓	✓
			PE1	✓	✓	✓
		MTIOC3C (input/output)	P16	✓	✓	✓
			P56	×	✓	×
			PC0	✓	✓	✓
			PC6	✓	✓	✓
			PJ3	✓	✓	✓
		MTIOC3D (input/output)	P16	✓	✓	✓
			P23	✓	✓	✓
			P81	✓	✓	×
			PB6	✓	✓	✓
			PC4	✓	✓	✓
			PE0	✓	✓	✓
	MTU4	MTIOC4A (input/output)	P21	✓	✓	✓
			P24	✓	✓	✓
			P82	✓	✓	×
			PA0	✓	✓	✓
			PB3	✓	✓	✓
			PE2	✓	✓	✓
		MTIOC4B (input/output)	P17	✓	✓	✓
			P30	✓	✓	✓
			P54	×	✓	✓
			PC2	✓	✓	✓
			PD1	✓	✓	✓
			PE3	✓	✓	✓
		MTIOC4C (input/output)	P25	✓	✓	✓
			P83	✓	✓	×
			P87	✓	✓	×
			PB1	✓	✓	✓
			PE1	✓	✓	✓
			PE5	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (4/17)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
Multi-function timer unit 3	MTU4	MTIOC4D (input/output)	P31	✓	✓	✓
			P55	×	✓	✓
			P86	✓	✓	×
			PC3	✓	✓	✓
			PD2	✓	✓	✓
			PE4	✓	✓	✓
	MTU5	MTIC5U (input)	P12	✓	×	×
			PA4	✓	✓	✓
			PD7	✓	✓	✓
		MTIC5V (input)	P11	✓	×	×
			PA6	✓	✓	✓
			PD6	✓	✓	✓
		MTIC5W (input)	P10	✓	×	×
			PB0	✓	✓	✓
			PD5	✓	✓	✓
	MTU6	MTIOC6A (input/output)	PE7	✓	✓	✓
		MTIOC6B (input/output)	PA5	✓	✓	✓
		MTIOC6C (input/output)	PE6	✓	✓	✓
		MTIOC6D (input/output)	PA0	✓	✓	✓
	MTU7	MTIOC7A (input/output)	PA2	✓	✓	✓
		MTIOC7B (input/output)	PA1	✓	✓	✓
		MTIOC7C (input/output)	P67	✓	✓	×
		MTIOC7D (input/output)	P66	✓	✓	×
	MTU8	MTIOC8A (input/output)	PD6	✓	✓	✓
		MTIOC8B (input/output)	PD4	✓	✓	✓
		MTIOC8C (input/output)	PD5	✓	✓	✓
		MTIOC8D (input/output)	PD3	✓	✓	✓
	MTU	MTCLKA (input)	P14	✓	✓	✓
			P24	✓	✓	✓
			PA4	✓	✓	✓
			PC6	✓	✓	✓
		MTCLKB (input)	P15	✓	✓	✓
			P25	✓	✓	✓
			PA6	✓	✓	✓
			PC7	✓	✓	✓
		MTCLKC (input)	P22	✓	✓	✓
			PA1	✓	✓	✓
			PC4	✓	✓	✓
		MTCLKD (input)	P23	✓	✓	✓
			PA3	✓	✓	✓
			PC5	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (5/17)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
Port output enable 3	POE0	POE0# (input)	P32	✓	✓	✓
			P93	✓	✓	×
			PC4	✓	✓	✓
			PD1	✓	✓	✓
			PD7	✓	✓	✓
	POE4	POE4# (input)	P33	✓	✓	✓
			P92	✓	✓	×
			PB5	✓	✓	✓
			PD0	✓	✓	✓
			PD6	✓	✓	✓
	POE8	POE8# (input)	P17	✓	✓	✓
			P30	✓	✓	✓
			PD3	✓	✓	✓
			PE3	✓	✓	✓
			PJ5	✓	✓	×
	POE10	POE10# (input)	P32	✓	✓	✓
			P34	✓	✓	✓
			PA6	✓	✓	✓
			PD5	✓	✓	✓
	POE11	POE11# (input)	P33	✓	✓	✓
			PB3	✓	✓	✓
			PD4	✓	✓	✓
General PWM timer	GPT0	GTIOC0A (input/output)	P23	✓	✓	✓
			P83	✓	✓	×
			PA5	✓	✓	✓
			PD3	✓	✓	✓
			PE5	✓	✓	✓
		GTIOC0B (input/output)	P17	✓	✓	✓
			P81	✓	✓	×
			PA0	✓	✓	✓
			PD2	✓	✓	✓
			PE2	✓	✓	✓
	GPT1	GTIOC1A (input/output)	P22	✓	✓	✓
			PC5	✓	✓	✓
			PA2	✓	✓	✓
			PE4	✓	✓	✓
			PD1	✓	✓	✓
		GTIOC1B (input/output)	P67	✓	✓	×
			P87	✓	✓	×
			PC3	✓	✓	✓
			PD0	✓	✓	✓
			PE1	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (6/17)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
General PWM timer	GPT2	GTIOC2A (input/output)	P21	✓	✓	✓
			P82	✓	✓	×
			PA1	✓	✓	✓
			PE3	✓	✓	✓
		GTIOC2B (input/output)	P66	✓	✓	×
			P86	✓	✓	×
			PC2	✓	✓	✓
			PE0	✓	✓	✓
	GPT3	GTIOC3A (input/output)	PC7	✓	✓	✓
			PE7	✓	✓	✓
		GTIOC3B (input/output)	PC6	✓	✓	✓
			PE6	✓	✓	✓
	GPT	GTETRG (input)	P15	✓	✓	✓
			PC4	✓	✓	✓
			PA6	✓	✓	✓
16-bit timer pulse unit	TPU0	TIOCA0 (input/output)	P86	✓	✓	×
			PA0	✓	✓	✓
		TIOCB0 (input/output)	P17	✓	✓	✓
			PA1	✓	✓	✓
		TIOCC0 (input/output)	P32	✓	✓	✓
		TIOCD0 (input/output)	P33	✓	✓	✓
			PA3	✓	✓	✓
	TPU1	TIOCA1 (input/output)	P56	×	✓	×
			PA4	✓	✓	✓
		TIOCB1 (input/output)	P16	✓	✓	✓
			PA5	✓	✓	✓
	TPU2	TIOCA2 (input/output)	P87	✓	✓	×
			PA6	✓	✓	✓
		TIOCB2 (input/output)	P15	✓	✓	✓
			PA7	✓	✓	✓
	TPU3	TIOCA3 (input/output)	P21	✓	✓	✓
			PB0	✓	✓	✓
		TIOCB3 (input/output)	P20	✓	✓	✓
			PB1	✓	✓	✓
		TIOCC3 (input/output)	P22	✓	✓	✓
			PB2	✓	✓	✓
		TIOCD3 (input/output)	P23	✓	✓	✓
			PB3	✓	✓	✓
	TPU4	TIOCA4 (input/output)	P25	✓	✓	✓
			PB4	✓	✓	✓
		TIOCB4 (input/output)	P24	✓	✓	✓
			PB5	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (7/17)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
16-bit timer pulse unit	TPU5	TIOCA5 (input/output)	P13	✓	✓	✓
			PB6	✓	✓	✓
		TIOCB5 (input/output)	P14	✓	✓	✓
			PB7	✓	✓	✓
	TPU (unit 0)	TCLKA (input)	P14	✓	✓	✓
			PC2	✓	✓	✓
		TCLKB (input)	P15	✓	✓	✓
			PA3	✓	✓	✓
			PC3	✓	✓	✓
		TCLKC (input)	P16	✓	✓	✓
			PB2	✓	✓	✓
			PC0	✓	✓	✓
		TCLKD (input)	P17	✓	✓	✓
			PB3	✓	✓	✓
			PC1	✓	✓	✓
Programmable pulse generator	PPG0	PO0 (output)	P20	✓	✓	✓
		PO1 (output)	P21	✓	✓	✓
		PO2 (output)	P22	✓	✓	✓
		PO3 (output)	P23	✓	✓	✓
		PO4 (output)	P24	✓	✓	✓
		PO5 (output)	P25	✓	✓	✓
		PO6 (output)	P26	✓	✓	✓
		PO7 (output)	P27	✓	✓	✓
		PO8 (output)	P30	✓	✓	✓
		PO9 (output)	P31	✓	✓	✓
		PO10 (output)	P32	✓	✓	✓
		PO11 (output)	P33	✓	✓	✓
		PO12 (output)	P34	✓	✓	✓
		PO13 (output)	P13	✓	✓	✓
			P15	✓	✓	✓
		PO14 (output)	P16	✓	✓	✓
		PO15 (output)	P14	✓	✓	✓
			P17	✓	✓	✓
	PPG1	PO16 (output)	P73	✓	✓	×
			PA0	✓	✓	✓
		PO17 (output)	PA1	✓	✓	✓
			PC0	✓	✓	✓
		PO18 (output)	PA2	✓	✓	✓
			PC1	✓	✓	✓
			PE1	✓	✓	✓
		PO19 (output)	P74	✓	✓	×
			PA3	✓	✓	✓
		PO20 (output)	P75	✓	✓	×
			PA4	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (8/17)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
Programmable pulse generator	PPG1	PO21 (output)	PA5	✓	✓	✓
			PC2	✓	✓	✓
		PO22 (output)	P76	✓	✓	×
			PA6	✓	✓	✓
		PO23 (output)	P77	✓	✓	×
			PA7	✓	✓	✓
			PE2	✓	✓	✓
		PO24 (output)	PB0	✓	✓	✓
			PC3	✓	✓	✓
		PO25 (output)	PB1	✓	✓	✓
			PC4	✓	✓	✓
		PO26 (output)	P80	✓	✓	×
			PB2	✓	✓	✓
			PE3	✓	✓	✓
		PO27 (output)	P81	✓	✓	×
			PB3	✓	✓	✓
		PO28 (output)	P82	✓	✓	×
			PB4	✓	✓	✓
			PE4	✓	✓	✓
		PO29 (output)	PB5	✓	✓	✓
			PC5	✓	✓	✓
		PO30 (output)	PB6	✓	✓	✓
			PC6	✓	✓	✓
		PO31 (output)	PB7	✓	✓	✓
			PC7	✓	✓	✓
8-bit timer	TMR0	TMO0 (output)	P22	✓	✓	✓
			PB3	✓	✓	✓
		TMCI0 (input)	P01	✓	✓	×
			P21	✓	✓	✓
			PB1	✓	✓	✓
		TMRI0 (input)	P00	✓	✓	×
			P20	✓	✓	✓
			PA4	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (9/17)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	145-pin 144-pin	100-pin
8-bit timer	TMR1	TMO1 (output)	P17	✓	✓	✓
			P26	✓	✓	✓
		TMCI1 (input)	P02	✓	✓	×
			P12	✓	✓	✓
			P54	×	✓	✓
			PC4	✓	✓	✓
		TMRI1 (input)	P24	✓	✓	✓
			PB5	✓	✓	✓
	TMR2	TMO2 (output)	P16	✓	✓	✓
			PC7	✓	✓	✓
		TMCI2 (input)	P15	✓	✓	✓
			P31	✓	✓	✓
			PC6	✓	✓	✓
		TMRI2 (input)	P14	✓	✓	✓
			PC5	✓	✓	✓
	TMR3	TMO3 (output)	P13	✓	✓	✓
			P32	✓	✓	✓
			P55	×	✓	✓
		TMCI3 (input)	P11	✓	×	×
			P27	✓	✓	✓
			P34	✓	✓	✓
			PA6	✓	✓	✓
		TMRI3 (input)	P10	✓	×	×
			P30	✓	✓	✓
			P33	✓	✓	✓
Compare match timer W	CMTW0	TOC0 (output)	PC7	✓	✓	✓
		TIC0 (input)	PC6	✓	✓	✓
	CMTW1	TOC1 (output)	PE7	✓	✓	✓
		TIC1 (input)	PE6	✓	✓	✓
	CMTW2	TOC2 (output)	PD3	✓	✓	✓
		TIC2 (input)	PD2	✓	✓	✓
	CMTW3	TOC3 (output)	PE3	✓	✓	✓
		TIC3 (input)	PE2	✓	✓	✓
Ethernet controller	RMII0	REF50CK0 (input)	P76	✓	✓	×
			PB2	✓	✓	✓
			PE5	✓	✓	✓
		RMII0_CRS_DV (input)	P83	✓	✓	×
			PB7	✓	✓	✓
		RMII0_TXD0 (output)	P81	✓	✓	×
			PB5	✓	✓	✓
		RMII0_TXD1 (output)	P82	✓	✓	×
			PB6	✓	✓	✓
		RMII0_RXD0 (input)	P75	✓	✓	×
			PB1	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (10/17)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
Ethernet controller	RMII0	RMII0_RXD1 (input)	P74	✓	✓	×
			PB0	✓	✓	✓
		RMII0_TXD_EN (output)	P80	✓	✓	×
			PA0	✓	✓	✓
			PB4	✓	✓	✓
		RMII0_RX_ER (input)	P77	✓	✓	×
			PB3	✓	✓	✓
	MII0	ET0_CRS (input)	P83	✓	✓	×
			PB7	✓	✓	✓
		ET0_RX_DV (input)	PC2	✓	✓	✓
		ET0_EXOUT (output)	P55	×	✓	✓
			PA6	✓	✓	✓
			PJ3	✓	✓	✓
		ET0_LINKSTA (input)	P34	✓	✓	✓
			P54	×	✓	✓
			PA5	✓	✓	✓
		ET0_ETXD0 (output)	P81	✓	✓	×
			PB5	✓	✓	✓
		ET0_ETXD1 (output)	P82	✓	✓	×
			PB6	✓	✓	✓
		ET0_ETXD2 (output)	PC5	✓	✓	✓
		ET0_ETXD3 (output)	PC6	✓	✓	✓
		ET0_ERXD0 (input)	P75	✓	✓	×
			PB1	✓	✓	✓
		ET0_ERXD1 (input)	P74	✓	✓	×
			PB0	✓	✓	✓
		ET0_ERXD2 (input)	PC1	✓	✓	✓
			PE4	✓	✓	✓
		ET0_ERXD3 (input)	PC0	✓	✓	✓
			PE3	✓	✓	✓
		ET0_TX_EN (output)	P80	✓	✓	×
			PA0	✓	✓	✓
			PB4	✓	✓	✓
		ET0_TX_ER (output)	PC3	✓	✓	✓
		ET0_RX_ER (input)	P77	✓	✓	×
			PB3	✓	✓	✓
		ET0_TX_CLK (input)	PC4	✓	✓	✓
		ET0_RX_CLK (input)	P76	✓	✓	×
			PB2	✓	✓	✓
			PE5	✓	✓	✓
		ET0_COL (input)	PC7	✓	✓	✓
		ET0_WOL (output)	P73	✓	✓	×
			PA1	✓	✓	✓
			PA7	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (11/17)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
Ethernet controller	MII0	ET0_MDC (output)	P72	✓	✓	×
			PA4	✓	✓	✓
		ET0_MDIO (input/output)	P71	✓	✓	×
			PA3	✓	✓	✓
	RMII1	REF50CK1 (input)	PG0	✓	×	×
		RMII1_CRS_DV (input)	P92	✓	×	×
		RMII1_TXD0 (output)	PG3	✓	×	×
		RMII1_TXD1 (output)	PG4	✓	×	×
		RMII1_RXD0 (input)	P94	✓	×	×
		RMII1_RXD1 (input)	P95	✓	×	×
		RMII1_TXD_EN (output)	P60	✓	×	×
		RMII1_RX_ER (input)	PG1	✓	×	×
	MII1	ET1_CRS (input)	P92	✓	×	×
		ET1_RX_DV (input)	P90	✓	×	×
		ET1_EXOUT (output)	P26	✓	×	×
		ET1_LINKSTA (input)	P93	✓	×	×
		ET1_ETXD0 (output)	PG3	✓	×	×
		ET1_ETXD1 (output)	PG4	✓	×	×
		ET1_ETXD2 (output)	PG5	✓	×	×
		ET1_ETXD3 (output)	PG6	✓	×	×
		ET1_ERXD0 (input)	P94	✓	×	×
		ET1_ERXD1 (input)	P95	✓	×	×
		ET1_ERXD2 (input)	P96	✓	×	×
		ET1_ERXD3 (input)	P97	✓	×	×
		ET1_TX_EN (output)	P60	✓	×	×
		ET1_TX_ER (output)	PG7	✓	×	×
		ET1_RX_ER (input)	PG1	✓	×	×
		ET1_TX_CLK (input)	PG2	✓	×	×
		ET1_RX_CLK (input)	PG0	✓	×	×
		ET1_COL (input)	P91	✓	×	×
		ET1_WOL (output)	P27	✓	×	×
		ET1_MDC (output)	P31	✓	×	×
		ET1_MDIO (input/output)	P30	✓	×	×
Serial communications interface	SCI0	RXD0 (input)/SMISO0 (input/output)/ SSCL0 (input/output)	P21	✓	✓	✓
			P33	✓	✓	✓
		TXD0 (output)/SMOSI0 (input/output)/ SSDA0 (input/output)	P20	✓	✓	✓
			P32	✓	✓	✓
		SCK0 (input/output)	P22	✓	✓	✓
			P34	✓	✓	✓
		CTS0# (input)/RTS0# (output)/ SS0# (input)	P23	✓	✓	✓
			PJ3	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (12/17)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
Serial communications interface	SCI1	RXD1 (input)/SMISO1 (input/output)/ SSCL1 (input/output)	P15	✓	✓	✓
			P30	✓	✓	✓
			PF2	✓	×	×
		TXD1 (output)/SMOSI1 (input/output)/ SSDA1 (input/output)	P16	✓	✓	✓
			P26	✓	✓	✓
			PF0	✓	×	×
		SCK1 (input/output)	P17	✓	✓	✓
			P27	✓	✓	✓
			PF1	✓	×	×
		CTS1# (input)/RTS1# (output)/ SS1# (input)	P14	✓	✓	✓
			P31	✓	✓	✓
	SCI2	RXD2 (input)/SMISO2 (input/output)/ SSCL2 (input/output)	P12	✓	✓	✓
			P52	✓	✓	✓
		TXD2 (output)/SMOSI2 (input/output)/ SSDA2 (input/output)	P13	✓	✓	✓
			P50	✓	✓	✓
		SCK2 (input/output)	P11	✓	×	×
			P51	✓	✓	✓
		CTS2# (input)/RTS2# (output)/ SS2# (input)	P54	×	✓	✓
			PJ5	✓	✓	×
	SCI3	RXD3 (input)/SMISO3 (input/output)/ SSCL3 (input/output)	P16	✓	✓	✓
			P25	✓	✓	✓
		TXD3 (output)/SMOSI3 (input/output)/ SSDA3 (input/output)	P17	✓	✓	✓
			P23	✓	✓	✓
		SCK3 (input/output)	P15	✓	✓	✓
			P24	✓	✓	✓
		CTS3# (input)/RTS3# (output)/ SS3# (input)	P26	✓	✓	✓
	SCI4	RXD4 (input)/SMISO4 (input/output)/ SSCL4 (input/output)	PB0	✓	✓	×
		TXD4 (output)/SMOSI4 (input/output)/ SSDA4 (input/output)	PB1	✓	✓	×
		SCK4 (input/output)	PB3	✓	✓	×
		CTS4# (input)/RTS4# (output)/ SS4# (input)	PB2	✓	✓	×
	SCI5	RXD5 (input)/SMISO5 (input/output)/ SSCL5 (input/output)	PA2	✓	✓	✓
			PA3	✓	✓	✓
			PC2	✓	✓	✓
		TXD5 (output)/SMOSI5 (input/output)/ SSDA5 (input/output)	PA4	✓	✓	✓
			PC3	✓	✓	✓
		SCK5 (input/output)	PA1	✓	✓	✓
			PC1	✓	✓	✓
			PC4	✓	✓	✓
		CTS5# (input)/RTS5# (output)/ SS5# (input)	PA6	✓	✓	✓
			PC0	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (13/17)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
Serial communications interface	SCI6	RXD6 (input)/SMISO6 (input/output)/ SSCL6 (input/output)	P01	✓	✓	×
			P33	✓	✓	✓
			PB0	✓	✓	✓
		TXD6 (output)/SMOSI6 (input/output)/ SSDA6 (input/output)	P00	✓	✓	×
			P32	✓	✓	✓
			PB1	✓	✓	✓
		SCK6 (input/output)	P02	✓	✓	×
			P34	✓	✓	✓
			PB3	✓	✓	✓
		CTS6# (input)/RTS6# (output)/ SS6# (input)	PB2	✓	✓	✓
			PJ3	✓	✓	✓
	SCI7	RXD7 (input)/SMISO7 (input/output)/ SSCL7 (input/output)	P92	✓	✓	×
		TXD7 (output)/SMOSI7 (input/output)/ SSDA7 (input/output)	P90	✓	✓	×
		SCK7 (input/output)	P91	✓	✓	×
		CTS7# (input)/RTS7# (output)/ SS7# (input)	P93	✓	✓	×
	SCIF8	RXD8 (input)	PC6	✓	✓	✓
		TXD8 (output)	PC7	✓	✓	✓
		SCK8 (input/output)/RTS8# (output)	PC5	✓	✓	✓
		CTS8# (input)	PC4	✓	✓	✓
	SCIF9	RXD9 (input)	PB6	✓	✓	✓
		TXD9 (output)	PB7	✓	✓	✓
		SCK9 (input/output)/RTS9# (output)	PB5	✓	✓	✓
		CTS9# (input)	PB4	✓	✓	✓
	SCIF10	RXD10 (input)	P81	✓	✓	×
			P86	✓	✓	×
		TXD10 (output)	P82	✓	✓	×
			P87	✓	✓	×
		SCK10 (input/output)	P80	✓	✓	×
			P83	✓	✓	×
		RTS10# (output)	P80	✓	✓	×
		CTS10# (input)	P83	✓	✓	×
	SCIF11	RXD11 (input)	P76	✓	✓	×
		TXD11 (output)	P77	✓	✓	×
		SCK11 (input/output)/RTS11# (output)	P75	✓	✓	×
		CTS11# (input)	P74	✓	✓	×
	SCI12	RXD12 (input)/SMISO12 (input/output)/ SSCL12 (input/output)/RXDX12 (input)	PE2	✓	✓	✓
		TXD12 (output)/SMOSI12 (input/output)/ SSDA12 (input/output)/TXDX12 (output)/ SIOX12 (input/output)	PE1	✓	✓	✓
		SCK12 (input/output)	PE0	✓	✓	✓
		CTS12# (input)/RTS12# (output)/ SS12# (input)	PE3	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (14/17)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
I²C bus interface	RIIC0	SCL0[FM+] (input/output)	P12	✓	✓	✓
		SDA0[FM+] (input/output)	P13	✓	✓	✓
	RIIC2	SCL2-DS (input/output)	P16	✓	✓	✓
		SDA2-DS (input/output)	P17	✓	✓	✓
USB 2.0FS host/ function module	USB0	USB0_VBUS (input)	P16	✓	✓	✓
		USB0_EXICEN (output)	P21	✓	✓	✓
		USB0_VBUSEN (output)	P16	✓	✓	✓
			P24	✓	✓	✓
			P32	✓	✓	✓
		USB0_OVRCURA (input)	P14	✓	✓	✓
		USB0_OVRCURB (input)	P16	✓	✓	✓
			P22	✓	✓	✓
		USB0_ID (input)	P20	✓	✓	✓
USB 2.0FS host/ function module	USBA	USBA_VBUS (input)	P11	✓	×	×
		USBA_EXICEN (output)	P21	✓	×	×
		USBA_VBUSEN (output)	P11	✓	×	×
			P15	✓	×	×
		USBA_OVRCURA (input)	P10	✓	×	×
		USBA_OVRCURB (input)	P22	✓	×	×
		USBA_ID (input)	P20	✓	×	×
CAN module	CAN0	CRX0 (input)	P33	✓	✓	✓
			PD2	✓	✓	✓
		CTX0 (output)	P32	✓	✓	✓
			PD1	✓	✓	✓
	CAN1	CRX1-DS (input)	P15	✓	✓	✓
		CRX1 (input)	P55	×	✓	✓
		CTX1 (output)	P14	✓	✓	✓
			P54	×	✓	✓
	CAN2	CRX2 (input)	P67	✓	✓	×
		CTX2 (output)	P66	✓	✓	×
Serial peripheral interface	RSPI0	RSPCKA (input/output)	PA5	✓	✓	✓
			PC5	✓	✓	✓
		MOSIA (input/output)	PA6	✓	✓	✓
			PC6	✓	✓	✓
		MISOA (input/output)	PA7	✓	✓	✓
			PC7	✓	✓	✓
		SSLA0 (input/output)	PA4	✓	✓	✓
			PC4	✓	✓	✓
		SSLA1 (output)	PA0	✓	✓	✓
			PC0	✓	✓	✓
		SSLA2 (output)	PA1	✓	✓	✓
			PC1	✓	✓	✓
		SSLA3 (output)	PA2	✓	✓	✓
			PC2	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (15/17)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
Realtime clock		RTCOUT (output)	P16	✓	✓	✓
			P32	✓	✓	✓
		RTCIC0 (input)*1	P30	✓	✓	✓
		RTCIC1 (input)*1	P31	✓	✓	✓
		RTCIC2 (input)*1	P32	✓	✓	✓
12-bit A/D converter		AN000 (input)*1	P40	✓	✓	✓
		AN001 (input)*1	P41	✓	✓	✓
		AN002 (input)*1	P42	✓	✓	✓
		AN003 (input)*1	P43	✓	✓	✓
		AN004 (input)*1	P44	✓	✓	✓
		AN005 (input)*1	P45	✓	✓	✓
		AN006 (input)*1	P46	✓	✓	✓
		AN007 (input)*1	P47	✓	✓	✓
		ADTRG0# (input)	P07	✓	✓	✓
			P16	✓	✓	✓
			P25	✓	✓	✓
		AN100 (input)*1	PE2	✓	✓	✓
		AN101 (input)*1	PE3	✓	✓	✓
		AN102 (input)*1	PE4	✓	✓	✓
		AN103 (input)*1	PE5	✓	✓	✓
		AN104 (input)*1	PE6	✓	✓	✓
		AN105 (input)*1	PE7	✓	✓	✓
		AN106 (input)*1	PD6	✓	✓	✓
		AN107 (input)*1	PD7	✓	✓	✓
		AN108 (input)*1	PD0	✓	✓	✓
		AN109 (input)*1	PD1	✓	✓	✓
		AN110 (input)*1	PD2	✓	✓	✓
		AN111 (input)*1	PD3	✓	✓	✓
		AN112 (input)*1	PD4	✓	✓	✓
		AN113 (input)*1	PD5	✓	✓	✓
		AN114 (input)*1	P90	✓	✓	×
		AN115 (input)*1	P91	✓	✓	×
		AN116 (input)*1	P92	✓	✓	×
		AN117 (input)*1	P93	✓	✓	×
		AN118 (input)*1	P00	✓	✓	×
		AN119 (input)*1	P01	✓	✓	×
		AN120 (input)*1	P02	✓	✓	×
		ANEX0 (output)*1	PE0	✓	✓	✓
		ANEX1 (input)*1	PE1	✓	✓	✓
		ADTRG1# (input)	P13	✓	✓	✓
			P17	✓	✓	✓
12-bit D/A converter		DA0 (output)*1	P03	✓	✓	×
		DA1 (output)*1	P05	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (16/17)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
Parallel data capture unit		PIXCLK (input)	P24	✓	✓	×
		VSYNC (input)	P32	✓	✓	×
		HSYNC (input)	P25	✓	✓	×
		PIXD0 (input)	P15	✓	✓	×
		PIXD1 (input)	P86	✓	✓	×
		PIXD2 (input)	P87	✓	✓	×
		PIXD3 (input)	P17	✓	✓	×
		PIXD4 (input)	P20	✓	✓	×
		PIXD5 (input)	P21	✓	✓	×
		PIXD6 (input)	P22	✓	✓	×
		PIXD7 (input)	P23	✓	✓	×
		PCKO (output)	P33	✓	✓	×
Serial sound interface	SSI0	SSISCK0 (input/output)	P23	✓	✓	✓
		SSIWS0 (input/output)	P21	✓	✓	✓
		SSIRXD0 (output)	P20	✓	✓	✓
		SSITXD0 (output)	P17	✓	✓	✓
	SSI1	SSISCK1 (input/output)	P24	✓	✓	✓
		SSIWS1 (input/output)	P15	✓	✓	✓
		SSIDATA1 (input/output)	P25	✓	✓	✓
	SSI	AUDIO_CLK (input)	P22	✓	✓	✓
MMC host interface		MMC_RES# (output)	P75	✓	✓	×
			PE7	✓	✓	✓
		MMC_CLK (output)	P77	✓	✓	×
			PD5	✓	✓	✓
		MMC_CD (input)	PC2	✓	✓	×
			PE6	✓	✓	✓
		MMC_CMD (input/output)	P76	✓	✓	×
			PD4	✓	✓	✓
		MMC_D0 (input/output)	PC3	✓	✓	×
			PD6	✓	✓	✓
		MMC_D1 (input/output)	PC4	✓	✓	×
			PD7	✓	✓	✓
		MMC_D2 (input/output)	P80	✓	✓	×
			PD2	✓	✓	✓
		MMC_D3 (input/output)	P81	✓	✓	×
			PD3	✓	✓	✓
		MMC_D4 (input/output)	P82	✓	✓	×
			PE0	✓	✓	✓
		MMC_D5 (input/output)	PC5	✓	✓	×
			PE1	✓	✓	✓
		MMC_D6 (input/output)	PC6	✓	✓	×
			PE2	✓	✓	✓
		MMC_D7 (input/output)	PC7	✓	✓	×
			PE3	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (17/17)

Module/Function	Channel	Pin Functions	Allocation Port	Package 177-pin 176-pin	Package 145-pin 144-pin	Package 100-pin
SD host interface		SDHI_CLK (output)	P77	✓	✓	×
			PD5	✓	✓	✓
		SDHI_CMD (input/output)	P76	✓	✓	×
			PD4	✓	✓	✓
		SDHI_CD (input)	P81	✓	✓	×
			PE6	✓	✓	✓
		SDHI_WP (input)	P80	✓	✓	×
			PE7	✓	✓	✓
		SDHI_D0 (input/output)	PC3	✓	✓	×
			PD6	✓	✓	✓
		SDHI_D1 (input/output)	PC4	✓	✓	×
			PD7	✓	✓	✓
		SDHI_D2 (input/output)	P75	✓	✓	×
			PD2	✓	✓	✓
		SDHI_D3 (input/output)	PC2	✓	✓	×
			PD3	✓	✓	✓
Clock frequency accuracy measurement circuit		CACREF (input)	PC7	✓	✓	✓
			PA0	✓	✓	✓
Quad serial peripheral interface		QSPCLK (input/output)	P77	✓	✓	×
			PD5	✓	✓	✓
		QSSL (input/output)	P76	✓	✓	×
			PD4	✓	✓	✓
		QMO/QIO0 (input/output)	PC3	✓	✓	×
			PD6	✓	✓	✓
		QMI/QIO1 (input/output)	PC4	✓	✓	×
			PD7	✓	✓	✓
		QIO2 (input/output)	P80	✓	✓	×
			PD2	✓	✓	✓
		QIO3 (input/output)	P81	✓	✓	×
			PD3	✓	✓	✓

Note 1. To use this pin function, set the corresponding pin as general input (set the PORTm.PDR.Bn and PORTm.PMR.Bn bits to 0).

23.2 Register Descriptions

The registers and bits of unsupported pins, depending on the package, are reserved. The write value to the reserved bits is the value after a reset.

23.2.1 Write-Protect Register (PWPR)

Address(es): 0008 C11Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	B0WI	PFSWE	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PFSWE	PFS Register Write Enable	0: Writing to the PFS register is disabled 1: Writing to the PFS register is enabled	R/W
b7	B0WI	PFSWE Bit Write Disable	0: Writing to the PFSWE bit is enabled 1: Writing to the PFSWE bit is disabled	R/W

PFSWE Bit (PFS Register Write Enable)

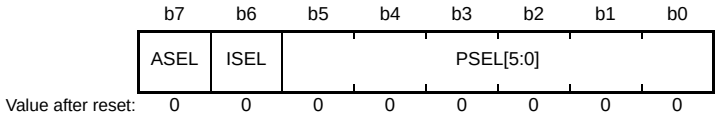
Writing to PmnPFS register is enabled only when the PFSWE bit is set to 1.
To set the PFSWE bit to 1, write 1 to the PFSWE bit after writing 0 to the B0WI bit.

B0WI Bit (PFSWE Bit Write Disable)

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

23.2.2 P0n Pin Function Control Register (P0nPFS) (n = 0 to 3, 5, 7)

Address(es): P00PFS 0008 C140h, P01PFS 0008 C141h, P02PFS 0008 C142h, P03PFS 0008 C143h, P05PFS 0008 C145h, P07PFS 0008 C147h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin. 1: Used as analog pin. P03: DA0 (177/176/145/144 pin) P05: DA1 (177/176/145/144/100 pin)	R/W

The port mn pin function control register (PmnPFS) selects the pin function. Bits PSEL[5:0] select the peripheral function which is assigned to bits.
The ISEL bit is set when a pin is used as an IRQ input pin. This setting can be used with the combination of the peripheral function, though IRQn (external pin interrupt) of the same number should not be enabled by two or more pins.
The ASEL bit is set when a pin is used as an analog pin. When the pin is set as an analog pin by the ASEL bit, select the general I/O port by the port mode register (PORTm.PMR) and specify input by the port direction register (PORTm.PDR). The pin state cannot be read at this point, since the PmnPFS register is protected by the write-protect register (PWPR). Modify the register after releasing the protection.
The ISEL bit to which IRQn is not specified is reserved. The ASEL bit to which analog input/output is not specified is reserved.

Table 23.2 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LFQFP

	Pin			
PSEL[5:0] Settings	**P00**	**P01**	**P02**	**P07**
000000b (initial value)	Hi-Z			
000101b	TMRI0	TMCI0	TMCI1	—
001001b	—	—	—	ADTRG0#
001010b	TXD6 SMOSI6 SSDA6	RXD6 SMISO6 SSCL6	SCK6	—

—: Do not specify this value.

Table 23.3 Register Settings for Input/Output Pin Function in 100-Pin TFLGA, 100-Pin LFQFP

	Pin
PSEL[5:0] Settings	**P07**
000000b (initial value)	Hi-Z
000101b	—
001001b	ADTRG0#
001010b	—

—: Do not specify this value.

23.2.3 P1n Pin Function Control Registers (P1nPFS) (n = 0 to 7)

Address(es): P10PFS 0008 C148h, P11PFS 0008 C149h, P12PFS 0008 C14Ah, P13PFS 0008 C14Bh, P14PFS 0008 C14Ch, P15PFS 0008 C14Dh, P16PFS 0008 C14Eh, P17PFS 0008 C14Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.4 Register Settings for Input/Output Pin Function in 177-Pin TFLGA, 176-Pin LFBGA, 176-Pin LFQFP

	Pin							
PSEL[5:0] Settings	**P10**	**P11**	**P12**	**P13**	**P14**	**P15**	**P16**	**P17**
000000b (initial value)	Hi-Z							
000001b	MTIC5W	MTIC5V	MTIC5U	MTIOC0B	MTIOC3A	MTIOC0B	MTIOC3C	MTIOC3A
000010b	—	—	—	—	MTCLKA	MTCLKB	MTIOC3D	MTIOC3B
000011b	—	—	—	TIOCA5	TIOCB5	TIOCB2	TIOCB1	TIOCB0
000100b	—	—	—	—	TCLKA	TCLKB	TCLKC	TCLKD
000101b	TMRI3	TMCI3	TMCI1	TMO3	TMRI2	TMCI2	TMO2	TMO1
000110b	—	—	—	PO13	PO15	PO13	PO14	PO15
000111b	—	—	—	—	—	—	RTCOUT	POE8#
001000b	—	—	—	—	—	—	—	MTIOC4B
001001b	—	—	—	ADTRG1#	—	—	ADTRG0#	ADTRG1#
001010b	—	SCK2	RXD2 SMISO2 SSCL2	TXD2 SMOSI2 SSDA2	—	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1	SCK1
001011b	—	—	—	—	CTS1# RTS1# SS1#	SCK3	RXD3 SMISO3 SSCL3	TXD3 SMOSI3 SSDA3
001111b	—	—	SCL0[FM+]	SDA0[FM+]	—	—	SCL2-DS	SDA2-DS
010000b	—	—	—	—	CTX1	CRX1-DS	—	—
010001b	—	—	—	—	—	—	USB0_VBUS	—
010010b	—	—	—	—	USB0_OVRCURA	—	USB0_VBUSEN	—
010011b	—	—	—	—	—	—	USB0_OVRCURB	—
010100b	—	USBA_VBUS	—	—	—	—	—	—
010101b	USBA_OVRCURA	USBA_VBUSEN	—	—	—	USBA_VBUSEN	—	—
010111b	—	—	—	—	—	SSIWS1	—	SSITXD0
011100b	—	—	—	—	—	PIXD0	—	PIXD3
011110b	—	—	—	—	—	GTETRG	—	GTIOC0B

—: Do not specify this value.

Table 23.5 Register Settings for Input/Output Pin Function in 145-/100-Pin TFLGA, 144-/100-Pin LFQFP

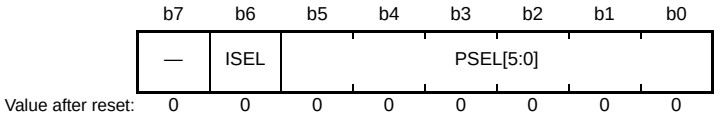
	Pin					
PSEL[5:0] Settings	**P12**	**P13**	**P14**	**P15**	**P16**	**P17**
000000b (initial value)	Hi-Z					
000001b	—	MTIOC0B	MTIOC3A	MTIOC0B	MTIOC3C	MTIOC3A
000010b	—	—	MTCLKA	MTCLKB	MTIOC3D	MTIOC3B
000011b	—	TIOCA5	TIOCB5	TIOCB2	TIOCB1	TIOCB0
000100b	—	—	TCLKA	TCLKB	TCLKC	TCLKD
000101b	TMCI1	TMO3	TMRI2	TMCI2	TMO2	TMO1
000110b	—	PO13	PO15	PO13	PO14	PO15
000111b	—	—	—	—	RTCOUT	POE8#
001000b	—	—	—	—	—	MTIOC4B
001001b	—	ADTRG1#	—	—	ADTRG0#	ADTRG1#
001010b	RXD2 SMISO2 SSCL2	TXD2 SMOSI2 SSDA2	—	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1	SCK1
001011b	—	—	CTS1# RTS1# SS1#	SCK3	RXD3 SMISO3 SSCL3	TXD3 SMOSI3 SSDA3
001111b	SCL0[FM+]	SDA0[FM+]	—	—	SCL2-DS	SDA2-DS
010000b	—	—	CTX1	CRX1-DS	—	—
010001b	—	—	—	—	USB0_VBUS	—
010010b	—	—	USB0_OVRCURA	—	USB0_VBUSEN	—
010011b	—	—	—	—	USB0_OVRCURB	—
010101b	—	—	—	—	—	—
010111b	—	—	—	SSIWS1	—	SSITXD0
011100b*1	—	—	—	PIXD0	—	PIXD3
011110b	—	—	—	GTETRG	—	GTIOC0B

—: Do not specify this value.

Note 1. This setting is not supported by 100-pin products.

23.2.4 P2n Pin Function Control Registers (P2nPFS) (n = 0 to 7)

Address(es): P20PFS 0008 C150h, P21PFS 0008 C151h, P22PFS 0008 C152h, P23PFS 0008 C153h, P24PFS 0008 C154h, P25PFS 0008 C155h, P26PFS 0008 C156h, P27PFS 0008 C157h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.6 Register Settings for Input/Output Pin Function in 177-/145-/100-Pin TFLGA, 176-Pin LFBGA, 176-/144-/100-Pin LFQFP

	Pin							
PSEL[5:0] Settings	**P20**	**P21**	**P22**	**P23**	**P24**	**P25**	**P26**	**P27**
000000b (initial value)	Hi-Z							
000001b	MTIOC1A	MTIOC1B	MTIOC3B	MTIOC3D	MTIOC4A	MTIOC4C	MTIOC2A	MTIOC2B
000010b	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB	—	—
000011b	TIOCB3	TIOCA3	TIOCC3	TIOCD3	TIOCB4	TIOCA4	—	—
000101b	TMRI0	TMCI0	TMO0	—	TMRI1	—	TMO1	TMCI3
000110b	PO0	PO1	PO2	PO3	PO4	PO5	PO6	PO7
001000b	—	MTIOC4A	—	—	—	—	—	—
001001b	—	—	—	—	—	ADTRG0#	—	—
001010b	TXD0 SMOSI0 SSDA0	RXD0 SMISO0 SSCL0	SCK0	TXD3 SMOSI3 SSDA3	SCK3	RXD3 SMISO3 SSCL3	TXD1 SMOSI1 SSDA1	SCK1
001011b	—	—	—	CTS0# RTS0# SS0#	—	—	CTS3# RTS3# SS3#	—
010011b	USB0_ID	USB0_EXICEN	USB0_OVRCURB	—	USB0_VBUSEN	—	—	—
010100b*1	—	—	—	—	—	—	ET1_EXOUT	ET1_WOL
010110b	USBA_ID	USBA_EXICEN	USBA_OVRCURB	—	—	—	—	—
011000b	—	—	EDREQ0	EDACK0	EDREQ1	EDACK1	—	—
011110b	—	GTIOC2A	GTIOC1A	GTIOC0A	—	—	—	—
010111b	SSIRXD0	SSIWS0	AUDIO_CLK	SSISCK0	SSISCK1	SSIDATA1	—	—
011100b*2	PIXD4	PIXD5	PIXD6	PIXD7	PIXCLK	HSYNC	—	—

—: Do not specify this value.

Note 1. This setting is only supported by 176- and 177-pin products.

Note 2. This setting is not supported by 100-pin products.

23.2.5 P3n Pin Function Control Registers (P3nPFS) (n = 0 to 4)

Address(es): P30PFS 0008 C158h, P31PFS 0008 C159h, P32PFS 0008 C15Ah, P33PFS 0008 C15Bh, P34PFS 0008 C15Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.7 Register Settings for Input/Output Pin Function in 177-/145-/100-Pin TFLGA, 176-Pin LFBGA, 176-/144-/100-Pin LFQFP

	Pin				
PSEL[5:0] Settings	**P30**	**P31**	**P32**	**P33**	**P34**
000000b (initial value)	Hi-Z				
000001b	MTIOC4B	MTIOC4D	MTIOC0C	MTIOC0D	MTIOC0A
000011b	—	—	TIOCC0	TIOCD0	—
000101b	TMRI3	TMCI2	TMO3	TMRI3	TMCI3
000110b	PO8	PO9	PO10	PO11	PO12
000111b	POE8#	—	RTCOUT	—	POE10#
001000b	—	—	POE0#	POE4#	—
001010b	RXD1 SMISO1 SSCL1	—	TXD6 SMOSI6 SSDA6	RXD6 SMISO6 SSCL6	SCK6
001011b	—	CTS1# RTS1# SS1#	TXD0 SMOSI0 SSDA0	RXD0 SMISO0 SSCL0	SCK0
010000b	—	—	CTX0	CRX0	—
010001b	—	—	—	—	ET0_LINKSTA
010011b	—	—	USB0_VBUSEN	—	—
010100b*1	ET1_MDIO	ET1_MDC	—	—	—
011000b	—	—	—	EDREQ1	—
011100b*2	—	—	VSYNC	PCKO	—
100001b	—	—	POE10#	POE11#	—

—: Do not specify this value.

Note 1. This setting is only supported by 176- and 177-pin products.

Note 2. This setting is not supported by 100-pin products.

23.2.6 P4n Pin Function Control Registers (P4nPFS) (n = 0 to 7)

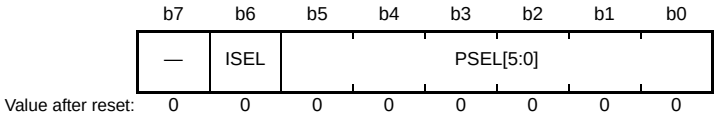
Address(es): P40PFS 0008 C160h, P41PFS 0008 C161h, P42PFS 0008 C162h, P43PFS 0008 C163h, P44PFS 0008 C164h, P45PFS 0008 C165h, P46PFS 0008 C166h, P47PFS 0008 C167h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin	R/W
b7	ASEL	Analog Input Function Select	0: Used other than as analog pin. 1: Used as analog pin.	R/W

23.2.7 P5n Pin Function Control Registers (P5nPFS) (n = 0 to 2, 4, to 6)

Address(es): P50PFS 0008 C168h, P51PFS 0008 C169h, P52PFS 0008 C16Ah, P54PFS 0008 C16Ch, P55PFS 0008 C16Dh, P56PFS 0008 C16Eh



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.8 Register Settings for Input/Output Pin Function in 177-Pin TFLGA, 176-Pin LFBGA, 176-Pin LFQFP

	Pin		
PSEL[5:0] Settings	**P50**	**P51**	**P52**
000000b (initial value)	Hi-Z		
001010b	TXD2 SMOSI2 SSDA2	SCK2	RXD2 SMISO2 SSCL2

Table 23.9 Register Settings for Input/Output Pin Function in 144-Pin LFQFP, 145-Pin TFLGA

	Pin					
PSEL[5:0] Settings	**P50**	**P51**	**P52**	**P54**	**P55**	**P56**
000000b (initial value)	Hi-Z					
000001b	—	—	—	MTIOC4B	MTIOC4D	MTIOC3C
000011b	—	—	—	—	—	TIOCA1
000101b	—	—	—	TMCI1	TMO3	—
001010b	TXD2 SMOSI2 SSDA2	SCK2	RXD2 SMISO2 SSCL2	—	—	—
001011b	—	—	—	CTS2# RTS2# SS2#	—	—
010000b	—	—	—	CTX1	CRX1	—
010001b	—	—	—	ET0_LINKSTA	ET0_EXOUT	—
011000b	—	—	—	EDACK0	EDREQ0	EDACK1

—: Do not specify this value.

Table 23.10 Register Settings for Input/Output Pin Function in 100-Pin TFLGA, 100-Pin LFQFP

	Pin				
PSEL[5:0] Settings	**P50**	**P51**	**P52**	**P54**	**P55**
000000b (initial value)	Hi-Z				
000001b	—	—	—	MTIOC4B	MTIOC4D
000101b	—	—	—	TMCI1	TMO3
001010b	TXD2 SMOSI2 SSDA2	SCK2	RXD2 SMISO2 SSCL2	—	—
001011b	—	—	—	CTS2# RTS2# SS2#	—
010000b	—	—	—	CTX1	CRX1
010001b	—	—	—	ET0_LINKSTA	ET0_EXOUT
011000b	—	—	—	EDACK0	EDREQ0

—: Do not specify this value.

23.2.8 P6n Pin Function Control Registers (P6nPFS) (n = 0, 6, 7)

Address(es): P60PFS 0008 C170h, P66PFS 0008 C176h, P67PFS 0008 C177h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.11 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LFQFP

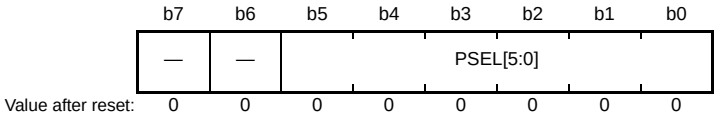
	Pin		
PSEL[5:0] Settings	**P60**	**P66**	**P67**
000000b (initial value)	Hi-Z		
001000b	—	MTIOC7D	MTIOC7C
010000b	—	CTX2	CRX2
010100b*1	ET1_TX_EN	—	—
010101b*1	RMII1_TXD_EN	—	—
011110b	—	GTIOC2B	GTIOC1B

—: Do not specify this value.

Note 1. This setting is only supported by 176- and 177-pin products.

23.2.9 P7n Pin Function Control Registers (P7nPFS) (n = 1 to 7)

Address(es): P71PFS 0008 C179h, P72PFS 0008 C17Ah, P73PFS 0008 C17Bh, P74PFS 0008 C17Ch, P75PFS 0008 C17Dh, P76PFS 0008 C17Eh, P77PFS 0008 C17Fh



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

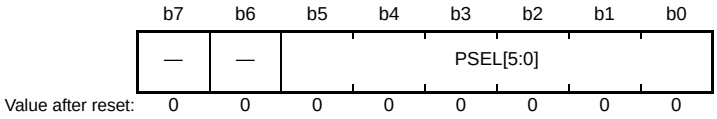
Table 23.12 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LFQFP

	Pin						
PSEL[5:0] Settings	**P71**	**P72**	**P73**	**P74**	**P75**	**P76**	**P77**
000000b (initial value)	Hi-Z						
000110b	—	—	PO16	PO19	PO20	PO22	PO23
001010b	—	—	—	—	SCK11	RXD11	TXD11
001011b	—	—	—	CTS11#	RTS11#	—	—
010001b	ET0_MDIO	ET0_MDC	ET0_WOL	ET0_ERXD1	ET0_ERXD0	ET0_RX_CLK	ET0_RX_ER
010010b	—	—	—	RMII0_RXD1	RMII0_RXD0	REF50CK0	RMII0_RX_ER
011001b	—	—	—	—	MMC_RES#	MMC_CMD	MMC_CLK
011010b	—	—	—	—	SDHI_D2	SDHI_CMD	SDHI_CLK
011011b	—	—	—	—	—	QSSL	QSPCLK

—: Do not specify this value.

23.2.10 P8n Pin Function Control Registers (P8nPFS) (n = 0 to 3, 6, 7)

Address(es): P80PFS 0008 C180h, P81PFS 0008 C181h, P82PFS 0008 C182h, P83PFS 0008 C183h, P86PFS 0008 C186h, P87PFS 0008 C187h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 23.13 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LFQFP

	Pin					
PSEL[5:0] Settings	**P80**	**P81**	**P82**	**P83**	**P86**	**P87**
000000b (initial value)	Hi-Z					
000001b	MTIOC3B	MTIOC3D	MTIOC4A	MTIOC4C	—	—
000011b	—	—	—	—	TIOCA0	TIOCA2
000110b	PO26	PO27	PO28	—	—	—
001000b	—	—	—	—	MTIOC4D	MTIOC4C
001010b	SCK10	RXD10	TXD10	SCK10	RXD10	TXD10
001011b	RTS10#	—	—	CTS10#	—	—
010001b	ET0_TX_EN	ET0_ETXD0	ET0_ETXD1	ET0_CRS	—	—
010010b	RMII0_TXD_EN	RMII0_TXD0	RMII0_TXD1	RMII0_CRS_DV	—	—
011000b	EDREQ0	EDACK0	EDREQ1	EDACK1	—	—
011100b	—	—	—	—	PIXD1	PIXD2
011001b	MMC_D2	MMC_D3	MMC_D4	—	—	—
011010b	SDHI_WP	SDHI_CD	—	—	—	—
011011b	QIO2	QIO3	—	—	—	—
011110b	—	GTIOC0B	GTIOC2A	GTIOC0A	GTIOC2B	GTIOC1B

—: Do not specify this value.

23.2.11 P9n Pin Function Control Registers (P9nPFS) (n = 0 to 7)

Address(es): P90PFS 0008 C188h, P91PFS 0008 C189h, P92PFS 0008 C18Ah, P93PFS 0008 C18Bh, P94PFS 0008 C18Ch, P95PFS 0008 C18Dh, P96PFS 0008 C18Eh, P97PFS 0008 C18Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	—	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	ASEL	Analog Input Function Select	0: Used other than as analog pin. 1: Used as analog pin.	R/W

Table 23.14 Register Settings for Input/Output Pin Function in 177-Pin TFLGA, 176-Pin LFBGA, 176-Pin LFQFP

	Pin							
PSEL[5:0] Settings	P90	P91	P92	P93	P94	P95	P96	P97
000000b (initial value)	Hi-Z							
001000b	—	—	POE4#	POE0#	—	—	—	—
001010b	TXD7 SMOSI7 SSDA7	SCK7	RXD7 SMISO7 SSCL7	—	—	—	—	—
001011b	—	—	—	CTS7# RTS7# SS7#	—	—	—	—
010100b	ET1_RX_DV	ET1_COL	ET1_CRS	ET1_LINKSTA	ET1_ERXD0	ET1_ERXD1	ET1_ERXD2	ET1_ERXD3
010101b	—	—	RMII1_CRS_DV	—	RMII1_RXD0	RMII1_RXD1	—	—

—: Do not specify this value.

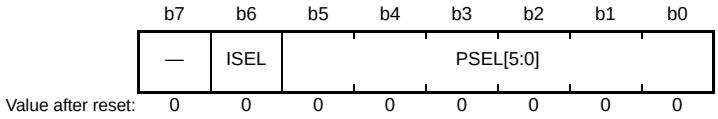
Table 23.15 Register Settings for Input/Output Pin Function in 145-Pin TFLGA, 144-Pin LFQFP

	Pin			
PSEL[5:0] Settings	P90	P91	P92	P93
000000b (initial value)	Hi-Z			
001000b	—	—	POE4#	POE0#
001010b	TXD7 SMOSI7 SSDA7	SCK7	RXD7 SMISO7 SSCL7	—
001011b	—	—	—	CTS7# RTS7# SS7#

—: Do not specify this value.

23.2.12 PAn Pin Function Control Registers (PAnPFS) (n = 0 to 7)

Address(es): PA0PFS 0008 C190h, PA1PFS 0008 C191h, PA2PFS 0008 C192h, PA3PFS 0008 C193h, PA4PFS 0008 C194h, PA5PFS 0008 C195h, PA6PFS 0008 C196h, PA7PFS 0008 C197h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.16 Register Settings for Input/Output Pin Function in 177-/145-/100-Pin TFLGA, 176-Pin LFBGA, 176-/144-/100-Pin LFQFP

	Pin							
PSEL[5:0] Settings	**PA0**	**PA1**	**PA2**	**PA3**	**PA4**	**PA5**	**PA6**	**PA7**
000000b (initial value)	Hi-Z							
000001b	MTIOC4A	MTIOC0B	—	MTIOC0D	MTIC5U	—	MTIC5V	—
000010b	—	MTCLKC	—	MTCLKD	MTCLKA	—	MTCLKB	—
000011b	TIOCA0	TIOCB0	—	TIOCD0	TIOCA1	TIOCB1	TIOCA2	TIOCB2
000100b	—	—	—	TCLKB	—	—	—	—
000101b	—	—	—	—	TMRI0	—	TMCI3	—
000110b	PO16	PO17	PO18	PO19	PO20	PO21	PO22	PO23
000111b	CACREF	—	—	—	—	—	POE10#	—
001000b	MTIOC6D	MTIOC7B	MTIOC7A	—	—	MTIOC6B	—	—
001010b	—	SCK5	RXD5 SMISO5 SSCL5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	—	—	—
001011b	—	—	—	—	—	—	CTS5# RTS5# SS5#	—
001101b	SSLA1	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA
010001b	ET0_TX_EN	ET0_WOL	—	ET0_MDIO	ET0_MDC	ET0_LINKSTA	ET0_EXOUT	ET0_WOL
010010b	RMII0_TXD_EN	—	—	—	—	—	—	—
011110b	GTIOC0B	GTIOC2A	GTIOC1A	—	—	GTIOC0A	GTETRG	—

—: Do not specify this value.

23.2.13 PBn Pin Function Control Registers (PBnPFS) (n = 0 to 7)

Address(es): PB0PFS 0008 C198h, PB1PFS 0008 C199h, PB2PFS 0008 C19Ah, PB3PFS 0008 C19Bh, PB4PFS 0008 C19Ch, PB5PFS 0008 C19Dh, PB6PFS 0008 C19Eh, PB7PFS 0008 C19Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.17 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LFQFP

	Pin							
PSEL[5:0] Settings	**PB0**	**PB1**	**PB2**	**PB3**	**PB4**	**PB5**	**PB6**	**PB7**
000000b (initial value)	Hi-Z							
000001b	MTIC5W	MTIOC0C	—	MTIOC0A	—	MTIOC2A	MTIOC3D	MTIOC3B
000010b	—	MTIOC4C	—	MTIOC4A	—	MTIOC1B	—	—
000011b	TIOCA3	TIOCB3	TIOCC3	TIOCD3	TIOCA4	TIOCB4	TIOCA5	TIOCB5
000100b	—	—	TCLKC	TCLKD	—	—	—	—
000101b	—	TMCI0	—	TMO0	—	TMRI1	—	—
000110b	PO24	PO25	PO26	PO27	PO28	PO29	PO30	PO31
000111b	—	—	—	POE11#	—	POE4#	—	—
001010b	RXD4 SMISO4 SSCL4	TXD4 SMOSI4 SSDA4	CTS4# RTS4# SS4#	SCK4	—	SCK9	RXD9	TXD9
001011b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	CTS6# RTS6# SS6#	SCK6	CTS9#	RTS9#	—	—
010001b	ET0_ERXD1	ET0_ERXD0	ET0_RX_CLK	ET0_RX_ER	ET0_TX_EN	ET0_ETXD0	ET0_ETXD1	ET0_CRS
010010b	RMII0_RXD1	RMII0_RXD0	REF50CK0	RMII0_RX_ER	RMII0_TXD_EN	RMII0_TXD0	RMII0_TXD1	RMII0_CRS_DV

—: Do not specify this value.

Table 23.18 Register Settings for Input/Output Pin Function in 100-Pin TFLGA, 100-Pin LFQFP

	Pin							
PSEL[5:0] Settings	**PB0**	**PB1**	**PB2**	**PB3**	**PB4**	**PB5**	**PB6**	**PB7**
000000b (initial value)	Hi-Z							
000001b	MTIC5W	MTIOC0C	—	MTIOC0A	—	MTIOC2A	MTIOC3D	MTIOC3B
000010b	—	MTIOC4C	—	MTIOC4A	—	MTIOC1B	—	—
000011b	TIOCA3	TIOCB3	TIOCC3	TIOCD3	TIOCA4	TIOCB4	TIOCA5	TIOCB5
000100b	—	—	TCLKC	TCLKD	—	—	—	—
000101b	—	TMCI0	—	TMO0	—	TMRI1	—	—
000110b	PO24	PO25	PO26	PO27	PO28	PO29	PO30	PO31
000111b	—	—	—	POE11#	—	POE4#	—	—
001010b	—	—	—	—	—	SCK9	RXD9	TXD9
001011b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	CTS6# RTS6# SS6#	SCK6	CTS9#	RTS9#	—	—
010001b	ET0_ERXD1	ET0_ERXD0	ET0_RX_CLK	ET0_RX_ER	ET0_TX_EN	ET0_ETXD0	ET0_ETXD1	ET0_CRS
010010b	RMII0_RXD1	RMII0_RXD0	REF50CK0	RMII0_RX_ER	RMII0_TXD_EN	RMII0_TXD0	RMII0_TXD1	RMII0_CRS_DV

—: Do not specify this value.

23.2.14 PCn Pin Function Control Register (PCnPFS) (n = 0 to 7)

Address(es): PC0PFS 0008 C1A0h, PC1PFS 0008 C1A1h, PC2PFS 0008 C1A2h, PC3PFS 0008 C1A3h, PC4PFS 0008 C1A4h, PC5PFS 0008 C1A5h, PC6PFS 0008 C1A6h, PC7PFS 0008 C1A7h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.19 Register Settings for Input/Output Pin Function in 177-/145-/100-Pin TFLGA, 176-Pin LFBGA, 176-/144-/100-Pin LFQFP

	Pin							
PSEL[5:0] Settings	**PC0**	**PC1**	**PC2**	**PC3**	**PC4**	**PC5**	**PC6**	**PC7**
000000b (initial value)	Hi-Z							
000001b	MTIOC3C	MTIOC3A	MTIOC4B	MTIOC4D	MTIOC3D	MTIOC3B	MTIOC3C	MTIOC3A
000010b	—	—	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB
000011b	TCLKC	TCLKD	TCLKA	TCLKB	—	—	—	—
000101b	—	—	—	—	TMCI1	TMRI2	TMCI2	TMO2
000110b	PO17	PO18	PO21	PO24	PO25	PO29	PO30	PO31
000111b	—	—	—	—	POE0#	—	—	CACREF
001010b	—	SCK5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	SCK5	SCK8	RXD8	TXD8
001011b	CTS5# RTS5# SS5#	—	—	—	CTS8#	RTS8#	—	—
001101b	SSLA1	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA
010001b	ET0_ERXD3	ET0_ERXD2	ET0_RX_DV	ET0_TX_ER	ET0_TX_CLK	ET0_ETXD2	ET0_ETXD3	ET0_COL
011001b*1	—	—	MMC_CD	MMC_D0	MMC_D1	MMC_D5	MMC_D6	MMC_D7
011010b*1	—	—	SDHI_CD	SDHI_D0	SDHI_D1	—	—	—
011011b*1	—	—	—	QIO0 QMO	QIO1 QMI	—	—	—
011101b	—	—	—	—	—	—	TIC0	TOC0
011110b	—	—	GTIOC2B	GTIOC1B	GTETRG	GTIOC1A	GTIOC3B	GTIOC3A

—: Do not specify this value.

Note 1. This setting is not supported by 100-pin products.

23.2.15 PDn Pin Function Control Register (PDnPFS) (n = 0 to 7)

Address(es): PD0PFS 0008 C1A8h, PD1PFS 0008 C1A9h, PD2PFS 0008 C1AAh, PD3PFS 0008 C1ABh, PD4PFS 0008 C1ACh, PD5PFS 0008 C1ADh, PD6PFS 0008 C1AEh, PD7PFS 0008 C1AFh

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin	R/W
b7	ASEL	Analog Input Function Select	0: Used other than as analog pin. 1: Used as analog pin.	R/W

Table 23.20 Register Settings for Input/Output Pin Function in 177-/145-/100-Pin TFLGA, 176-Pin LFBGA, 176-/144-/100-Pin LFQFP

	Pin							
PSEL[5:0] Settings	**PD0**	**PD1**	**PD2**	**PD3**	**PD4**	**PD5**	**PD6**	**PD7**
000000b (initial value)	Hi-Z							
000001b	—	MTIOC4B	MTIOC4D	—	—	MTIC5W	MTIC5V	MTIC5U
000111b	—	—	—	POE8#	POE11#	POE10#	POE4#	POE0#
001000b	POE4#	POE0#	—	MTIOC8D	MTIOC8B	MTIOC8C	MTIOC8A	—
010000b	—	CTX0	CRX0	—	—	—	—	—
011001b	—	—	MMC_D2	MMC_D3	MMC_CMD	MMC_CLK	MMC_D0	MMC_D1
011010b	—	—	SDHI_D2	SDHI_D3	SDHI_CMD	SDHI_CLK	SDHI_D0	SDHI_D1
011011b	—	—	QIO2	QIO3	QSSL	QSPCLK	QIO0 QMO	QIO1 QMI
011101b	—	—	TIC2	TOC2	—	—	—	—
011110b	GTIOC1B	GTIOC1A	GTIOC0B	GTIOC0A	—	—	—	—

—: Do not specify this value.

23.2.16 PEn Pin Function Control Register (PEnPFS) (n = 0 to 7)

Address(es): PE0PFS 0008 C1B0h, PE1PFS 0008 C1B1h, PE2PFS 0008 C1B2h, PE3PFS 0008 C1B3h, PE4PFS 0008 C1B4h, PE5PFS 0008 C1B5h, PE6PFS 0008 C1B6h, PE7PFS 0008 C1B7h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin	R/W
b7	ASEL	Analog Input Function Select	0: Used other than as analog pin. 1: Used as analog pin.	R/W

Table 23.21 Register Settings for Input/Output Pin Function in 177-/145-/100-Pin TFLGA, 176-Pin LFBGA, 176-/144-/100-Pin LFQFP

	Pin							
PSEL[5:0] Settings	**PE0**	**PE1**	**PE2**	**PE3**	**PE4**	**PE5**	**PE6**	**PE7**
000000b (initial value)	Hi-Z							
000001b	—	MTIOC4C	MTIOC4A	MTIOC4B	MTIOC4D	MTIOC4C	—	—
000010b	—	—	—	—	MTIOC1A	MTIOC2B	—	—
000110b	—	PO18	PO23	PO26	PO28	—	—	—
000111b	—	—	—	POE8#	—	—	—	—
001000b	MTIOC3D	MTIOC3B	—	—	—	—	MTIOC6C	MTIOC6A
001100b	SCK12	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12	CTS12# RTS12# SS12#	—	—	—	—
010001b	—	—	—	ET0_ERXD3	ET0_ERXD2	ET0_RX_CLK	—	—
010010b	—	—	—	—	—	REF50CK0	—	—
011001b	MMC_D4	MMC_D5	MMC_D6	MMC_D7	—	—	MMC_CD	MMC_RES#
011010b	—	—	—	—	—	—	SDHI_CD	SDHI_WP
011101b	—	—	TIC3	TOC3	—	—	TIC1	TOC1
011110b	GTIOC2B	GTIOC1B	GTIOC0B	GTIOC2A	GTIOC1A	GTIOC0A	GTIOC3B	GTIOC3A

—: Do not specify this value.

23.2.17 PFn Pin Function Control Register (PFnPFS) (n = 0 to 2, 5)

Address(es): PF0PFS 0008 C1B8h, PF1PFS 0008 C1B9h, PF2PFS 0008 C1BAh, PF5PFS 0008 C1BDh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.22 Register Settings for Input/Output Pin Function in 177-Pin TFLGA, 176-Pin LFBGA, 176-Pin LFQFP

	Pin		
PSEL[5:0] Settings	**PF0**	**PF1**	**PF2**
000000b (initial value)	Hi-Z		
001010b	TXD1 SMOSI1 SSDA1	SCK1	RXD1 SMISO1 SSCL1

23.2.18 PGn Pin Function Control Register (PGnPFS) (n = 0 to 7)

Address(es): PG0PFS 0008 C1C0h, PG1PFS 0008 C1C1h, PG2PFS 0008 C1C2h, PG3PFS 0008 C1C3h, PG4PFS 0008 C1C4h, PG5PFS 0008 C1C5h, PG6PFS 0008 C1C6h, PG7PFS 0008 C1C7h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 23.23 Register Settings for Input/Output Pin Function in 177-Pin TFLGA, 176-Pin LFBGA, 176-Pin LFQFP

	Pin							
PSEL[5:0] Settings	**PG0**	**PG1**	**PG2**	**PG3**	**PG4**	**PG5**	**PG6**	**PG7**
000000b (initial value)	Hi-Z							
010100b	ET1_RX_CLK	ET1_RX_ER	ET1_TX_CLK	ET1_ETXD0	ET1_ETXD1	ET1_ETXD2	ET1_ETXD3	ET1_TX_ER
010101b	REF50CK1	RMII1_RX_ER	—	RMII1_TXD0	RMII1_TXD1	—	—	—

—: Do not specify this value.

23.2.19 PJn Pin Function Control Register (PJnPFS) (n = 3, 5)

Address(es): PJ3PFS 0008 C1D3h, PJ5PFS 0008 C1D5h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see the table below.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 23.24 Register Settings for Input/Output Pin Function in 177-/145-/100-Pin TFLGA, 176-Pin LFBGA, 176-/144-/100-Pin LFQFP

	Pin	
PSEL[5:0] Settings	**PJ3**	**PJ5***1
000000b (initial value)	Hi-Z	
000001b	MTIOC3C	—
001010b	CTS6# RTS6# SS6#	—
001011b	CTS0# RTS0# SS0#	CTS2# RTS2# SS2#
010001b	ET0_EXOUT	—
011000b	EDACK1	—
100001b	—	POE8#

—: Do not specify this value.

Note 1. This setting is not supported by 100-pin products.

23.2.20 CS Output Enable Register (PFCSE)

Address(es): 0008 C100h

	b7	b6	b5	b4	b3	b2	b1	b0
	CS7E	CS6E	CS5E	CS4E	CS3E	CS2E	CS1E	CS0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CS0E	CS0 Enable	0: Disables CSn# output. 1: Enables CSn# output. (n = 0 to 7)	R/W
b1	CS1E	CS1 Enable		R/W
b2	CS2E	CS2 Enable		R/W
b3	CS3E	CS3 Enable		R/W
b4	CS4E	CS4 Enable		R/W
b5	CS5E	CS5 Enable		R/W
b6	CS6E	CS6 Enable		R/W
b7	CS7E	CS7 Enable		R/W

CSnE Bit (CSn Enable) (n = 0 to 7)

These bits enable or disable the corresponding pins to output the CSn# signal.

To enable output of the CSn# signal, set the corresponding CSnE bit in PFCSE to 1.

23.2.21 CS Output Pin Select Register 0 (PFCSS0)

Address(es): 0008 C102h

	b7	b6	b5	b4	b3	b2	b1	b0
	CS3S[1:0]		CS2S[1:0]		CS1S[1:0]		—	CS0S
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CS0S	CS0# Output Pin Select*1	0: Set P60 as CS0# output pin 1: Set PC7 as CS0# output pin	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3, b2	CS1S[1:0]	CS1# Output Pin Select*2	b3 b2 0 0: Set P61 as CS1# output pin 0 1: Set P71 as CS1# output pin 1 X: Set PC6 as CS1# output pin	R/W
b5, b4	CS2S[1:0]	CS2# Output Pin Select*3	b5 b4 0 0: Set P62 as CS2# output pin 0 1: Set P72 as CS2# output pin 1 X: Set PC5 as CS2# output pin	R/W
b7, b6	CS3S[1:0]	CS3# Output Pin Select*4	b7 b6 0 0: Set P63 as CS3# output pin 0 1: Set P73 as CS3# output pin 1 X: Set PC4 as CS3# output pin	R/W

X: Don't care

Note 1. P60 is not present in 100-pin products. Thus, even if this bit is set to "0b", PC7 is set as CS0# output pin.

Note 2. P61 and P71 are not present in 100-pin products. Thus, even if these bits are set to other than "1xb", PC6 is set as CS1# output pin.

Note 3. P62 and P72 are not present in 100-pin products. Thus, even if these bits are set to other than "1xb", PC5 is set as CS2# output pin.

Note 4. P63 and P73 are not present in 100-pin products. Thus, even if these bits are set to other than "1xb", PC4 is set as CS3# output pin.

CS0S Bit (CS0# Output Pin Select)

CSnS[1:0] Bits (CSn# Output Pin Select) (n = 1 to 3)

When CSn# output is enabled (the PFCSE.CSnE bit = 1), the CSn# output pin is selected.

23.2.22 CS Output Pin Select Register 1 (PFCSS1)

Address(es): 0008 C103h

	b7	b6	b5	b4	b3	b2	b1	b0
	CS7S[1:0]		CS6S[1:0]		CS5S[1:0]		CS4S[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CS4S[1:0]	CS4# Output Pin Select*1	b1 b0 0 0: Set P64 as CS4# output pin 0 1: Set P74 as CS4# output pin 1 X: Set P24 as CS4# output pin	R/W
b3, b2	CS5S[1:0]	CS5# Output Pin Select*2	b3 b2 0 0: Set P65 as CS5# output pin 0 1: Set P75 as CS5# output pin 1 X: Set P25 as CS5# output pin	R/W
b5, b4	CS6S[1:0]	CS6# Output Pin Select*3	b5 b4 0 0: Set P66 as CS6# output pin 0 1: Set P76 as CS6# output pin 1 X: Set P26 as CS6# output pin	R/W
b7, b6	CS7S[1:0]	CS7# Output Pin Select*4	b7 b6 0 0: Set P67 as CS7# output pin 0 1: Set P77 as CS7# output pin 1 X: Set P27 as CS7# output pin	R/W

X: Don't care

Note 1. P64 and P74 are not present in 100-pin products. Thus, even if these bits are set to other than "1xb", P24 is set as CS4# output pin.

Note 2. P65 and P75 are not present in 100-pin products. Thus, even if these bits are set to other than "1xb", P25 is set as CS5# output pin.

Note 3. P66 and P76 are not present in 100-pin products. Thus, even if these bits are set to other than "1xb", P26 is set as CS6# output pin.

Note 4. P67 and P77 are not present in 100-pin products. Thus, even if these bits are set to other than "1xb", P27 is set as CS7# output pin.

CSnS[1:0] Bits (CSn# Output Pin Select) (n = 4 to 7)

When CSn# output is enabled (the PFCSE.CSnE bit = 1), the CSn# output pin is selected.

23.2.23 Address Output Enable Register 0 (PFAOE0)

Address(es): 0008 C104h

	b7	b6	b5	b4	b3	b2	b1	b0
	A15E	A14E	A13E	A12E	A11E	A10E	A9E	A8E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	A8E	Address A8 Output Enable	0: Disables A8 output. 1: Enables A8 output.	R/W
b1	A9E	Address A9 Output Enable	0: Disables A9 output. 1: Enables A9 output.	R/W
b2	A10E	Address A10 Output Enable	0: Disables A10 output. 1: Enables A10 output.	R/W
b3	A11E	Address A11 Output Enable	0: Disables A11 output. 1: Enables A11 output.	R/W
b4	A12E	Address A12 Output Enable	0: Disables A12 output. 1: Enables A12 output.	R/W
b5	A13E	Address A13 Output Enable	0: Disables A13 output. 1: Enables A13 output.	R/W
b6	A14E	Address A14 Output Enable	0: Disables A14 output. 1: Enables A14 output.	R/W
b7	A15E	Address A15 Output Enable	0: Disables A15 output. 1: Enables A15 output.	R/W

23.2.24 Address Output Enable Register 1 (PFAOE1)

Address(es): 0008 C105h

	b7	b6	b5	b4	b3	b2	b1	b0
	A23E	A22E	A21E	A20E	A19E	A18E	A17E	A16E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	A16E	Address A16 Output Enable	0: Disables A16 output. 1: Enables A16 output.	R/W
b1	A17E	Address A17 Output Enable	0: Disables A17 output. 1: Enables A17 output.	R/W
b2	A18E	Address A18 Output Enable	0: Disables A18 output. 1: Enables A18 output.	R/W
b3	A19E	Address A19 Output Enable	0: Disables A19 output. 1: Enables A19 output.	R/W
b4	A20E	Address A20 Output Enable	0: Disables A20 output. 1: Enables A20 output.	R/W
b5	A21E	Address A21 Output Enable	0: Disables A21 output. 1: Enables A21 output.	R/W
b6	A22E	Address A22 Output Enable	0: Disables A22 output. 1: Enables A22 output.	R/W
b7	A23E	Address A23 Output Enable	0: Disables A23 output. 1: Enables A23 output.	R/W

23.2.25 External Bus Control Register 0 (PFBCR0)

Address(es): 0008 C106h

	b7	b6	b5	b4	b3	b2	b1	b0
	WR32BC32E	WR1BC1E	DH32E	DHE	BCLKO	ADRHMS2	ADRHMS	ADRLE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADRLE	A0 to A7 Output Enable	0: Configures PA0 to PA7 as the I/O port pins. 1: Configures PA0 to PA7 as the external address bus A0 to A7.	R/W
b1	ADRHMS	A16 to A23 Output Enable	See Table 23.25.	R/W
b2	ADRHMS2	A16 to A20 Output Enable		R/W
b3	BCLKO	BCLK Forced Output	0: BCLK is output when EXBE = 1 and not output when EXBE = 0. 1: BCLK is output regardless of the setting of EXBE.	R/W
b4	DHE	D8 to D15 Output Enable	0: Configures PE0 to PE7 as the I/O port pins. 1: Configures PE0 to PE7 as the external data bus D8 to D15.	R/W
b5	DH32E	D16 to D31 Output Enable	0: Set PG7 to PG0, and P97 to P90 as I/O port 1: Set PG7 to PG0, and P97 to P90 as external data bus D31 to D16.	R/W
b6	WR1BC1E	WR1#/BC1# Output Enable	0: Configures P51 as the I/O port pin. 1: Configures P51 as the WR1# or BC1# pin.	R/W
b7	WR32BC32E	WR3#/BC3# Output Enable WR2#/BC2# Output Enable	0: Set P13 and P12 as I/O port 1: Set P13 as WR#2 or BC2#, and set P12 as WR#3 or BC3#	R/W

BCLKO Bit (BCLK Forced Output)

This bit enables or disables forced output on the BCLK pin.

When this bit is set to 0, BCLK output is enabled or disabled according to the setting of EXBE; when this bit is 1, BCLK is output regardless of the setting of the EXBE bit.

Note that if the bit is set to 1, BCLK is output regardless of PMR.

[Setting procedure]

Output enabled: PSTOP1 (stopped) → BCLKO = 1 → PSTOP1 (stopped)

Output disabled: PSTOP1 (operating) → PSTOP1 (stopped) → BCLKO = 0

Table 23.25 Settings for External Address Buses A16 to A23

ADRHMS Bit	ADRHMS2 Bit	Settings for External Address Buses A16 to A23
0	0	Set PC0 to PC7.
0	1	Set PC0, PC1, P71, P72, P74, and PC5 to PC7.
1	0	Set P90 to P97.
1	1	Setting prohibited

23.2.26 External Bus Control Register 1 (PFBCR1)

Address(es): 0008 C107h

	b7	b6	b5	b4	b3	b2	b1	b0
	SDCLKE	DQM1E	—	MDSDE	ALES	ALEOE	WAITS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	WAITS[1:0]	WAIT Select	b1 b0 0 0: Setting invalid*1 0 1: Configures P55 as the WAIT# input pin.*2 1 0: Configures PC5 as the WAIT# input pin. 1 1: Configures P51 as the WAIT# input pin.	R/W
b2	ALEOE	ALE Output Enable	0: Disables ALE pin output. 1: Enables ALE pin output.	R/W
b3	ALES	ALE Select	0: Configures P54 as the ALE pin. (for the products other than 177- and 176-pin products) 1: Configures P10 as the ALE pin. (for 177- and 176-pin products)	R/W
b4	MDSDE	SDRAM Pin Enable	0: Disables SDRAM pin output. (CKE, SDCS#, RAS#, CAS#, WE#, and DQM0) 1: Enables SDRAM pin output. (CKE, SDCS#, RAS#, CAS#, WE#, and DQM0)	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	DQM1E	DQM1 Enable	0: Disables DQM1 output 1: Enables DQM1 output	R/W
b7	SDCLKE	SDCLK Enable	0: Disables SDCLK output 1: Enables SDCLK output	R/W

Note 1. Even if these bits are set to “00b” in 145-, 144-, and 100-pin products, P55 is set as WAIT# input pin.
Note 2. Setting is prohibited in 176- and 177-pin products.

WAITS[1:0] Bits (WAIT Select)

The port pin specified by the WAIT[1:0] bits becomes the WAIT# pin when the external bus is enabled. However, if the specified port pin is not to be used as the WAIT# pin, the external wait enable bit (EWENB) in the CSn mode register (CSnMOD) can be cleared (disabling external wait) to make the pin available for use as a general input port pin.
If the specified WAIT# pin is not to be used as the WAIT input or as a general input port pin, pull the level on the WAIT# pin up or down.

ALEOE Bit (ALE Output Enable)

This bit enables or disables output of ALE.

ALES Bit (ALE Select)

This bit selects the pins to output ALE.
To modify this bit, make sure that the ALEOE bit is set to 0. Furthermore, if the ALE pin is to be used, set this bit to 1 for 177- and 176-pin products, and to 0 for the products other than 177- and 176-pin products.

MDSDE Bit (SDRAM Pin Enable)

This bit enables or disables output of the SDRAM pin (CKE, SDCS#, RAS#, CAS#, WE#, and DQM0).
The DQM1 pin is enabled or disabled individually by the DQM1E bit, while the MDSDE bit is 1. The SDCLK pin is enabled or disabled independently by the SDCLKE bit, regardless of the MDSDE setting.

DQM1E Bit (DQM1 Enable)

This bit enables or disables output of the DQM1 pin.

When the MDSDE bit is set to 1, setting of the DQM1E bit is enabled. When the MDSDE bit is set to 0, setting of the DQM1E bit has no effect.

SDCLKE Bit (SDCLK Enable)

This bit enables or disables output of the SDCLK pin.

The SCKCR.PSTOP0 bit should be set to 1 before changing the setting of the SDCLKE bit.

23.2.27 Ethernet Control Register (PFENET)

Address(es): 0008 C10Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	PHYMODE1	PHYMODE0	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	PHYMODE0	Ethernet Channel 0 Mode Set	0: RMII mode (ETHERC0) 1: MII mode (ETHERC0)	R/W
b5	PHYMODE1	Ethernet Channel 1 Mode Set	0: RMII mode (ETHERC1) 1: MII mode (ETHERC1)	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PHYMODE0 Bit (Ethernet Channel 0 Mode Set)

This bit specifies the PHY mode of the ETHERC0.

Select the same mode as the one specified by the pin function select bits (PmnPFS.PSEL[5:0]).

When the signals for the RMII mode have been specified by the PmnPFS.PSEL[5:0] bits, set the PHYMODE bit to 0 (RMII mode), whereas when the signals for the MII mode have been specified by the PmnPFS.PSEL[5:0] bits, set the PHYMODE bit to 1 (MII mode).

PHYMODE1 Bit (Ethernet Channel 1 Mode Set)

This bit specifies the PHY mode of the ETHERC1.

Select the same mode as the one specified by the pin function select bits (PmnPFS.PSEL[5:0]).

When the signals for the RMII mode have been specified by the PmnPFS.PSEL[5:0] bits, set the PHYMODE bit to 0 (RMII mode), whereas when the signals for the MII mode have been specified by the PmnPFS.PSEL[5:0] bits, set the PHYMODE bit to 1 (MII mode).

23.3 How to Set the External Bus Interface

If the external bus interface is to be used, set the MPC registers according to Table 23.26 and then set the external bus enable bit (EXBE) in system control register 0 (SYSCR0) to 1.
Table 23.26 lists how to set up port pins to act as the external bus interface. For details on the relevant registers of the MPC, refer to section 23.2, Register Descriptions.

Table 23.26 How to Set the External Bus Interface (1/3)

Port	Output Signal	Settings of MPC Registers		
		177-Pin, 176-Pin	145-Pin, 144-Pin	100-Pin
P10	ALE	PFBCR1.ALEOE = 1, PFBCR1.ALES = 1	(not provided)	
P12	WR3#/BC3#	PFBCR0.WR32BC32E = 1		
P13	WR2#/BC2#	PFBCR0.WR32BC32E = 1		
P24	CS4#	PFCSE.CS4E = 1, PFCSS1.CS4S[1:0] = 10/11		
P25	CS5#	PFCSE.CS5E = 1, PFCSS1.CS5S[1:0] = 10/11		
P26	CS6#	PFCSE.CS6E = 1, PFCSS1.CS6S[1:0] = 10/11		
P27	CS7#	PFCSE.CS7E = 1, PFCSS1.CS7S[1:0] = 10/11		
P50	WR0#/WR#	—		
P51	WR1#/BC1#	PFBCR0.WR1BC1E = 1		
	WAIT#	PFBCR1.WAITS[1:0] = 11		
P52	RD#	—		
P53	BCLK	—		
P54	ALE	(not provided)	PFBCR1.ALEOE = 1, PFBCR1.ALES = 0	
P55	WAIT#	(not provided)	PFBCR1.WAITS[1:0] = 00/01	
P60	CS0#	PFCSE.CS0E = 1, PFCSS0.CS0S = 0		(not provided)
P61	CS1#	PFCSE.CS1E = 1, PFCSS0.CS1S[1:0] = 00		(not provided)
	SDCS#	PFBCR1.MDSDE = 1		(not provided)
P62	CS2#	PFCSE.CS2E = 1, PFCSS0.CS2S[1:0] = 00		(not provided)
	RAS#	PFBCR1.MDSDE = 1		(not provided)
P63	CS3#	PFCSE.CS3E = 1, PFCSS0.CS3S[1:0] = 00		(not provided)
	CAS#	PFBCR1.MDSDE = 1		(not provided)
P64	CS4#	PFCSE.CS4E = 1, PFCSS1.CS4S[1:0] = 00		(not provided)
	WE#	PFBCR1.MDSDE = 1		(not provided)
P65	CS5#	PFCSE.CS5E = 1, PFCSS1.CS5S[1:0] = 00		(not provided)
	CKE	PFBCR1.MDSDE = 1		(not provided)
P66	CS6#	PFCSE.CS6E = 1, PFCSS1.CS6S[1:0] = 00		(not provided)
	DQM0	PFBCR1.MDSDE = 1		(not provided)
P67	CS7#	PFCSE.CS7E = 1, PFCSS1.CS7S[1:0] = 00		(not provided)
	DQM1	PFBCR1.MDSDE = 1, PFBCR1.DQM1E = 1		(not provided)
P70	SDCLK	PFBCR1.SDCLKE = 1		(not provided)
P71	CS1#	PFCSE.CS1E = 1, PFCSS0.CS1S[1:0] = 01		(not provided)
	A18	PFAOE1.A18E = 1, PFBCR0.ADRHMS = 0, PFBCR0.ADRHMS2 = 1		(not provided)
P72	CS2#	PFCSE.CS2E = 1, PFCSS0.CS2S[1:0] = 01		(not provided)
	A19	PFAOE1.A19E = 1, PFBCR0.ADRHMS = 0, PFBCR0.ADRHMS2 = 1		(not provided)
P73	CS3#	PFCSE.CS3E = 1, PFCSS0.CS3S[1:0] = 01		(not provided)
P74	CS4#	PFCSE.CS4E = 1, PFCSS1.CS4S[1:0] = 01		(not provided)
	A20	PFAOE1.A20E = 1, PFBCR0.ADRHMS = 0, PFBCR0.ADRHMS2 = 1		(not provided)
P75	CS5#	PFCSE.CS5E = 1, PFCSS1.CS5S[1:0] = 01		(not provided)

Table 23.26 How to Set the External Bus Interface (2/3)

Port	Output Signal	Settings of MPC Registers		
		177-Pin, 176-Pin	145-Pin, 144-Pin	100-Pin
P76	CS6#	PFCSE.CS6E = 1, PFCSS1.CS6S[1:0] = 01		(not provided)
P77	CS7#	PFCSE.CS7E = 1, PFCSS1.CS7S[1:0] = 01		(not provided)
P90	D16	PFBCR0.DH32E = 1	(not provided)	(not provided)
	A16	PFAOE1.A16E = 1, PFBCR0.ADRHMS = 1		(not provided)
P91	D17	PFBCR0.DH32E = 1	(not provided)	(not provided)
	A17	PFAOE1.A17E = 1, PFBCR0.ADRHMS = 1		(not provided)
P92	D18	PFBCR0.DH32E = 1	(not provided)	(not provided)
	A18	PFAOE1.A18E = 1, PFBCR0.ADRHMS = 1		(not provided)
P93	D19	PFBCR0.DH32E = 1	(not provided)	(not provided)
	A19	PFAOE1.A19E = 1, PFBCR0.ADRHMS = 1		(not provided)
P94	D20	PFBCR0.DH32E = 1	(not provided)	(not provided)
	A20	PFAOE1.A20E = 1, PFBCR0.ADRHMS = 1	(not provided)	(not provided)
P95	D21	PFBCR0.DH32E = 1	(not provided)	(not provided)
	A21	PFAOE1.A21E = 1, PFBCR0.ADRHMS = 1	(not provided)	(not provided)
P96	D22	PFBCR0.DH32E = 1	(not provided)	(not provided)
	A22	PFAOE1.A22E = 1, PFBCR0.ADRHMS = 1	(not provided)	(not provided)
P97	D23	PFBCR0.DH32E = 1	(not provided)	(not provided)
	A23	PFAOE1.A23E = 1, PFBCR0.ADRHMS = 1	(not provided)	(not provided)
PA0	A0	PFBCR0.ADRLE = 1, CSnMOD.WRMOD = 0		
	BC0#	PFBCR0.ADRLE = 1, CSnMOD.WRMOD = 1		
	DQM2	PFBCR0.ADRLE = 1, SDCCR.EXENB = 1, SDCCR.BSIZE[1:0] = 01	(not provided)	(not provided)
PA1	A1	PFBCR0.ADRLE = 1		
	DQM3	PFBCR0.ADRLE = 1, SDCCR.EXENB = 1, SDCCR.BSIZE[1:0] = 01	(not provided)	(not provided)
PA2	A2	PFBCR0.ADRLE = 1		
PA3	A3	PFBCR0.ADRLE = 1		
PA4	A4	PFBCR0.ADRLE = 1		
PA5	A5	PFBCR0.ADRLE = 1		
PA6	A6	PFBCR0.ADRLE = 1		
PA7	A7	PFBCR0.ADRLE = 1		
PB0	A8	PFAOE0.A8E = 1		
PB1	A9	PFAOE0.A9E = 1		
PB2	A10	PFAOE0.A10E = 1		
PB3	A11	PFAOE0.A11E = 1		
PB4	A12	PFAOE0.A12E = 1		
PB5	A13	PFAOE0.A13E = 1		
PB6	A14	PFAOE0.A14E = 1		
PB7	A15	PFAOE0.A15E = 1		
PC0	A16	PFAOE1.A16E = 1, PFBCR0.ADRHMS = 0		
PC1	A17	PFAOE1.A17E = 1, PFBCR0.ADRHMS = 0		

Table 23.26 How to Set the External Bus Interface (3/3)

Port	Output Signal	Settings of MPC Registers		
		177-Pin, 176-Pin	145-Pin, 144-Pin	100-Pin
PC2	A18	PFAOE1.A18E = 1, PFBCR0.ADRHMS = 0, PFBCR0.ADRHMS2 = 0		
PC3	A19	PFAOE1.A19E = 1, PFBCR0.ADRHMS = 0, PFBCR0.ADRHMS2 = 0		
PC4	A20	PFAOE1.A20E = 1, PFBCR0.ADRHMS = 0, PFBCR0.ADRHMS2 = 0		
	CS3#	PFCSE.CS3E = 1, PFCSS0.CS3S[1:0] = 10/11		
PC5	A21	PFAOE1.A21E = 1, PFBCR0.ADRHMS = 0		
	CS2#	PFCSE.CS2E = 1, PFCSS0.CS2S[1:0] = 10/11		
	WAIT#	PFBCR1.WAITS[1:0] = 10		
PC6	A22	PFAOE1.A22E = 1, PFBCR0.ADRHMS = 0		
	CS1#	PFCSE.CS1E = 1, PFCSS0.CS1S[1:0] = 10/11		
PC7	A23	PFAOE1.A23E = 1, PFBCR0.ADRHMS = 0		
	CS0#	PFCSE.CS0E = 1, PFCSS0.CS0S = 1		
PD0	D0[A0/D0]	—		
PD1	D1[A1/D1]	—		
PD2	D2[A2/D2]	—		
PD3	D3[A3/D3]	—		
PD4	D4[A4/D4]	—		
PD5	D5[A5/D5]	—		
PD6	D6[A6/D6]	—		
PD7	D7[A7/D7]	—		
PE0	D8[A8/D8]	PFBCR0.DHE = 1		
PE1	D9[A9/D9]	PFBCR0.DHE = 1		
PE2	D10[A10/D10]	PFBCR0.DHE = 1		
PE3	D11[A11/D11]	PFBCR0.DHE = 1		
PE4	D12[A12/D12]	PFBCR0.DHE = 1		
PE5	D13[A13/D13]	PFBCR0.DHE = 1		
PE6	D14[A14/D14]	PFBCR0.DHE = 1		
PE7	D15[A15/D15]	PFBCR0.DHE = 1		
PG0	D24	PFBCR0.DH32E = 1	(not provided)	(not provided)
PG1	D25	PFBCR0.DH32E = 1	(not provided)	(not provided)
PG2	D26	PFBCR0.DH32E = 1	(not provided)	(not provided)
PG3	D27	PFBCR0.DH32E = 1	(not provided)	(not provided)
PG4	D28	PFBCR0.DH32E = 1	(not provided)	(not provided)
PG5	D29	PFBCR0.DH32E = 1	(not provided)	(not provided)
PG6	D30	PFBCR0.DH32E = 1	(not provided)	(not provided)
PG7	D31	PFBCR0.DH32E = 1	(not provided)	(not provided)

23.4 Usage Notes

23.4.1 Procedure for Specifying Input/Output Pin Function

Use the following procedure to specify the input/output pin functions.

(1) Clear the pin mode register (PMR) for the target pin to 0 to select the general I/O port.
(2) Specify the assignments of input/output signals for peripheral modules to the desired pins.
(3) Enable writing to the Pmn pin function control register (PmnPFS) through the write-protect register (PWPR) setting. (m = 0 to 9, A to G, and J, n = 0 to 7)
(4) Specify the input/output function for the pin through the PSEL[5:0] bit settings in the PmnPFS register.
(5) Clear the PFSWE bit in the PWPR register to 0 to disable writing to the PmnPFS register.
(6) Set the PMR to 1 as necessary to switch to the selected input/output function for the pin.

23.4.2 Notes on MPC Register Setting

(1) Settings of the Pmn pin function control register (PmnPFS) should be made only while the PMR register for the target pin is set to 0. If a Pny pin function control register is set while the PMR register is 1, unexpected edges may be input through the input pin or unexpected pulses are output through the output pin.
(2) Only the allowed values (functions) should be specified in the Pmn pin function control registers. If a value that is not allowed for the register is specified, correct operation is not guaranteed.
(3) Do not assign a single function to multiple pins through the MPC settings.
(4) Ports 0, 4, 9, D, and E also function as analog Input/output pins for the A/D converter and D/A converter. When using these ports as analog Input/output pins, set them to general input pins by clearing the corresponding bits in the port mode register (PMR) and port direction register (PDR) to 0 and set the PmnPFS.ASEL bit to 1, to avoid degradation of accuracy.
(5) The initial value of the time capture event input pin enable bit (TCEN) of the time capture control register y (RTCCRy, y = 0 to 2) is undefined after a reset. Therefore, set this bit to 0 to avoid unnecessary input.
(6) Points to note regarding the port mode register (PMR), port direction register (PDR), and Pmj pin function control register (PmnPFS) settings for pins that have multiplexed pin functions are listed in Table 23.27. The pin states are readable if the value of the ASEL bit is 0. Ensure that the PMR.Bj bit is 0 when changes to the PSEL[5:0] bits are made.

Table 23.27 Register Settings

Item	PMR.Bn	PDR.Bn	PmnPFS			Point to Note
			ASEL	ISEL	PSEL[5:0]	
After a reset	0	0	0	0	000000b	Pins function as general input port pins after release from the reset state.
General input ports	0	0	0	0/1	x	Set the PmnPFS.ISEL bit to 1 if these are multiplexed with interrupt inputs.
General output ports	0	1	0	0	x	
Peripheral functions	1	x	0	0/1	Peripheral functions (see Table 23.2 to Table 23.24)	Set the PmnPFS.ISEL bit to 1 if these are multiplexed with interrupt inputs.
Interrupt inputs	0	0	0	1	x	
NMI	x	x	x	x*1	x	Register settings are not required.
Analog inputs and outputs	0	0	1	x*1	x	Set these as general input port pins so that the output buffers are turned off.
Time-capture event-input pins	0	0	x	0/1	x	Set these as general input port pins so that the output buffers are turned off.
External bus	0	x*2	0	0	x	Set the PMR.Bn bit to 0 and do not select the peripheral function.
JTAG interface	0	x	x	0	x	Set the PMR.Bn bit and the PmnPFS.ISEL bit to 0 and switch the input buffers off.
FINE interface	0	x	x	0	x	Set the PMR.Bn bit and the PmnPFS.ISEL bit to 0 and switch the input buffers off.
EXTAL/XTAL	0	0	x	x*1	x	Set these as general input port pins so that the output buffers are turned off.

x: Setting not required.

0/1: Setting the PmnPFS.ISEL bit to 0 makes the pin incapable of functioning as an IRQ pin.
Setting the PmnPFS.ISEL bit to 1 makes the pin capable of functioning as an IRQ pin (if the IRQ is selected from the multiplexed functions).

Note 1. Even if the PmnPFS.ISEL bit is set to 1, the pin will not function as an IRQn input pin.

Note 2. To use the WAIT# input pin, set the corresponding bit in the PORTm.PDR register to 0.

Note:
- The pin state is readable when the PmnPFS.ASEL bit is 0.
- If the value of the PmnPFS.PSEL[5:0] bits is to be changed, do so while the PMR.Bn bit is 0.
- If an RIIC function is assigned to a port pin, clear the PMR.Bn (to 0); pulling up is automatically turned off for outputs from peripheral modules other than the RIIC.
- If an input pin for time-capture events is not in use, clear the time capture event input pin enable bit (TCEN) in time capture control register y (RTCCRy) (y = 0 to 2) to 0 (disabled). The value of the RTCCRy.TCEN bit after a reset is undefined.
- Do not make settings to assign multiple external bus signals to a single pin.

23.4.3 Notes on the Use of Analog Functions

To use an analog function, set the corresponding bits in both the port mode register (PMR) and port direction register (PDR) to 0 so that the pin acts as a general input port. After that, set the pin function select bit in the Pmn pin function control register (PmnPFS.ASEL) to 1.

24. Multi-Function Timer Pulse Unit (MTU3a)

24.1 Overview

This MCU has an on-chip multi-function timer pulse unit (MTU3a), consisting of eight 16-bit timer channels and one 32-bit timer channel.

Table 24.1 shows the specifications of the MTU and Table 24.2 lists the functions of the MTU. Figure 24.1 and Figure 24.2 show block diagrams of the MTU.

Table 24.1 MTU Specifications

Item	Description
Pulse input/output	28 lines max.
Pulse input	3 lines
Count clock	11 clocks for each channel (14 clocks for MTU0, 12 clocks for MTU2, 10 clocks for MTU5, and four clocks for MTU1 & MTU2 (LWA = 1))
Operating frequency	Up to 120 MHz
Available operations	[MTU0 to MTU4, MTU6, MTU7, MTU8] • Waveform output on compare match • Input capture function (noise filter setting available) • Counter-clearing operation • Simultaneous writing to multiple timer counters (TCNT) (excluding MTU8) • Simultaneous clearing on compare match or input capture (excluding MTU8) • Simultaneous input and output to registers in synchronization with counter operations (excluding MTU8) • Up to 12-phase PWM output in combination with synchronous operation (excluding MTU8)
	[MTU0, MTU3, MTU4, MTU6, MTU7, MTU8] • Buffer operation specifiable
	[MTU1, MTU2] • Phase counting mode can be specified independently • 32-bit phase counting mode can be specified for interlocked operation of MTU1 and MTU2 (when TMDR3.LWA = 1) • Cascade connection operation available
	[MTU3, MTU4, MTU6, MTU7] • Through interlocked operation of MTU3/4 and MTU6/7, the positive and negative signals in six phases (12 phases in total) can be output in complementary PWM and reset PWM operation. • In complementary PWM mode, transfer of values from buffer registers to temporary registers on peaks and troughs of the timer-counter values or writing to the buffer registers (MTU4.TGRD and MTU7.TGRD) • Double-buffering selectable in complementary PWM mode
	[MTU3, MTU4] • Through interlocking with MTU0, a mode for driving AC synchronous motors (brushless DC motors) by using complementary PWM output and reset PWM output is settable and allows the selection of two types of waveform output (chopping or level)
	[MTU5] • Capable of operation as a dead-time compensation counter
	[MTU0/MTU5, MTU1, MTU2, MTU8] 32-bit phase counting mode specifiable by combining MTU1 and MTU2 and through interlocked operation with MTU0/MTU5 and MTU8
Interrupt skipping function	• In complementary PWM mode, interrupts on crests and troughs of counter values and triggers to start conversion by the A/D converter can be skipped
Interrupt sources	43 sources
Buffer operation	Automatic transfer of register data (transfer from the buffer register to the timer register)
Trigger generation	A/D converter start triggers can be generated
	A/D converter start request delaying function enables A/D converter to be started with any desired timing and to be synchronized with PWM output
Low power consumption function	Module stop mode can be set

Table 24.2 MTU Functions (1/2)

Item	MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU5	MTU6	MTU7	MTU8
Count clock	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB MTCLKC MTCLKD MTIOC1A	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB MTCLKC	MTCLKA MTCLKB MTCLKC MTCLKD	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTIOC1A	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB
External clock for phase counting mode	—	MTCLKA MTCLKB	MTCLKA MTCLKB MTCLKC MTCLKD	MTCLKA MTCLKB MTCLKC MTCLKD	—	—	—	—	—	—
General registers (TGR)	TGRA TGRB TGRE	TGRA TGRB	TGRA TGRB	TGRALW TGRBLW	TGRA TGRB	TGRA TGRB	TGRU TGRV TGRW	TGRA TGRB	TGRA TGRB	TGRA TGRB
General registers/ buffer registers	TGRC TGRD TGRF	—	—	—	TGRC TGRD TGRE	TGRC TGRD TGRE TGRF	—	TGRC TGRD TGRE	TGRC TGRD TGRE TGRF	TGRC TGRD
I/O pins	MTIOC0A MTIOC0B MTIOC0C MTIOC0D	MTIOC1A MTIOC1B	MTIOC2A MTIOC2B	MTIOC1A MTIOC1B	MTIOC3A MTIOC3B MTIOC3C MTIOC3D	MTIOC4A MTIOC4B MTIOC4C MTIOC4D	MTIC5U MTIC5V MTIC5W	MTIOC6A MTIOC6B MTIOC6C MTIOC6D	MTIOC7A MTIOC7B MTIOC7C MTIOC7D	MTIOC8A MTIOC8B MTIOC8C MTIOC8D
Counter clear function	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGRALW/ TGRBLW input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output: 0 output	✓	✓	✓	—	✓	✓	—	✓	✓	✓
Compare match output: 1 output	✓	✓	✓	—	✓	✓	—	✓	✓	✓
Compare match output: Toggle output	✓	✓	✓	—	✓	✓	—	✓	✓	✓
Input capture function	✓	✓	✓	✓*1	✓	✓	✓	✓	✓	✓*2
Synchronous operation	✓	✓	✓	—	✓	✓	—	✓	✓	—
PWM mode 1	✓	✓	✓	—	✓	✓	—	✓	✓	—
PWM mode 2	✓	✓	✓	—	—	—	—	—	—	—
Complementary PWM mode	—	—	—	—	✓	✓	—	✓	✓	—
Reset-synchronized PWM mode	—	—	—	—	✓	✓	—	✓	✓	—
AC synchronous motor drive mode	✓	—	—	—	✓	✓	—	—	—	—
Phase counting mode	—	✓	✓	✓	—	—	—	—	—	—
Buffer operation	✓	—	—	—	✓	✓	—	✓	✓	✓
Dead time compensation counter function	—	—	—	—	—	—	✓	—	—	—
DMAC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGRALW/ TGRBLW input capture	TGR compare match or input capture	TGR compare match or input capture and TCNT overflow/ underflow (only in complementary PWM mode)	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture and TCNT overflow/ underflow (only in complementary PWM mode)	TGR compare match or input capture
DTC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGRALW/ TGRBLW input capture	TGR compare match or input capture	TGR compare match or input capture and TCNT overflow/ underflow (only in complementary PWM mode)	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture and TCNT overflow/ underflow (only in complementary PWM mode)	TGR compare match or input capture

Table 24.2 MTU Functions (2/2)

Item	MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU5	MTU6	MTU7	MTU8
A/D converter start trigger	TGRA compare match or input capture TGRE compare match	TGRA compare match or input capture	TGRA compare match or input capture	TGRALW input capture	TGRA compare match or input capture	TGRA compare match or input capture, or TCNT underflow (trough) in complementary PWM mode	—	TGRA compare match or input capture	TGRA compare match or input capture, or TCNT underflow (trough) in complementary PWM mode	—
Interrupt sources	Seven sources •Compare match or input capture 0A •Compare match or input capture 0B •Compare match or input capture 0C •Compare match or input capture 0D •Compare match0E •Compare match0F •Overflow	Four sources •Compare match or input capture 1A •Compare match or input capture 1B •Overflow •Underflow	Four sources •Compare match or input capture 2A •Compare match or input capture 2B •Overflow •Underflow	Four sources •Input capture A •Input capture B •Overflow •Underflow	Five sources •Compare match or input capture 3A •Compare match or input capture 3B •Compare match or input capture 3C •Compare match or input capture 3D •Overflow	Five sources •Compare match or input capture 4A •Compare match or input capture 4B •Compare match or input capture 4C •Compare match or input capture 4D •Overflow or underflow (only in com-plementary PWM mode)	Three sources •Compare match or input capture 5U •Compare match or input capture 5V •Compare match or input capture 5W	Five sources •Compare match or input capture 6A •Compare match or input capture 6B •Compare match or input capture 6C •Compare match or input capture 6D •Overflow	Five sources •Compare match or input capture 7A •Compare match or input capture 7B •Compare match or input capture 7C •Compare match or input capture 7D •Overflow or underflow (only in com-plementary PWM mode)	Five sources •Compare match or input capture 8A •Compare match or input capture 8B •Compare match or input capture 8C •Compare match or input capture 8D •Overflow
Event link function (output)	Seven sources •Compare match 0A •Compare match 0B •Compare match 0C •Compare match 0D •Compare match 0E •Compare match 0F •Overflow	—	—	—	Five sources •Compare match 3A •Compare match 3B •Compare match 3C •Compare match 3D •Overflow	Six sources •Compare match 4A •Compare match 4B •Compare match 4C •Compare match 4D •Overflow •Underflow (only in com-plementary PWM mode)	—	—	—	—
Event link function (input)	•Start counting •Input capture (to be captured in the TGRA) •Clear [Restart] counting	—	—	—	•Start counting •Input capture (to be captured in the TGRA) •Clear [Restart] counting	•Start counting •Input capture (to be captured in the TGRA) •Clear [Restart] counting	—	—	—	—
A/D converter start request delaying function	—	—	—	—	—	•A/D converter start request at a match between TADCORA and TCNT or A/D converter start request at a match between TADCORB and TCNT	—	—	•A/D converter start request at a match between TADCORA and TCNT or A/D converter start request at a match between TADCORB and TCNT	—
Interrupt skipping 1	—	—	—	—	•Skips TGRA compare match interrupts	•Skips TCIV interrupts	—	•Skips TGRA compare match interrupts	•Skips TCIV interrupts	—
Interrupt skipping 2	—	—	—	—	—	•Skipping in compare count between TADCORA and TCNT, and TADCORB and TCNT	—	—	•Skipping in compare count between TADCORA and TCNT, and TADCORB and TCNT	—
Module stop function	MSTPCRA.MSTPA9*3									

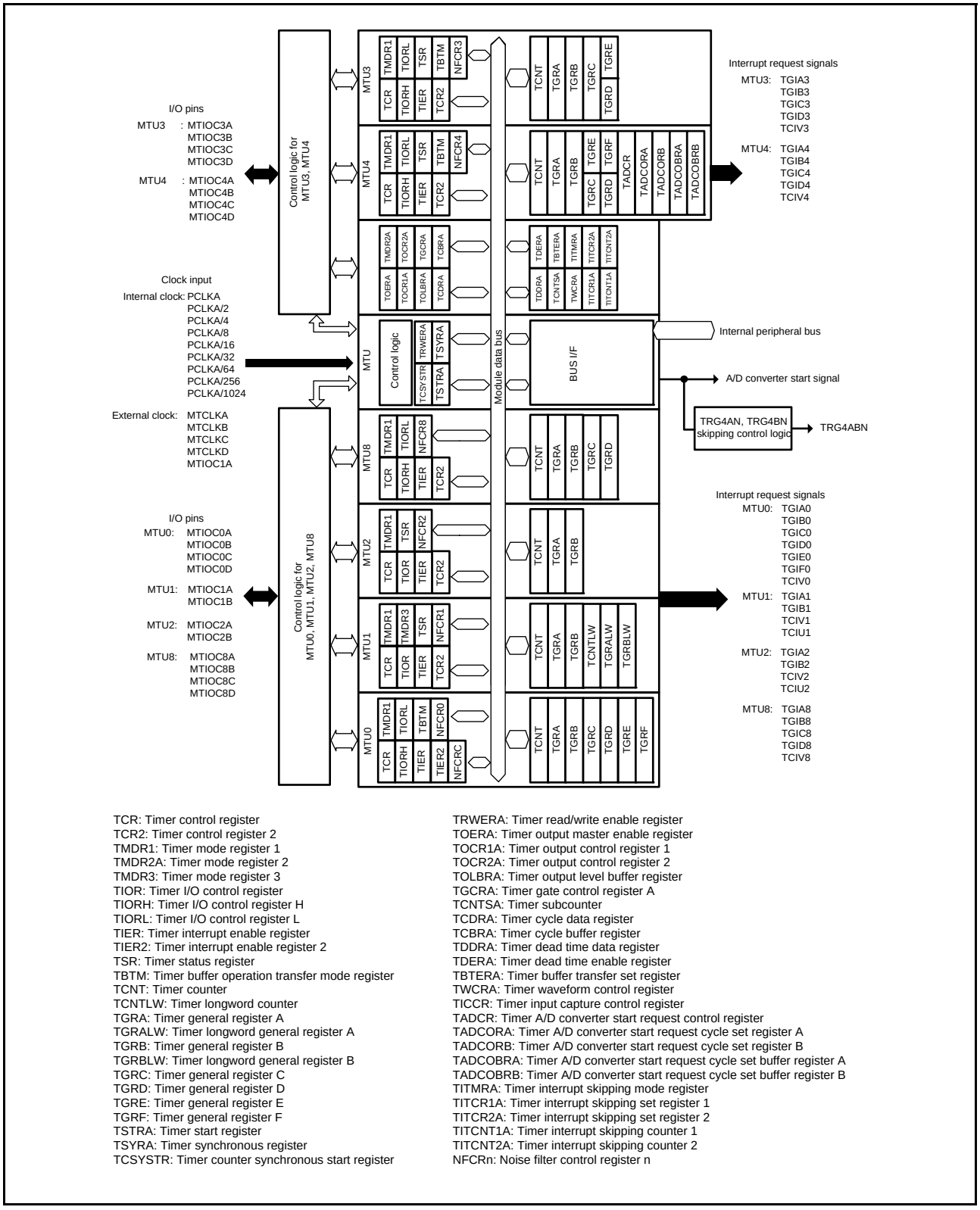
✓: Possible —: Not possible

Note 1. When LWA is 1, the TGRALW capture source can be selected from either of the following: an input from MTIOC1A or

MTU0.TGRA compare match/input capture event.
The TGRBLW capture source can be selected from any of the following: an input from MTIOC1B, MTU0.TGRC compare match/ input capture event, or MTU8.TGRC compare match event.

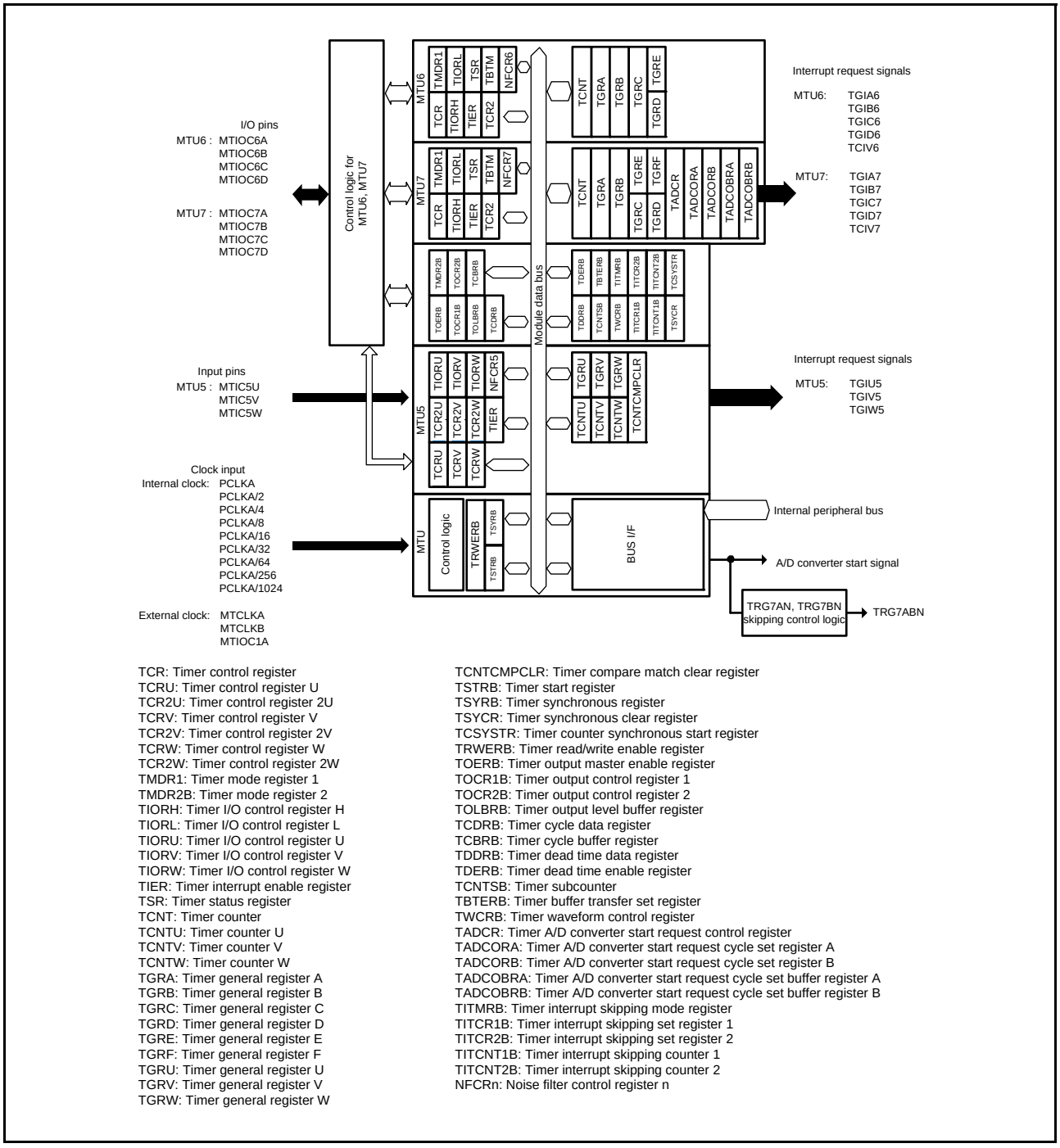
Note 2. Capture in MTU8 is supported only in normal mode.

Note 3. For details on the module stop function, refer to section 11, Low Power Consumption.



TCR: Timer control register
TCR2: Timer control register 2
TMDR1: Timer mode register 1
TMDR2A: Timer mode register 2
TMDR3: Timer mode register 3
TIOR: Timer I/O control register
TIORH: Timer I/O control register H
TIORL: Timer I/O control register L
TIER: Timer interrupt enable register
TIER2: Timer interrupt enable register 2
TSR: Timer status register
TBTM: Timer buffer operation transfer mode register
TCNT: Timer counter
TCNTLW: Timer longword counter
TGRA: Timer general register A
TGRALW: Timer longword general register A
TGRB: Timer general register B
TGRBLW: Timer longword general register B
TGRC: Timer general register C
TGRD: Timer general register D
TGRE: Timer general register E
TGRF: Timer general register F
TSTRA: Timer start register
TSYRA: Timer synchronous register
TCSYSTR: Timer counter synchronous start register
TRWERA: Timer read/write enable register
TOERA: Timer output master enable register
TOCR1A: Timer output control register 1
TOCR2A: Timer output control register 2
TOLBRA: Timer output level buffer register
TGCRA: Timer gate control register A
TCNTSA: Timer subcounter
TCDRA: Timer cycle data register
TCBRA: Timer cycle buffer register
TDDRA: Timer dead time data register
TDERA: Timer dead time enable register
TBTERA: Timer buffer transfer set register
TWCRA: Timer waveform control register
TICCR: Timer input capture control register
TADCR: Timer A/D converter start request control register
TADCORA: Timer A/D converter start request cycle set register A
TADCORB: Timer A/D converter start request cycle set register B
TADCOBRA: Timer A/D converter start request cycle set buffer register A
TADCOBRB: Timer A/D converter start request cycle set buffer register B
TITMRA: Timer interrupt skipping mode register
TITCR1A: Timer interrupt skipping set register 1
TITCR2A: Timer interrupt skipping set register 2
TITCNT1A: Timer interrupt skipping counter 1
TITCNT2A: Timer interrupt skipping counter 2
NFCRn: Noise filter control register n

Figure 24.1 Block Diagram of MTU (MTU0 to MTU4, MTU8)



TCR: Timer control register
TCRU: Timer control register U
TCR2U: Timer control register 2U
TCRV: Timer control register V
TCR2V: Timer control register 2V
TCRW: Timer control register W
TCR2W: Timer control register 2W
TMDR1: Timer mode register 1
TMDR2B: Timer mode register 2
TIORH: Timer I/O control register H
TIORL: Timer I/O control register L
TIORU: Timer I/O control register U
TIORV: Timer I/O control register V
TIORW: Timer I/O control register W
TIER: Timer interrupt enable register
TSR: Timer status register
TCNT: Timer counter
TCNTU: Timer counter U
TCNTV: Timer counter V
TCNTW: Timer counter W
TGRA: Timer general register A
TGRB: Timer general register B
TGRC: Timer general register C
TGRD: Timer general register D
TGRE: Timer general register E
TGRF: Timer general register F
TGRU: Timer general register U
TGRV: Timer general register V
TGRW: Timer general register W

TCNTCMPCLR: Timer compare match clear register
TSTRB: Timer start register
TSYRB: Timer synchronous register
TSYCR: Timer synchronous clear register
TCSYSTR: Timer counter synchronous start register
TRWERB: Timer read/write enable register
TOERB: Timer output master enable register
TOCR1B: Timer output control register 1
TOCR2B: Timer output control register 2
TOLBRB: Timer output level buffer register
TCDRB: Timer cycle data register
TCBRB: Timer cycle buffer register
TDDRB: Timer dead time data register
TDERB: Timer dead time enable register
TCNTSB: Timer subcounter
TBTERB: Timer buffer transfer set register
TWCRB: Timer waveform control register
TADCR: Timer A/D converter start request control register
TADCORA: Timer A/D converter start request cycle set register A
TADCORB: Timer A/D converter start request cycle set register B
TADCOBRA: Timer A/D converter start request cycle set buffer register A
TADCOBRB: Timer A/D converter start request cycle set buffer register B
TITMRB: Timer interrupt skipping mode register
TITCR1B: Timer interrupt skipping set register 1
TITCR2B: Timer interrupt skipping set register 2
TITCNT1B: Timer interrupt skipping counter 1
TITCNT2B: Timer interrupt skipping counter 2
NFCRn: Noise filter control register n

Figure 24.2 Block Diagram of MTU (MTU5 to MTU7)

Table 24.3 shows the configuration of pins for the MTU.

Table 24.3 Pin Configuration of the MTU

Channel	Pin Name	I/O	Function
MTU	MTCLKA	Input	External clock A input pin (MTU1 phase counting mode A phase input)
	MTCLKB	Input	External clock B input pin (MTU1 phase counting mode B phase input)
	MTCLKC	Input	External clock C input pin (MTU1 phase counting mode A phase input)
	MTCLKD	Input	External clock D input pin (MTU2 phase counting mode B phase input)
MTU0	MTIOC0A	I/O	MTU0 TGRA input capture input/output compare output/PWM output pin
	MTIOC0B	I/O	MTU0 TGRB input capture input/output compare output/PWM output pin
	MTIOC0C	I/O	MTU0 TGRC input capture input/output compare output/PWM output pin
	MTIOC0D	I/O	MTU0 TGRD input capture input/output compare output/PWM output pin
MTU1	MTIOC1A	I/O	MTU1 TGRA input capture input/output compare output/PWM output pin
	MTIOC1B	I/O	MTU1 TGRB input capture input/output compare output/PWM output pin
MTU2	MTIOC2A	I/O	MTU2 TGRA input capture input/output compare output/PWM output pin
	MTIOC2B	I/O	MTU2 TGRB input capture input/output compare output/PWM output pin
MTU3	MTIOC3A	I/O	MTU3 TGRA input capture input/output compare output/PWM output pin
	MTIOC3B	I/O	MTU3 TGRB input capture input/output compare output/PWM output pin
	MTIOC3C	I/O	MTU3 TGRC input capture input/output compare output/PWM output pin
	MTIOC3D	I/O	MTU3 TGRD input capture input/output compare output/PWM output pin
MTU4	MTIOC4A	I/O	MTU4 TGRA input capture input/output compare output/PWM output pin
	MTIOC4B	I/O	MTU4 TGRB input capture input/output compare output/PWM output pin
	MTIOC4C	I/O	MTU4 TGRC input capture input/output compare output/PWM output pin
	MTIOC4D	I/O	MTU4 TGRD input capture input/output compare output/PWM output pin
MTU5	MTIC5U	Input	MTU5 TGRU input capture input/external pulse input pin
	MTIC5V	Input	MTU5 TGRV input capture input/external pulse input pin
	MTIC5W	Input	MTU5 TGRW input capture input/external pulse input pin
MTU6	MTIOC6A	I/O	MTU6 TGRA input capture input/output compare output/PWM output pin
	MTIOC6B	I/O	MTU6 TGRB input capture input/output compare output/PWM output pin
	MTIOC6C	I/O	MTU6 TGRC input capture input/output compare output/PWM output pin
	MTIOC6D	I/O	MTU6 TGRD input capture input/output compare output/PWM output pin
MTU7	MTIOC7A	I/O	MTU7 TGRA input capture input/output compare output/PWM output pin
	MTIOC7B	I/O	MTU7 TGRB input capture input/output compare output/PWM output pin
	MTIOC7C	I/O	MTU7 TGRC input capture input/output compare output/PWM output pin
	MTIOC7D	I/O	MTU7 TGRD input capture input/output compare output/PWM output pin
MTU8	MTIOC8A	I/O	MTU8.TGRA input capture input/output compare output pin
	MTIOC8B	I/O	MTU8.TGRB input capture input/output compare output pin
	MTIOC8C	I/O	MTU8.TGRC input capture input/output compare output pin
	MTIOC8D	I/O	MTU8.TGRD input capture input/output compare output pin

24.2 Register Descriptions

24.2.1 Timer Control Register (TCR)

- MTU0.TCR, MTU1.TCR, MTU2.TCR, MTU3.TCR, MTU4.TCR, MTU6.TCR, MTU7.TCR, MTU8.TCR

Address(es): MTU0.TCR 000C 1300h, MTU1.TCR 000C 1380h, MTU2.TCR 000C 1400h, MTU3.TCR 000C 1200h, MTU4.TCR 000C 1201h, MTU6.TCR 000C 1A00h, MTU7.TCR 000C 1A01h, MTU8.TCR 000C 1600h

	b7	b6	b5	b4	b3	b2	b1	b0
	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC[2:0]	Time Prescaler Select	Refer to Table 24.6 to Table 24.9.	R/W
b4, b3	CKEG[1:0]	Clock Edge Select	b4 b3 0 0: Count at rising edge 0 1: Count at falling edge 1 x: Count at both edges	R/W
b7 to b5	CCLR[2:0]	Counter Clear Source Select	Refer to Table 24.4 and Table 24.5.	R/W

x: Don't care

The TCR register controls the TCNT operation for each channel in combination with the TCR2 register. The MTU has a total of 11 TCR registers, one each for MTU0 to MTU4, MTU6, MTU7, and MTU8 and three (TCRU, TCRV, and TCRW) for MTU5. TCR values should be specified only while TCNT operation is stopped.

TPSC[2:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock source. The count clock source can be selected independently for each channel. Refer to Table 24.6 to Table 24.9 for details.

CKEG[1:0] Bits (Clock Edge Select)

These bits select the clock edge, including the MTIOC1A pin. When the internal clock is counted at both edges, the count clock period is halved (e.g. PCLKA/4 at both edges = PCLKA/2 at rising edge). If phase counting mode is used on MTU1 and MTU2, the setting of these bits is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the count clock source is PCLKA/2 or slower. When PCLKA/1 or the overflow/underflow in another channel is selected for the count clock source, a value can be written to these bits but counter operation compiles with the initial value.

CCLR[2:0] Bits (Counter Clear Source Select)

These bits select the TCNT counter clearing source. Refer to Table 24.4 and Table 24.5 for details.

Table 24.4 CCLR[2:0] (MTU0, MTU3, MTU4, MTU6, MTU7, MTU8)

Channel	Bit 7 CCLR[2]	Bit 6 CCLR[1]	Bit 5 CCLR[0]	Description
MTU0 MTU3 MTU4 MTU6 MTU7 MTU8	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/ synchronous operation*1
	1	0	0	TCNT clearing disabled
	1	0	1	TCNT cleared by TGRC compare match/input capture*2
	1	1	0	TCNT cleared by TGRD compare match/input capture*2
	1	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/ synchronous operation*1

Note 1. Synchronous operation is selected by setting the TSYRA.SYNC or TSYRB.SYNC bit to 1 except for MTU8.
Note 2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority and compare match/input capture does not occur.

Table 24.5 CCLR[2:0] (MTU1 and MTU2)

Channel	Bit 7 Reserved*2	Bit 6 CCLR[1]	Bit 5 CCLR[0]	Description
MTU1 MTU2	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation*1

Note 1. Synchronous operation is selected by setting the TSYRA.SYNC and TSYRB.SYNC bits to 1.
Note 2. Bit 7 is reserved in MTU1 and MTU2. It is read as 0. The write value is ignored.

- MTU5.TCRU, MTU5.TCRV, MTU5.TCRW

Address(es): MTU5.TCRU 000C 1C84h, MTU5.TCRV 000C 1C94h, MTU5.TCRW 000C 1CA4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TPSC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TPSC[1:0]	Time Prescaler Select	Refer to Table 24.10.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TPSC[1:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock source. Refer to Table 24.10 for details.

24.2.2 Timer Control Register 2 (TCR2)

- MTU0.TCR2, MTU3.TCR2, MTU4.TCR2, MTU6.TCR2, MTU7.TCR2, MTU8.TCR2

Address(es): MTU0.TCR2 000C 1328h, MTU3.TCR2 000C 124Ch, MTU4.TCR2 000C 124Dh, MTU6.TCR2 000C 1A4Ch, MTU7.TCR2 000C 1A4Dh, MTU8.TCR2 000C 1606h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

- MTU1.TCR2, MTU2.TCR2

Address(es): MTU1.TCR2 000C 1394h, MTU2.TCR2 000C 140Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PCB[1:0]		TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC2[2:0]	Time Prescaler Select	Refer to Table 24.6 to Table 24.9.	R/W
b4, b3	PCB[1:0]	Phase Counting Mode Function Expansion Control	Functional Expansion Control for Phase Counting Modes 2, 3, and 5	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TCR2 register controls the TCNT operation for each channel in combination with the TCR register. The MTU has a total of 11 TCR2 registers, one each for MTU0 to MTU4, MTU6, MTU7, and MTU8 and three (TCRU, TCRV, and TCRW) for MTU5. TCR2 values should be specified only while TCNT operation is stopped.

TPSC2[2:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock source. The count clock source can be selected independently for each channel. Refer to Table 24.6 to Table 24.9 for details.

PCB[1:0] Bits (Phase Counting Mode Function Expansion Control)

These bits control extended functions for phase counting mode 2, 3, and 5 in MTU1 and MTU2. Refer to section 24.3.6, Phase Counting Mode.

- MTU5.TCR2U, MTU5.TCR2V, MTU5.TCR2W

Address(es): MTU5.TCR2U 000C 1C85h, MTU5.TCR2V 000C 1C95h, MTU5.TCR2W 000C 1CA5h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CKEG[1:0]		TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC2[2:0]	Time Prescaler Select	Refer to Table 24.10.	R/W
b4, b3	CKEG[1:0]	Clock Edge Select	b4 b3 0 0: Counts at the rising edge. 0 1: Counts at the falling edge. 1 x: Counts at both edges.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don't care

TPSC2[2:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock source. Refer to Table 24.10 for details.

CKEG[1:0] Bits (Clock Edge Select)

These bits select the edge of the count clock signal input from the MTIOC1A pin.

Table 24.6 TPSC[2:0], TPSC2[2:0] (MTU0)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU0	0	0	0	0	0	0	Internal clock: counts on PCLKA/1
	0	0	0	0	0	1	Internal clock: counts on PCLKA/4
	0	0	0	0	1	0	Internal clock: counts on PCLKA/16
	0	0	0	0	1	1	Internal clock: counts on PCLKA/64
	0	0	0	1	0	0	External clock: counts on MTCLKA pin input
	0	0	0	1	0	1	External clock: counts on MTCLKB pin input
	0	0	0	1	1	0	External clock: counts on MTCLKC pin input
	0	0	0	1	1	1	External clock: counts on MTCLKD pin input
	0	0	1	x	x	x	Internal clock: counts on PCLKA/2
	0	1	0	x	x	x	Internal clock: counts on PCLKA/8
	0	1	1	x	x	x	Internal clock: counts on PCLKA/32
	1	0	0	x	x	x	Internal clock: counts on PCLKA/256
	1	0	1	x	x	x	Internal clock: counts on PCLKA/1024
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	External clock: counts on MTIOC1A pin input

x: Don't care

Table 24.7 TPSC[2:0], TPSC2[2:0] (MTU1)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU1	0	0	0	0	0	0	Internal clock: counts on PCLKA/1
	0	0	0	0	0	1	Internal clock: counts on PCLKA/4
	0	0	0	0	1	0	Internal clock: counts on PCLKA/16
	0	0	0	0	1	1	Internal clock: counts on PCLKA/64
	0	0	0	1	0	0	External clock: counts on MTCLKA pin input
	0	0	0	1	0	1	External clock: counts on MTCLKB pin input
	0	0	0	1	1	0	Internal clock: counts on PCLKA/256
	0	0	0	1	1	1	Overflow/underflow of MTU2.TCNT
	0	0	1	x	x	x	Internal clock: counts on PCLKA/2
	0	1	0	x	x	x	Internal clock: counts on PCLKA/8
	0	1	1	x	x	x	Internal clock: counts on PCLKA/32
	1	0	0	x	x	x	Internal clock: counts on PCLKA/1024
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

x: Don't care

Note: This setting has no effect when MTU1 is in phase counting mode.

Table 24.8 TPSC[2:0], TPSC2[2:0] (MTU2)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU2	0	0	0	0	0	0	Internal clock: counts on PCLKA/1
	0	0	0	0	0	1	Internal clock: counts on PCLKA/4
	0	0	0	0	1	0	Internal clock: counts on PCLKA/16
	0	0	0	0	1	1	Internal clock: counts on PCLKA/64
	0	0	0	1	0	0	External clock: counts on MTCLKA pin input
	0	0	0	1	0	1	External clock: counts on MTCLKB pin input
	0	0	0	1	1	0	External clock: counts on MTCLKC pin input
	0	0	0	1	1	1	Internal clock: counts on PCLKA/1024
	0	0	1	x	x	x	Internal clock: counts on PCLKA/2
	0	1	0	x	x	x	Internal clock: counts on PCLKA/8
	0	1	1	x	x	x	Internal clock: counts on PCLKA/32
	1	0	0	x	x	x	Internal clock: counts on PCLKA/256
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

x: Don't care

Note: This setting has no effect when the MTU2 is in phase counting mode.

Table 24.9 TPSC[2:0], TPSC2[2:0] (MTU3, MTU4, MTU6, MTU7, MTU8)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU3 MTU4 MTU6 MTU7 MTU8	0	0	0	0	0	0	Internal clock: counts on PCLKA/1
	0	0	0	0	0	1	Internal clock: counts on PCLKA/4
	0	0	0	0	1	0	Internal clock: counts on PCLKA/16
	0	0	0	0	1	1	Internal clock: counts on PCLKA/64
	0	0	0	1	0	0	Internal clock: counts on PCLKA/256
	0	0	0	1	0	1	Internal clock: counts on PCLKA/1024
	0	0	0	1	1	0	External clock: counts on MTCLKA pin input
	0	0	0	1	1	1	External clock: counts on MTCLKB pin input
	0	0	1	x	x	x	Internal clock: counts on PCLKA/2
	0	1	0	x	x	x	Internal clock: counts on PCLKA/8
	0	1	1	x	x	x	Internal clock: counts on PCLKA/32
	1	0	0	x	x	x	Setting prohibited
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

x: Don't care

Table 24.10 TPSC[1:0], TPSC2[2:0] (MTU5)

	TCR2 register			TCR register		
	Bit 2	Bit 1	Bit 0	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[1]	TPSC[0]	Description
MTU5	0	0	0	0	0	Internal clock: counts on PCLKA/1
	0	0	0	0	1	Internal clock: counts on PCLKA/4
	0	0	0	1	0	Internal clock: counts on PCLKA/16
	0	0	0	1	1	Internal clock: counts on PCLKA/64
	0	0	1	x	x	Internal clock: counts on PCLKA/2
	0	1	0	x	x	Internal clock: counts on PCLKA/8
	0	1	1	x	x	Internal clock: counts on PCLKA/32
	1	0	0	x	x	Internal clock: counts on PCLKA/256
	1	0	1	x	x	Internal clock: counts on PCLKA/1024
	1	1	0	x	x	Setting prohibited
	1	1	1	x	x	Internal clock: counts on MTIOC1A pin input

x: Don't care

24.2.3 Timer Mode Register 1 (TMDR1)

- MTU0.TMDR1

Address(es): MTU0.TMDR1 000C 1301h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	BFE	BFB	BFA	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

- MTU1.TMDR1, MTU2.TMDR1

Address(es): MTU1.TMDR1 000C 1381h, MTU2.TMDR1 000C 1401h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

- MTU3.TMDR1, MTU4.TMDR1, MTU6.TMDR1, MTU7.TMDR1, MTU8.TMDR1

Address(es): MTU3.TMDR1 000C 1202h, MTU4.TMDR1 000C 1203h, MTU6.TMDR1 000C 1A02h, MTU7.TMDR1 000C 1A03h, MTU8.TMDR1 000C 1601h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	BFB	BFA	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MD[3:0]	Mode Select	These bits specify the timer operating mode. Refer to Table 24.11 for details.	R/W
b4	BFA	Buffer Operation A	0: TGRA and TGRC operate normally 1: TGRA and TGRC used together for buffer operation	R/W
b5	BFB	Buffer Operation B	0: TGRB and TGRD operate normally 1: TGRB and TGRD used together for buffer operation	R/W
b6	BFE	Buffer Operation E	0: MTU0.TGRE and MTU0.TGRF operate normally 1: MTU0.TGRE and MTU0.TGRF used together for buffer operation	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The TMDR1 register specifies the operating mode of each channel. The MTU has a total of eight TMDR1 registers, one each for MTU0 to MTU4, MTU6, MTU7, and MTU8. TMDR1 register values should be specified only while TCNT operation is stopped.

Table 24.11 Operating Mode Setting by MD[3:0] Bits (MTU0 to MTU4 and MTU6 to MTU8)

Bit 3	Bit 2	Bit 1	Bit 0		MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU6	MTU7	MTU8
MD[3]	MD[2]	MD[1]	MD[0]	Description									
0	0	0	0	Normal mode	✓	✓	✓		✓	✓	✓	✓	✓
0	0	0	1	Setting prohibited									
0	0	1	0	PWM mode 1	✓	✓	✓		✓	✓	✓	✓	
0	0	1	1	PWM mode 2	✓	✓	✓						
0	1	0	0	Phase counting mode 1		✓	✓	✓					
0	1	0	1	Phase counting mode 2		✓	✓	✓					
0	1	1	0	Phase counting mode 3		✓	✓	✓					
0	1	1	1	Phase counting mode 4		✓	✓	✓					
1	0	0	0	Reset-synchronized PWM mode*1					✓		✓		
1	0	0	1	Phase counting mode 5		✓	✓	✓					
1	0	1	x	Setting prohibited									
1	1	0	0	Setting prohibited									
1	1	0	1	Complementary PWM mode 1 (transfer at crest)*1					✓		✓		
1	1	1	0	Complementary PWM mode 2 (transfer at trough)*1					✓		✓		
1	1	1	1	Complementary PWM mode 3 (transfer at crest and trough)*1					✓		✓		

x: Don't care

Note: Only set the corresponding operating mode listed above for each channel.

Note 1. Reset-synchronized PWM mode and complementary PWM mode can only be set for MTU3 and MTU6. When MTU3 or MTU6 is set to reset-synchronized PWM mode or complementary PWM mode, the MTU4 or MTU7 settings become ineffective and automatically conform to the MTU3 or MTU6 setting, respectively. MTU4 and MTU7 should be set to the initial values (normal mode).

BFA Bit (Buffer Operation A)

This bit specifies whether to operate TGRA in the normal way or to use TGRA and TGRC together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare does not take place in modes other than complementary PWM mode, but compare match with TGRC occurs in complementary PWM mode.

If a compare match occurs on MTU4 in the Tb interval in complementary PWM mode, the TGIEC bit in timer interrupt enable register (MTU4.TIER) should be set to 0.

In reset-synchronized PWM mode or complementary PWM mode, buffer operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the settings in the BFA bit of MTU3.TMDR1 (MTU6.TMDR1). The BFA bit of MTU4.TMDR1 (MTU7.TMDR1) should be set to 0.

In MTU1 and MTU2, which have no TGRC, this bit is reserved. It is read as 0. The write value should be 0.

Refer to Figure 24.50 for an illustration of the Tb interval in complementary PWM mode.

BFB Bit (Buffer Operation B)

This bit specifies whether to operate TGRB in the normal way or to use TGRB and TGRD together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare does not take place in modes other than complementary PWM mode, but compare match with TGRD occurs in complementary PWM mode.

If a compare match occurs on MTU4 in the Tb interval in complementary PWM mode, the TGIED bit in timer interrupt enable register (MTU4.TIER) should be set to 0.

In reset-synchronized PWM mode or complementary PWM mode, buffer operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the settings in the BFB bit of MTU3.TMDR1 (MTU6.TMDR1). The BFB bit of MTU4.TMDR1

(MTU7.TMDR1) should be set to 0.
In MTU1 and MTU2, which have no TGRD, this bit is reserved. It is read as 0. The write value should be 0.
Refer to Figure 24.50 for an illustration of the Tb interval in complementary PWM mode.

BFE Bit (Buffer Operation E)

This bit specifies whether to operate MTU0.TGRE and MTU0.TGRF in the normal way or to use them together for buffer operation. Compare match with TGRF occurs even when TGRF is used as a buffer register.
In MTU0 to MTU4, MTU6, MTU7, and MTU8, this bit is reserved. It is read as 0. The write value should be 0.

24.2.4 Timer Mode Registers 2 (TMDR2A, TMDR2B)

Address(es): MTU.TMDR2A 000C 1270h, MTU.TMDR2B 000C 1A70h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DRS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DRS	Double Buffer Select	0: Double buffer function is disabled 1: Double buffer function is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TMDR2A and TMDR2B specify the double buffer function in complementary PWM mode 3 (transfer at the crest and trough of the counter value). The MTU has two TMDR2 registers, and one each for MTU3 (TMDR2A) and MTU6 (TMDR2B). TMDR2A and TMDR2B values should be specified only while TCNT operation is stopped.

DRS Bit (Double Buffer Select)

This bit enables or disables the double buffer function in complementary PWM mode.

24.2.5 Timer Mode Register 3 (TMDR3)

Address(es): MTU1.TMDR3 000C 1391h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PHCKSEL	LWA
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	LWA	MTU1/MTU2 Combination Longword Access Control	0: 16-bit access is enabled. 1: 32-bit access is enabled.	R/W
b1	PHCKSEL	External Input Phase Clock Select	0: MTCLKA and MTCLKB are selected for the external phase clock. 1: MTCLKC and MTCLKD are selected for the external phase clock.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TMDR3 register controls longword access to a 32-bit register or counter in a combination of MTU1 and MTU2. There is only one TMDR3 register in MTU1. The counter (TCNTLW), general register A (TGRALW), and general register B (TGRBLW) of MTU1 and MTU2 are accessed in the combinations listed in Table 24.12.

LWA Bit (MTU1/MTU2 Combination Longword Access Control)

This bit selects a 32-bit access in a combination of MTU1 and MTU2. When LWA is set to 0, MTU1.TCNTLW, MTU1.TGRALW, and MTU1.TGRBLW cannot be accessed. This bit is read as 0. When LWA is set to 1, MTU1.TCNT, MTU2.TCNT, MTU1.TGRA, MTU2.TGRA, MTU1.TGRB, and MTU2.TGRB cannot be accessed. This bit is read as 0. MTU1 and MTU2 operate together while LWA is 1, so the settings of the timer control registers (TCR and TCR2) and timer I/O control register (TIOR) in MTU1 are enabled, and the settings of the control registers in MTU2 are disabled. Furthermore, MTU2 input capture and compare match are also disabled, which in turn disables any linked operation with the ELC.

When changing the value of the LWA bit, initialize the counters and general registers for MTU1 and MTU2 in advance.

PHCKSEL Bit (External Input Phase Clock Select)

When the MTU1 and MTU2 registers are combined for 32-bit phase counting mode or MTU2 phase counting mode, this bit selects either the A- or B-phase signal from the external clock. Refer to Table 24.66, Clock Input Pins in Phase Counting Mode for details.

Table 24.12 Setting and Combination of the TMDR3 Register

	TMDR3.LWA = 0		TMDR3.LWA = 1	
Register	**Symbol**	**Access mode**	**Symbol**	**Access mode**
Counter in MTU1*1	MTU1.TCNT	Word	MTU1.TCNTLW	Longword
Counter in MTU2	MTU2.TCNT	Word		
General register A in MTU1	MTU1.TGRA	Word	MTU1.TGRALW	Longword
General register A in MTU2	MTU2.TGRA	Word		
General register B in MTU1	MTU1.TGRB	Word	MTU1.TGRBLW	Longword
General register B in MTU2	MTU2.TGRB	Word		

Note 1. When the LWA bit is set to 1, setting the count clock for MTU1 as MTU2.TCNT overflow/underflow is not required.

24.2.6 Timer I/O Control Register (TIOR)

- MTU0.TIORH, MTU1.TIOR, MTU2.TIOR, MTU3.TIORH, MTU4.TIORH, MTU6.TIORH, MTU7.TIORH, MTU8.TIORH

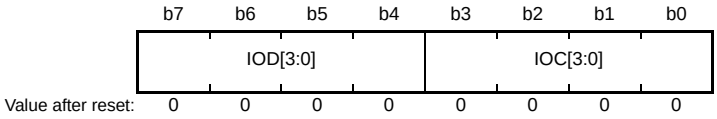
Address(es): MTU0.TIORH 000C 1302h, MTU1.TIOR 000C 1382h, MTU2.TIOR 000C 1402h, MTU3.TIORH 000C 1204h, MTU4.TIORH 000C 1206h, MTU6.TIORH 000C 1A04h, MTU7.TIORH 000C 1A06h, MTU8.TIORH 000C 1602h

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOA[3:0]	I/O Control A*1	Refer to the following tables. MTU0.TIORH: Table 24.27 MTU1.TIOR: Table 24.29 MTU2.TIOR: Table 24.30 MTU3.TIORH: Table 24.31 MTU4.TIORH: Table 24.33 MTU6.TIORH: Table 24.35 MTU7.TIORH: Table 24.37 MTU8.TIORH: Table 24.39	R/W
b7 to b4	IOB[3:0]	I/O Control B*1	Refer to the following tables. MTU0.TIORH: Table 24.13 MTU1.TIOR: Table 24.15 MTU2.TIOR: Table 24.16 MTU3.TIORH: Table 24.17 MTU4.TIORH: Table 24.19 MTU6.TIORH: Table 24.21 MTU7.TIORH: Table 24.23 MTU8.TIORH: Table 24.25	R/W

Note 1. When the value of IOn[3:0] (n = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

- MTU0.TIORL, MTU3.TIORL, MTU4.TIORL, MTU6.TIORL, MTU7.TIORL, MTU8.TIORL

Address(es): MTU0.TIORL 000C 1303h, MTU3.TIORL 000C 1205h, MTU4.TIORL 000C 1207h, MTU6.TIORL 000C 1A05h, MTU7.TIORL 000C 1A07h, MTU8.TIORL 000C 1603h

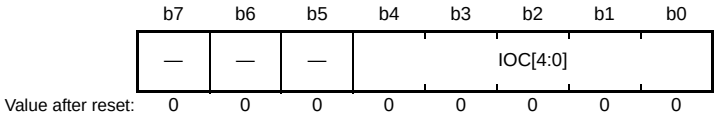


Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOC[3:0]	I/O Control C*1	Refer to the following tables. MTU0.TIORL: Table 24.28 MTU3.TIORL: Table 24.32 MTU4.TIORL: Table 24.34 MTU6.TIORL: Table 24.36 MTU7.TIORL: Table 24.38 MTU8.TIORL: Table 24.40	R/W
b7 to b4	IOD[3:0]	I/O Control D*1	Refer to the following tables. MTU0.TIORL: Table 24.14 MTU3.TIORL: Table 24.18 MTU4.TIORL: Table 24.20 MTU6.TIORL: Table 24.22 MTU7.TIORL: Table 24.24 MTU8.TIORL: Table 24.26	R/W

Note 1. When the value of IOn[3:0] (n = C, D) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

- MTU5.TIORU, MTU5.TIORV, MTU5.TIORW

Address(es): MTU5.TIORU 000C 1C86h, MTU5.TIORV 000C 1C96h, MTU5.TIORW 000C 1CA6h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	IOC[4:0]	I/O Control C	Refer to the following table. MTU5.TIORU, MTU5.TIORV, MTU5.TIORW: Table 24.41	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TIOR register controls the TGR register. The MTU has a total of 17 TIOR registers, two each for MTU0, MTU3, MTU4, MTU6, MTU7, and MTU8, one each for MTU1 and MTU2, and three (MTU5.TIORU/TIORV/TIORW) for MTU5. The TIOR register should be set when the TMDR register setting is normal mode, PWM mode, or phase counting mode.

Note that TIOR is affected by the TMDR1 setting.

The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTRA and the CST bit in TSTRB are cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

Table 24.13 TIORH (MTU0)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	MTU0.TGRB Function	MTIOC0B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*1

x: Don't care

Note 1. When PCLKA/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKA/1 as the count clock for MTU1.

Table 24.14 TIORL (MTU0)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	MTU0.TGRD Function	MTIOC0D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register*1	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*2

x: Don't care

Note 1. When the MTU0.TMDR1.BFB bit is set to 1 and MTU0.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 2. When PCLKA/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKA/1 as the count clock for MTU1.

Table 24.15 TIOR (MTU1)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	MTU1.TGRB Function	MTIOC1B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	0	0		Input capture at occurrence of compare match or input capture in the MTU0.TGRC register
1	1	1	x		Input capture at occurrence of compare match in the MTU8.TGRC register

x: Don't care

Table 24.16 TIOR (MTU2)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	MTU2.TGRB Function	MTIOC2B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.17 TIORH (MTU3)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	MTU3.TGRB Function	MTIOC3B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.18 TIORL (MTU3)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	MTU3.TGRD Function	MTIOC3D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU3.TMDR1.BFB bit is set to 1 and MTU3.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.19 TIORH (MTU4)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	MTU4.TGRB Function	MTIOC4B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.20 TIORL (MTU4)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	MTU4.TGRD Function	MTIOC4D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU4.TMDR1.BFB bit is set to 1 and MTU4.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.21 TIORH (MTU6)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	MTU6.TGRB Function	MTIOC6B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.22 TIORL (MTU6)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	MTU6.TGRD Function	MTIOC6D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU6.TMDR1.BFB bit is set to 1 and MTU6.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.23 TIORH (MTU7)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	MTU7.TGRB Function	MTIOC7B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.24 TIORL (MTU7)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	MTU7.TGRD Function	MTIOC7D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU7.TMDR1.BFB bit is set to 1 and MTU7.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.25 TIORH (MTU8)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	MTU8.TGRB Function	MTIOC8B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).

x: Don't care

Table 24.26 TIORL (MTU8)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	MTU8.TGRD Function	MTIOC8D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU8.TMDR1.BFB bit is set to 1 and the MTU8.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.27 TIORH (MTU0)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	MTU0.TGRA Function	MTIOC0A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	0	0		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*1
1	1	1	x		Input capture on generation of compare match with MTU8.TGRC

x: Don't care

Note 1. When PCLKA/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKA/1 as the count clock for MTU1.

Table 24.28 TIORL (MTU0)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	MTU0.TGRC Function	MTIOC0C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register*1	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*2

x: Don't care

Note 1. When the MTU0.TMDR1.BFB bit is set to 1 and MTU0.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 2. When PCLKA/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKA/1 as the count clock for MTU1.

Table 24.29 TIOR (MTU1)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	MTU1.TGRA Function	MTIOC1A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Input capture at generation of MTU0.TGRA compare match/input capture.

x: Don't care

Table 24.30 TIOR (MTU2)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	MTU2.TGRA Function	MTIOC2A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.31 TIORH (MTU3)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	MTU3.TGRA Function	MTIOC3A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.32 TIORL (MTU3)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	MTU3.TGRC Function	MTIOC3C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU3.TMDR1.BFB bit is set to 1 and MTU3.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.33 TIORH (MTU4)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	MTU4.TGRA Function	MTIOC4A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.34 TIORL (MTU4)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	MTU4.TGRC Function	MTIOC4C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU4.TMDR1.BFB bit is set to 1 and MTU4.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.35 TIORH (MTU6)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	MTU6.TGRA Function	MTIOC6A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.36 TIORL (MTU6)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	MTU6.TGRC Function	MTIOC6C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU6.TMDR1.BFB bit is set to 1 and MTU6.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.37 TIORH (MTU7)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	MTU7.TGRA Function	MTIOC7A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.38 TIORL (MTU7)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	MTU7.TGRC Function	MTIOC7C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU7.TMDR1.BFB bit is set to 1 and MTU7.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.39 TIORH (MTU8)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	MTU8.TGRA Function	MTIOC8A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.40 TIORL (MTU8)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	MTU8.TGRC Function	MTIOC8C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU8.TMDR1.BFA bit is set to 1 and the MTU8.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.41 TIORU, TIORV, and TIORW (MTU5)

Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[4]	IOC[3]	IOC[2]	IOC[1]	IOC[0]	MTU5.TGRU, MTU5.TGRV, MTU5.TGRW Function	MTIC5U, MTIC5V, MTIC5W Pin Function
0	0	0	0	0	Output compare register	No function
0	0	0	0	1		Setting prohibited
0	0	0	1	x		Setting prohibited
0	0	1	x	x		Setting prohibited
0	1	x	x	x		Setting prohibited
1	0	0	0	0	Input capture register*1	Setting prohibited
1	0	0	0	1		Input capture at rising edge.
1	0	0	1	0		Input capture at falling edge.
1	0	0	1	1		Input capture at both edges.
1	0	1	x	x		Input capture on generation of compare match with MTU8.TGRC
1	1	0	0	0		Setting prohibited
1	1	0	0	1		Measurement of low pulse width of external input signal. Capture at trough in complementary PWM mode.
1	1	0	1	0		Measurement of low pulse width of external input signal. Capture at crest of complementary PWM mode.
1	1	0	1	1		Measurement of low pulse width of external input signal. Capture at crest and trough of complementary PWM mode.
1	1	1	0	0		Setting prohibited
1	1	1	0	1		Measurement of high pulse width of external input signal. Capture at trough in complementary PWM mode.
1	1	1	1	0		Measurement of high pulse width of external input signal. Capture at crest of complementary PWM mode.
1	1	1	1	1		Measurement of high pulse width of external input signal. Capture at crest and trough of complementary PWM mode.

x: Don't care

Note 1. Set the IOC[4:0] bits to 19h, 1Ah, 1Bh, 1Dh, 1Eh, or 1Fh only when using external pulse width measurement or only when using dead time compensation linked with MTU6 and MTU7. For details, refer to section 24.3.11, External Pulse Width Measurement and section 24.3.12, Dead Time Compensation.

24.2.7 Timer Compare Match Clear Register (TCNTCMPCLR)

Address(es): MTU5.TCNTCMPCLR 000C 1CB6h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPCLR5W	TCNT Compare Clear 5W	0: Disables MTU5.TCNTW to be cleared to 0000h at MTU5.TCNTW and MTU5.TGRW compare match or input capture 1: Enables MTU5.TCNTW to be cleared to 0000h at MTU5.TCNTW and MTU5.TGRW compare match or input capture	R/W
b1	CMPCLR5V	TCNT Compare Clear 5V	0: Disables MTU5.TCNTV to be cleared to 0000h at MTU5.TCNTV and MTU5.TGRV compare match or input capture 1: Enables MTU5.TCNTV to be cleared to 0000h at MTU5.TCNTV and MTU5.TGRV compare match or input capture	R/W
b2	CMPCLR5U	TCNT Compare Clear 5U	0: Disables MTU5.TCNTU to be cleared to 0000h at MTU5.TCNTU and MTU5.TGRU compare match or input capture 1: Enables MTU5.TCNTU to be cleared to 0000h at MTU5.TCNTU and MTU5.TGRU compare match or input capture	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TCNTCMPCLR specifies requests to clear MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW. The MTU has one TCNTCMPCLR (on MTU5).

24.2.8 Timer Interrupt Enable Register (TIER)

- MTU1.TIER, MTU2.TIER

Address(es): MTU1.TIER 000C 1384h, MTU2.TIER 000C 1404h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

- MTU0.TIER, MTU3.TIER, MTU6.TIER

Address(es): MTU0.TIER 000C 1304h, MTU3.TIER 000C 1208h, MTU6.TIER 000C 1A08h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

- MTU4.TIER, MTU7.TIER

Address(es): MTU4.TIER 000C 1209h, MTU7.TIER 000C 1A09h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

- MTU8.TIER

Address(es): MTU8.TIER 000C 1604h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIEA	TGR Interrupt Enable A	0: Interrupt requests (TGIA) disabled 1: Interrupt requests (TGIA) enabled	R/W
b1	TGIEB	TGR Interrupt Enable B	0: Interrupt requests (TGIB) disabled 1: Interrupt requests (TGIB) enabled	R/W
b2	TGIEC	TGR Interrupt Enable C	0: Interrupt requests (TGIC) disabled 1: Interrupt requests (TGIC) enabled	R/W
b3	TGIED	TGR Interrupt Enable D	0: Interrupt requests (TGID) disabled 1: Interrupt requests (TGID) enabled	R/W
b4	TCIEV	Overflow Interrupt Enable	0: Interrupt requests (TCIV) disabled 1: Interrupt requests (TCIV) enabled	R/W
b5	TCIEU	Underflow Interrupt Enable	0: Interrupt requests (TCIU) disabled 1: Interrupt requests (TCIU) enabled	R/W
b6	TTGE2	A/D Converter Start Request Enable 2	0: A/D converter start request generation by MTUn.TCNT underflow (trough) disabled 1: A/D converter start request generation by MTUn.TCNT underflow (trough) enabled	R/W
b7	TTGE	A/D Converter Start Request Enable	0: A/D converter start request generation disabled 1: A/D converter start request generation enabled	R/W

n = 4, 7

The TIER register enables or disables interrupt requests from each channel. The MTU has a total of ten TIER registers, two for MTU0 and one each for MTU1 to MTU8.

TGIEA and TGIEB Bits (TGR Interrupt Enable A and B)

Each bit enables or disables interrupt requests (TGIn) (n = A, B).

TGIEC and TGIED Bits (TGR Interrupt Enable C and D)

Each bit enables or disables an interrupt request (TGIn) (n = C, D).
In MTU1 and MTU2, these bits are reserved. They are read as 0. The write value should be 0.

TCIEV Bit (Overflow Interrupt Enable)

This bit enables or disables interrupt requests (TCIV).

TCIEU Bit (Underflow Interrupt Enable)

This bit enables or disables interrupt requests (TCIU).
In MTU0, MTU3, MTU4, MTU6, MTU7, and MTU8, this bit is reserved. It is read as 0. The write value should be 0.

TTGE2 Bit (A/D Converter Start Request Enable 2)

This bit enables or disables generation of A/D converter start requests by MTUn.TCNT underflow (trough) in complementary PWM mode (n = 4, 7).
In MTU0 to MTU3, MTU6, and MTU8, this bit is reserved. It is read as 0. The write value should be 0.

TTGE Bit (A/D Converter Start Request Enable)

This bit enables or disables generation of A/D converter start requests by TGRA input capture/compare match.
MTU8 is a reserved bit. It is read as 0. The write value should be 0.

- MTU0.TIER2

Address(es): MTU0.TIER2 000C 1324h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE2	—	—	—	—	—	TGIEF	TGIEE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIEE	TGR Interrupt Enable E	0: Interrupt requests (TGIE) disabled 1: Interrupt requests (TGIE) enabled	R/W
b1	TGIEF	TGR Interrupt Enable F	0: Interrupt requests (TGIF) disabled 1: Interrupt requests (TGIF) enabled	R/W
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TTGE2	A/D Converter Start Request Enable 2	0: A/D converter start request generation by compare match between MTU0.TCNT and MTU0.TGRE disabled 1: A/D converter start request generation by compare match between MTU0.TCNT and MTU0.TGRE enabled	R/W

TGIEE and TGIEF Bits (TGR Interrupt Enable E and F)

Each bit enables or disables interrupt requests by compare match between MTU0.TCNT and MTU0.TGRn (n = E, F).

TTGE2 Bit (A/D Converter Start Request Enable 2)

Each bit enables or disables A/D converter start requests by compare match between MTU0.TCNT and MTU0.TGRE.

- MTU5.TIER

Address(es): MTU5.TIER 000C 1CB2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	TGIE5W	TGR Interrupt Enable 5W	0: Interrupt requests TGIW5 disabled 1: Interrupt requests TGIW5 enabled	R/W
b1	TGIE5V	TGR Interrupt Enable 5V	0: Interrupt requests TGIV5 disabled 1: Interrupt requests TGIV5 enabled	R/W
b2	TGIE5U	TGR Interrupt Enable 5U	0: Interrupt requests TGIU5 disabled 1: Interrupt requests TGIU5 enabled	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TGIE5n Bits (TGR Interrupt Enable 5n)

Each bit enables or disables interrupt requests (TGIn5) (n = U, V, W).

24.2.9 Timer Status Register (TSR)

- MTU1.TSR, MTU2.TSR

Address(es): MTU1.TSR 000C 1385h, MTU2.TSR 000C 1405h

	b7	b6	b5	b4	b3	b2	b1	b0
	TCFD	—	—	—	—	—	—	—
Value after reset:	1	1	0	0	0	0	0	0

- MTU3.TSR, MTU4.TSR, MTU6.TSR, MTU7.TSR

Address(es): MTU3.TSR 000C 122Ch, MTU4.TSR 000C 122Dh, MTU6.TSR 000C 1A2Ch, MTU7.TSR 000C 1A2Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	TCFD	—	—	—	—	—	—	—
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	The read value is undefined.The write value should be 1.	R/W
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	TCFD	Count Direction Flag	0: TCNT counts down 1: TCNT counts up	R

TSR indicates the states of each of the channels. The MTU has a total of six TSR registers, one each for MTU1 to MTU4, MTU6, and MTU7.

TCFD Flag (Count Direction Flag)

Status flag that indicates the direction in which TCNT is counting in MTU1 to MTU4, MTU6, and MTU7.

24.2.10 Timer Buffer Operation Transfer Mode Register (TBTM)

- MTU0.TBTM

Address(es): MTU0.TBTM 000C 1326h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TTSE	TTSB	TTSA
Value after reset:	0	0	0	0	0	0	0	0

- MTU3.TBTM, MTU4.TBTM, MTU6.TBTM, MTU7.TBTM

Address(es): MTU3.TBTM 000C 1238h, MTU4.TBTM 000C 1239h, MTU6.TBTM 000C 1A38h, MTU7.TBTM 000C 1A39h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TTSB	TTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TTSA	Timing Select A	0: When compare match A occurs in each channel, data is transferred from TGRC to TGRA 1: When TCNT is cleared in each channel, data is transferred from TGRC to TGRA	R/W
b1	TTSB	Timing Select B	0: When compare match B occurs in each channel, data is transferred from TGRD to TGRB 1: When TCNT is cleared in each channel, data is transferred from TGRD to TGRB	R/W
b2	TTSE	Timing Select E	0: When compare match E occurs in MTU0, data is transferred from MTU0.TGRF to MTU0.TGRE 1: When MTU0.TCNT is cleared, data is transferred from MTU0.TGRF to MTU0.TGRE	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TBTM specifies the timing for transferring data from the buffer register to the timer general register in PWM mode. The MTU has a total of five TBTM registers, one each for MTU0, MTU3, MTU4, MTU6, and MTU7.

TTSA Bit (Timing Select A)

This bit specifies the timing for transferring data from TGRC to TGRA in each channel when they are used together for buffer operation. When a channel is not set to PWM mode, do not set the TTSA bit in the channel to 1.

TTSB Bit (Timing Select B)

This bit specifies the timing for transferring data from TGRD to TGRB in each channel when they are used together for buffer operation. When a channel is not set to PWM mode, do not set the TTSB bit in the channel to 1.

TTSE Bit (Timing Select E)

This bit specifies the timing for transferring data from MTU0.TGRF to MTU0.TGRE when they are used together for buffer operation.
In MTU3, MTU4, MTU6, and MTU7, this bit is reserved. It is read as 0 and the write value should be 0. When a channel is not set to PWM mode, do not set the TTSE bit in the channel to 1.

24.2.11 Timer Input Capture Control Register (TICCR)

Address(es): MTU1.TICCR 000C 1390h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	I2BE	I2AE	I1BE	I1AE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	I1AE	Input Capture Enable	0: Does not include the MTIOC1A pin in the MTU2.TGRA input capture conditions 1: Includes the MTIOC1A pin in the MTU2.TGRA input capture conditions	R/W
b1	I1BE	Input Capture Enable	0: Does not include the MTIOC1B pin in the MTU2.TGRB input capture conditions 1: Includes the MTIOC1B pin in the MTU2.TGRB input capture conditions	R/W
b2	I2AE	Input Capture Enable	0: Does not include the MTIOC2A pin in the MTU1.TGRA input capture conditions 1: Includes the MTIOC2A pin in the MTU1.TGRA input capture conditions	R/W
b3	I2BE	Input Capture Enable	0: Does not include the MTIOC2B pin in the MTU1.TGRB input capture conditions 1: Includes the MTIOC2B pin in the MTU1.TGRB input capture conditions	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TICCR specifies input capture conditions when MTU1.TCNT and MTU2.TCNT are cascaded. The MTU has one TICCR for MTU1.

24.2.12 Timer Synchronous Clear Register (TSYCR)

Address(es): MTU6.TSYCR 000C 1A50h

	b7	b6	b5	b4	b3	b2	b1	b0
	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CE2B	Clear Enable 2B	0: Disables counter clearing by the MTU2.TGIB2 interrupt generation timing. 1: Enables counter clearing by the MTU2.TGIB2 interrupt generation timing.	R/W
b1	CE2A	Clear Enable 2A	0: Disables counter clearing by the MTU2.TGIA2 interrupt generation timing*1. 1: Enables counter clearing by the MTU2.TGIA2 interrupt generation timing*1.	R/W
b2	CE1B	Clear Enable 1B	0: Disables counter clearing by the MTU1.TGIB1 interrupt generation timing*1. 1: Enables counter clearing by the MTU1.TGIB1 interrupt generation timing*1.	R/W
b3	CE1A	Clear Enable 1A	0: Disables counter clearing by the MTU1.TGIA1 interrupt generation timing*1. 1: Enables counter clearing by the MTU1.TGIA1 interrupt generation timing*1.	R/W
b4	CE0D	Clear Enable 0D	0: Disables counter clearing by the MTU0.TGID0 interrupt generation timing*1. 1: Enables counter clearing by the MTU0.TGID0 interrupt generation timing*1.	R/W
b5	CE0C	Clear Enable 0C	0: Disables counter clearing by the MTU0.TGIC0 interrupt generation timing*1. 1: Enables counter clearing by the MTU0.TGIC0 interrupt generation timing*1.	R/W
b6	CE0B	Clear Enable 0B	0: Disables counter clearing by the MTU0.TGIB0 interrupt generation timing*1. 1: Enables counter clearing by the MTU0.TGIB0 interrupt generation timing*1.	R/W
b7	CE0A	Clear Enable 0A	0: Disables counter clearing by the MTU0.TGIA0 interrupt generation timing*1. 1: Enables counter clearing by the MTU0.TGIA0 interrupt generation timing*1.	R/W

Note 1. This does not depend on the TIERn.TGIEm bit setting. (n = 0, 1, 2; m = A, B, C, D)

TSYCR specifies synchronous clear conditions for MTU6.TCNT and MTU7.TCNT. The MTU has one TSYCR for MTU1.

CEnm Bits (Clear Enable nm; n = 0, 1, 2; m = A, B, C, D)

These bits enable or disable counter clearing by the MTUn.TGImn interrupt generation timing.

24.2.13 Timer Counter (TCNT)

- MTU0.TCNT to MTU7.TCNT

Address(es): MTU0.TCNT 000C 1306h, MTU1.TCNT 000C 1386h, MTU2.TCNT 000C 1406h, MTU3.TCNT 000C 1210h, MTU4.TCNT 000C 1212h, MTU5.TCNTU 000C 1C80h, MTU5.TCNTV 000C 1C90h, MTU5.TCNTW 000C 1CA0h, MTU6.TCNT 000C 1A10h, MTU7.TCNT 000C 1A12h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: TCNT must not be accessed in 8 bits; it should be accessed in 16 bits.

- MTU8.TCNT

Address(es): MTU8.TCNT 000C 1608h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: MTU8.TCNT must not be accessed in 8 or 16 bits; it should be accessed in 32 bits.

MTU0.TCNT to MTU7.TCNT are 16-bit readable/writable counters and MTU8.TCNT is a 32-bit readable/writable counter. The MTU has a total of 11 TCNT counters, one each for MTU0 to MTU4 and MTU6 to MTU8 and three (MTU5.TCNTU, TCNTV, and TCNTW) for MTU5. The TCNT counters in MTU0 to MTU4, MTU6, and MTU7 are initialized to 0000h by a reset, and the MTU8.TCNT counter is initialized to 00000000h by a reset. MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW are initialized to 0000h by a reset.

In MTU0 to MTU4, MTU6, and MTU7, the TCNT counters must not be accessed in 8-bit units; they should be accessed in 16- or 32-bit units. The MTU8.TCNT counter must not be accessed in 8- or 16-bit units; it should be accessed in 32-bit units.

The MTU1.TCNT and MTU2.TCNT counters are read as 0000h when TMDR3.LWA is 1. Refer to section 24.2.5, Timer Mode Register 3 (TMDR3) for details.

24.2.14 Timer Longword Counter (TCNTLW)

Address(es): MTU1.TCNTLW 000C 13A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: TCNTLW must not be accessed in 8 or 16 bits; it should be accessed in 32 bits.

The TCNTLW counter is a 32-bit readable/writable counter. Only one counter of this type is provided, and is formed by combining MTU1.TCNT and MTU2.TCNT. Such operation is only effective when TMDR3.LWA is 1. The TCNTLW counter is initialized to 0000 0000h by a reset. This counter is read as 0000 0000h when TMDR3.LWA is 0. Refer to section 24.2.5, Timer Mode Register 3 (TMDR3) for details. This register can only be used in 32-bit phase counting mode.

24.2.15 Timer General Register (TGR)

- MTU0.TGR to MTU7.TGR

Address(es): MTU0.TGRA 000C 1308h, MTU0.TGRB 000C 130Ah, MTU0.TGRC 000C 130Ch, MTU0.TGRD 000C 130Eh, MTU0.TGRE 000C 1320h, MTU0.TGRF 000C 1322h,
MTU1.TGRA 000C 1388h, MTU1.TGRB 000C 138Ah,
MTU2.TGRA 000C 1408h, MTU2.TGRB 000C 140Ah,
MTU3.TGRA 000C 1218h, MTU3.TGRB 000C 121Ah, MTU3.TGRC 000C 1224h, MTU3.TGRD 000C 1226h, MTU3.TGRE 000C 1272h
MTU4.TGRA 000C 121Ch, MTU4.TGRB 000C 121Eh, MTU4.TGRC 000C 1228h, MTU4.TGRD 000C 122Ah, MTU4.TGRE 000C 1274h, MTU4.TGRF 000C 1276h,
MTU5.TGRU 000C 1C82h, MTU5.TGRV 000C 1C92h, MTU5.TGRW 000C 1CA2h,
MTU6.TGRA 000C 1A18h, MTU6.TGRB 000C 1A1Ah, MTU6.TGRC 000C 1A24h, MTU6.TGRD 000C 1A26h, MTU6.TGRE 000C 1A72h,
MTU7.TGRA 000C 1A1Ch, MTU7.TGRB 000C 1A1Eh, MTU7.TGRC 000C 1A28h, MTU7.TGRD 000C 1A2Ah, MTU7.TGRE 000C 1A74h,
MTU7.TGRF 000C 1A76h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: TGR must not be accessed in 8 bits; it should be accessed in 16 bits. The initial value of TGR is FFFFh.

- MTU8.TGR

Address(es): MTU8.TGRA 000C 160Ch, MTU8.TGRB 000C 1610h, MTU8.TGRC 000C 1614h, MTU8.TGRD 000C 1618h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: MTU8.TGR must not be accessed in 8 or 16 bits; it should be accessed in 32 bits.

The MTU0.TGR to MTU7.TGR registers are 16-bit readable/writable registers; the MTU8.TGR register is a 32-bit readable/writable register. The MTU has a total of 39 TGR registers, six for MTU0, two each for MTU1 and MTU2, five each for MTU3 and MTU6, six each for MTU4 and MTU7, and three for MTU5, and four for MTU8.
The TGRA, TGRB, TGRC, and TGRD registers function as either output compare or input capture registers. The TGRC and TGRD registers for MTU0, MTU3, MTU4, MTU6, MTU7, and MTU8 can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.
MTU0.TGRE and MTU0.TGRF function as compare registers. When the MTU0.TCNT count matches the MTU0.TGRE value, an A/D converter start request can be issued. The TGRF register can also be designated for operation as a buffer register. TGR buffer register combination is TGRE and TGRF. MTU5.TGRU, MTU5.TGRV, and MTU5.TGRW function as compare match, input capture, or external pulse width measurement registers.
The MTU1.TGRA, MTU2.TGRA, MTU1.TGRB, and MTU2.TGRB registers are read as 0000h when TMDR3.LWA is 1. Refer to section 24.2.5, Timer Mode Register 3 (TMDR3) for details.

24.2.16 Timer Longword General Registers (TGRALW, TGRBLW)

Address(es): MTU1.TGRALW 000C 13A4h, MTU1.TGRBLW 000C 13A8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: TGRALW and TGRBLW must not be accessed in 8 or 16 bits; it should be accessed in 32 bits.

The TGRnLW register (n = A, B) is a 32-bit readable/writable register. Two general registers of this type are provided, and are formed by combining MTU1.TGRn and MTU2.TGRn. Such operation is only effective when TMDR3.LWA is 1. The TGRnLW register is initialized to FFFF FFFFh by a reset, but it is read as 0000 0000h when TMDR3.LWA is 0. Refer to section 24.2.5, Timer Mode Register 3 (TMDR3) for details.

The TGRnLW register functions as an output compare or input capture register when TMDR3.LWA is 1.

This register can only be used in 32-bit phase counting mode.

24.2.17 Timer Start Registers (TSTRA, TSTRB, TSTR)

- MTU.TSTRA (for MTU0, MTU1, MTU2, MTU3, MTU4, and MTU8

Address(es): MTU.TSTRA 000C 1280h

	b7	b6	b5	b4	b3	b2	b1	b0
	CST4	CST3	—	—	CST8	CST2	CST1	CST0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CST0	Counter Start 0	0: MTU0.TCNT counting is stopped 1: MTU0.TCNT performs count operation	R/W
b1	CST1	Counter Start 1	0: MTU1.TCNT counting is stopped 1: MTU1.TCNT performs count operation	R/W
b2	CST2	Counter Start 2	0: MTU2.TCNT counting is stopped 1: MTU2.TCNT performs count operation	R/W
b3	CST8	Counter start 8	0: MTU8.TCNT counting is stopped 1: MTU8.TCNT performs count operation.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CST3	Counter Start 3	0: MTU3.TCNT counting is stopped 1: MTU3.TCNT performs count operation	R/W
b7	CST4	Counter Start 4	0: MTU4.TCNT counting is stopped 1: MTU4.TCNT performs count operation	R/W

Note: When 1 is written to a bit in TCSYSTR, the corresponding bit in TSTRA is also set to 1 automatically.

The TSTRA register starts or stops TCNT operation in MTU0 to MTU4 and MTU8.
TSTRB starts or stops TCNT operation in MTU6 and MTU7.
TSTR starts or stops TCNT operation in MTU5.
Before setting the operating mode in TMDR1 or setting the TCNT count clock in TCR, be sure to stop the TCNT counter.

CSTn Bits (Counter Start n) (n = 0, 1, 2, 3, 4, 8)

Each bit starts or stops TCNT in the corresponding channel.
If 0 is written to the CSTn bit during operation with the MTIOC pin designated for output, the counter stops but the output compare signal level from the MTIOC pin is retained. If TIOR is written to while the CSTn bit is 0, the pin output level will be changed to the specified initial output value.

- MTU.TSTRB (for MTU6 and MTU7)

Address(es): MTU.TSTRB 000C 1A80h

	b7	b6	b5	b4	b3	b2	b1	b0
	CST7	CST6	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CST6	Counter Start 6	0: MTU6.TCNT counting is stopped 1: MTU6.TCNT performs count operation	R/W
b7	CST7	Counter Start 7	0: MTU7.TCNT counting is stopped 1: MTU7.TCNT performs count operation	R/W

Note: When 1 is written to a bit in TCSYSTR, the corresponding bit in TSTRB is also set to 1 automatically.

CSTn Bits (Counter Start n) (n = 6, 7)

Each bit starts or stops TCNT in the corresponding channel.
If 0 is written to the CSTn bit during operation with the MTIOC pin designated for output, the counter stops but the output compare signal level from the MTIOC pin is retained. If TIOR is written to while the CSTn bit is 0, the pin output level will be changed to the specified initial output value.

- MTU5.TSTR

Address(es): MTU5.TSTR 000C 1CB4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CSTU5	CSTV5	CSTW5
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CSTW5	Counter Start W5	0: MTU5.TCNTW counting is stopped 1: MTU5.TCNTW performs count operation	R/W
b1	CSTV5	Counter Start V5	0: MTU5.TCNTV counting is stopped 1: MTU5.TCNTV performs count operation	R/W
b2	CSTU5	Counter Start U5	0: MTU5.TCNTU counting is stopped 1: MTU5.TCNTU performs count operation	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

24.2.18 Timer Synchronous Registers (TSYRA, TSYRB)

- MTU.TSYRA (for MTU0 to MTU4)

Address(es): MTU.TSYRA 000C 1281h

	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SYNC0	Timer Synchronous Operation 0	0: MTU0.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU0.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b1	SYNC1	Timer Synchronous Operation 1	0: MTU1.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU1.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b2	SYNC2	Timer Synchronous Operation 2	0: MTU2.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU2.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b5 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	SYNC3	Timer Synchronous Operation 3	0: MTU3.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU3.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b7	SYNC4	Timer Synchronous Operation 4	0: MTU4.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU4.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W

TSYRA selects independent operation or synchronous operation of TCNT in MTU0 to MTU4.
TSYRB selects independent operation or synchronous operation of TCNT in MTU6 and MTU7.
A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

SYNCn Bits (Timer Synchronous Operation n) (n = 0, 1, 2, 3, 4)

Each bit selects whether operation is independent of or synchronized with other channels.
When synchronous operation is selected, the TCNT synchronous setting of multiple channels and synchronous clearing by counter clearing on another channel are possible.
To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of the TCR.CCLR[2:0] bits.

- MTU.TSYRB (for MTU6 and MTU7)

Address(es): MTU.TSYRB 000C 1A81h

	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC7	SYNC6	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	SYNC6	Timer Synchronous Operation 6	0: MTU6.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU6.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b7	SYNC7	Timer Synchronous Operation 7	0: MTU7.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU7.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W

SYNCn Bits (Timer Synchronous Operation n) (n = 6, 7)

Each bit selects whether operation is independent of or synchronized with other channels.

When synchronous operation is selected, the TCNT synchronous setting of multiple channels and synchronous clearing by counter clearing on another channel are possible.

To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits TCR.CCLR[2:0].

24.2.19 Timer Counter Synchronous Start Register (TCSYSTR)

Address(es): MTU.TCSYSTR 000C 1282h

	b7	b6	b5	b4	b3	b2	b1	b0
	SCH0	SCH1	SCH2	SCH3	SCH4	—	SCH6	SCH7
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SCH7	Synchronous Start 7	0: Does not specify synchronous start for MTU7.TCNT 1: Specifies synchronous start for MTU7.TCNT	R/(W)*1
b1	SCH6	Synchronous Start 6	0: Does not specify synchronous start for MTU6.TCNT 1: Specifies synchronous start for MTU6.TCNT	R/(W)*1
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R
b3	SCH4	Synchronous Start 4	0: Does not specify synchronous start for MTU4.TCNT 1: Specifies synchronous start for MTU4.TCNT	R/(W)*1
b4	SCH3	Synchronous Start 3	0: Does not specify synchronous start for MTU3.TCNT 1: Specifies synchronous start for MTU3.TCNT	R/(W)*1
b5	SCH2	Synchronous Start 2	0: Does not specify synchronous start for MTU2.TCNT 1: Specifies synchronous start for MTU2.TCNT	R/(W)*1
b6	SCH1	Synchronous Start 1	0: Does not specify synchronous start for MTU1.TCNT 1: Specifies synchronous start for MTU1.TCNT	R/(W)*1
b7	SCH0	Synchronous Start 0	0: Does not specify synchronous start for MTU0.TCNT 1: Specifies synchronous start for MTU0.TCNT	R/(W)*1

Note 1. Only 1 can be written to this bit, and doing so sets the flag.
TCSYSTR is automatically cleared after 1 is written to.

TCSYSTR specifies synchronous start of the counters.

SCH7 Bit (Synchronous Start 7)

This bit controls synchronous start of MTU7.TCNT.
[Clearing condition]

- When 1 is set to the TSTRA.CST7 bit while SCH7 = 1

SCH6 Bit (Synchronous Start 6)

This bit controls synchronous start of MTU6.TCNT.
[Clearing condition]

- When 1 is set to the TSTRA.CST6 bit while SCH6 = 1

SCH4 Bit (Synchronous Start 4)

This bit controls synchronous start of MTU4.TCNT.
[Clearing condition]

- When 1 is set to the TSTRA.CST4 bit while SCH4 = 1

SCH3 Bit (Synchronous Start 3)

This bit controls synchronous start of MTU3.TCNT.
[Clearing condition]

- When 1 is set to the TSTRA.CST3 bit while SCH3 = 1

SCH2 Bit (Synchronous Start 2)

This bit controls synchronous start of MTU2.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST2 bit while SCH2 = 1

SCH1 Bit (Synchronous Start 1)

This bit controls synchronous start of MTU1.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST1 bit while SCH1 = 1

SCH0 Bit (Synchronous Start 0)

This bit controls synchronous start of MTU0.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST0 bit while SCH0 = 1

24.2.20 Timer Read/Write Enable Registers (TRWERA, TRWERB)

Address(es): MTU.TRWERA 000C 1284h, MTU.TRWERB 000C 1A84h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RWE
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RWE	Read/Write Enable	0: Read/write access to the registers is disabled 1: Read/write access to the registers is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TRWERA enables or disables access to the registers and counters that have write-protection capability against accidental modification in MTU3 and MTU4.
TRWERB enables or disables access to the registers and counters that have write-protection capability against accidental modification in MTU6 and MTU7.

RWE Bit (Read/Write Enable)

This bit enables or disables access to the registers that have write-protection capability against accidental modification.
[Clearing condition]

- When 0 is written to the RWE bit after reading RWE = 1

- Registers and Counters having Write-Protection Capability against Accidental Modification (TRWERA)

24 registers: MTUn.TCR, MTUn.TCR2, MTUn.TMDR1, MTUn.TIORH, MTUn.TIORL, MTUn.TIER, MTUn.TGRA, MTUn.TGRB, MTU.TOERA, MTU.TOCR1A, MTU.TOCR2A, MTU.TGCRA, MTU.TCDRA, MTU.TDDRA, and MTUn.TCNT (n = 3, 4)

- Registers and Counters having Write-Protection Capability against Accidental Modification (TRWERB)

23 registers: MTUn.TCR, MTUn.TCR2, MTUn.TMDR1, MTUn.TIORH, MTUn.TIORL, MTUn.TIER, MTUn.TGRA, MTUn.TGRB, MTU.TOERB, MTU.TOCR1B, MTU.TOCR2B, MTU.TCDRB, MTU.TDDRB, and MTUn.TCNT (n = 6, 7)

24.2.21 Timer Output Master Enable Registers (TOERA, TOERB)

- MTU.TOERA

Address(es): MTU.TOERA 000C 120Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OE3B	Master Enable MTIOC3B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b1	OE4A	Master Enable MTIOC4A	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b2	OE4B	Master Enable MTIOC4B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b3	OE3D	Master Enable MTIOC3D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b4	OE4C	Master Enable MTIOC4C	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b5	OE4D	Master Enable MTIOC4D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b7, b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. To output the inactive level from each pin when the MTU output is set to disabled, first set the data direction register (PDR) and port output data register (PODR) of I/O ports to output the inactive level from general I/O ports, and then set the port mode register (PMR) to use general I/O ports. For details, refer to section 22, I/O Ports.

TOERA enables or disables output settings for output pins MTIOC4D, MTIOC4C, MTIOC3D, MTIOC4B, MTIOC4A, and MTIOC3B.

These pins do not output correctly if the bits in the TOERA register have not been set. In MTU3 and MTU4, set TOERA prior to setting TIOR.

Set MTU.TOERA after clearing the CST3 and CST4 bits in MTU.TSTRA to 0 (refer to Figure 24.44 and Figure 24.48).

- MTU.TOERB

Address(es): MTU.TOERB 000C 1A0Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	OE7D	OE7C	OE6D	OE7B	OE7A	OE6B
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OE6B	Master Enable MTIOC6B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b1	OE7A	Master Enable MTIOC7A	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b2	OE7B	Master Enable MTIOC7B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b3	OE6D	Master Enable MTIOC6D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b4	OE7C	Master Enable MTIOC7C	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b5	OE7D	Master Enable MTIOC7D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b7, b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. To output the inactive level from each pin when the MTU output is set to disabled, first set the data direction register (PDR) and port output data register (PODR) of I/O ports to output the inactive level from general I/O ports, and then set the port mode register (PMR) to use general I/O ports. For details, refer to section 22, I/O Ports.

TOERB enables or disables output settings for output pins MTIOC7D, MTIOC7C, MTIOC6D, MTIOC7B, MTIOC7A, and MTIOC6B.

These pins do not output correctly if the bits in the TOERB register have not been set. In MTU6, and MTU7, set TOERB prior to setting TIOR.

Set MTU.TOERB after clearing the CST6 and CST7 bits in MTU.TSTRB to 0 (refer to Figure 24.44 and Figure 24.48).

24.2.22 Timer Output Control Registers 1 (TOCR1A, TOCR1B)

Address(es): MTU.TOCR1A 000C 120Eh, MTU.TOCR1B 000C 1A0Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
Value after reset:	0	0	0	0	0*4	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLSP	Output Level Select P*1, *3	Refer to Table 24.42.	R/W
b1	OLSN	Output Level Select N*1, *3	Refer to Table 24.43.	R/W
b2	TOCS	TOC Select	0: TOCR1j setting is selected (j = A, B) 1: TOCR2j setting is selected	R/W
b3	TOCL	TOC Register Write Protection*2	0: Write access to the TOCS, OLSN, and OLSP bits is enabled 1: Write access to the TOCS, OLSN, and OLSP bits is disabled	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PSYE	PWM Synchronous Output Enable	0: Toggle output is disabled 1: Toggle output is enabled	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Clearing the TOCR1j.TOCS bit to 0 makes this bit setting valid.
Note 2. Setting the TOCR1j.TOCL bit to 1 prevents accidental modification when the CPU goes out of control.
Note 3. If dead-time is not generated, the negative-phase output is the exact inverse of the positive-phase output. In this case, only the OLSP bit is valid.
Note 4. This bit can be set to 1 only once after a reset. After 1 is written, 0 cannot be written to the bit.

TOCR1A and TOCR1B enable or disable PWM-synchronized toggle output in complementary PWM mode and reset-synchronized PWM mode, and control inversion of PWM output level.

OLSP Bit (Output Level Select P)

This bit selects the positive-phase output level in reset-synchronized PWM mode and complementary PWM mode. The initial output is selected while the counter is stopped.

OLSN Bit (Output Level Select N)

This bit selects the negative-phase output level in reset-synchronized PWM mode and complementary PWM mode. The initial output is selected while the counter is stopped.

TOCS Bit (TOC Select)

This bit selects either the TOCR1j or TOCR2j (j = A, B) setting to be used for the output level in complementary PWM mode and reset-synchronized PWM mode.

TOCL Bit (TOC Register Write Protection)

This bit enables or disables write access to the TOCS, OLSN, and OLSP bits in TOCR1j (j = A, B).

PSYE Bit (PWM Synchronous Output Enable)

This bit enables or disables toggle output synchronized with the PWM cycle.

Table 24.42 Output Level Select Function

Bit 0			Function	
			Compare Match Output	
OLSP	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Table 24.43 Output Level Select Function

Bit 1			Function	
			Compare Match Output	
OLSN	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Figure 24.3 shows an example of output in complementary PWM mode (one phase) when OLSN = 1 and OLSP = 1.

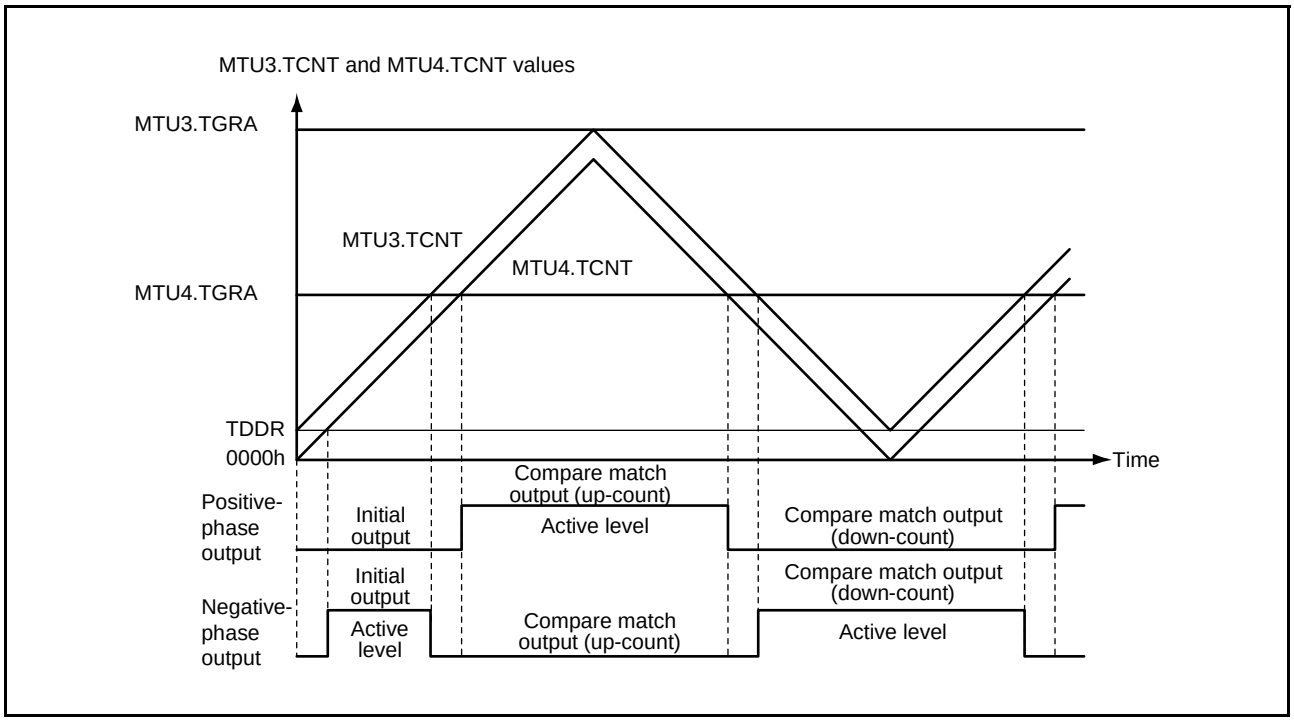


Figure 24.3 Example of Output in Complementary PWM Mode

24.2.23 Timer Output Control Registers 2 (TOCR2A, TOCR2B)

Address(es): MTU.TOCR2A 000C 120Fh, MTU.TOCR2B 000C 1A0Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLS1P	Output Level Select 1P*1, *2	This bit selects the output level on MTIOC3B or MTIOC6B in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 24.44.	R/W
b1	OLS1N	Output Level Select 1N*1, *2	This bit selects the output level on MTIOC3D or MTIOC6D in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 24.45.	R/W
b2	OLS2P	Output Level Select 2P*1, *2	This bit selects the output level on MTIOC4A or MTIOC7A in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 24.46.	R/W
b3	OLS2N	Output Level Select 2N*1, *2	This bit selects the output level on MTIOC4C or MTIOC7C in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 24.47.	R/W
b4	OLS3P	Output Level Select 3P*1, *2	This bit selects the output level on MTIOC4B or MTIOC7B in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 24.48.	R/W
b5	OLS3N	Output Level Select 3N*1, *2	This bit selects the output level on MTIOC4D or MTIOC7D in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 24.49.	R/W
b7, b6	BF[1:0]	TOLBR Buffer Transfer Timing Select	These bits select the timing for transferring data from TOLBRj to TOCR2j. Refer to Table 24.50 for details.	R/W

j = A, B

Note 1. Setting the TOCR1j.TOCS bit to 1 makes this bit setting valid.

Note 2. If dead-time is not generated, the negative-phase output is the exact inverse of the positive-phase output. In this case, only the OLSiP bits are valid (i = 1 to 3).

TOCR2A and TOCR2B control inversion of PWM output level in complementary PWM mode and reset-synchronized PWM mode.

The initial output is selected while the counter is stopped.

Table 24.44 MTIOCmB Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLS1P	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

m = 3, 6

Table 24.45 MTIOCmD Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLS1N	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

m = 3, 6

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 24.46 MTIOCmA Output Level Select Function

Bit 2	Function			
			Compare Match Output	
OLS2P	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

m = 4, 7

Table 24.47 MTIOCmC Output Level Select Function

Bit 3	Function			
			Compare Match Output	
OLS2N	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

m = 4, 7

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 24.48 MTIOCmB Output Level Select Function

Bit 4	Function			
			Compare Match Output	
OLS3P	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

m = 4, 7

Table 24.49 MTIOCmD Output Level Select Function

Bit 5	Function			
			Compare Match Output	
OLS3N	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

m = 4, 7

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 24.50 Setting of TOCR2j.BF[1:0] Bits

Bit 7	Bit 6	Description	
BF[1]	**BF[0]**	**Complementary PWM Mode**	**Reset-Synchronized PWM Mode**
0	0	Does not transfer data from the buffer register (TOLBRj) to TOCR2j.	Does not transfer data from the buffer register (TOLBRj) to TOCR2j.
0	1	Transfers data from the buffer register (TOLBRj) to TOCR2j at the crest of the MTUn.TCNT count.	Transfers data from the buffer register (TOLBRj) to TOCR2j when MTUm.TCNT or MTUn.TCNT is cleared.
1	0	Transfers data from the buffer register (TOLBRj) to TOCR2j at the trough of the MTUn.TCNT count.	Setting prohibited
1	1	Transfers data from the buffer register (TOLBRj) to TOCR2j at the crest and trough of the MTUn.TCNT count.	Setting prohibited

n = 4, 7; m = 3, 6; j = A, B

24.2.24 Timer Output Level Buffer Registers (TOLBRA, TOLBRB)

Address(es): MTU.TOLBRA 000C 1236h, MTU.TOLBRB 000C 1A36h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLS1P	Output Level Select 1P	Specify the buffer value to be transferred to the OLS1P bit in TOCR2j.	R/W
b1	OLS1N	Output Level Select 1N	Specify the buffer value to be transferred to the OLS1N bit in TOCR2j.	R/W
b2	OLS2P	Output Level Select 2P	Specify the buffer value to be transferred to the OLS2P bit in TOCR2j.	R/W
b3	OLS2N	Output Level Select 2N	Specify the buffer value to be transferred to the OLS2N bit in TOCR2j.	R/W
b4	OLS3P	Output Level Select 3P	Specify the buffer value to be transferred to the OLS3P bit in TOCR2j.	R/W
b5	OLS3N	Output Level Select 3N	Specify the buffer value to be transferred to the OLS3N bit in TOCR2j.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

j = A, B

TOLBRA and TOLBRB are buffer registers for TOCR2A and TOCR2B and specify the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Figure 24.4 shows an example of the PWM output level setting procedure in buffer operation.

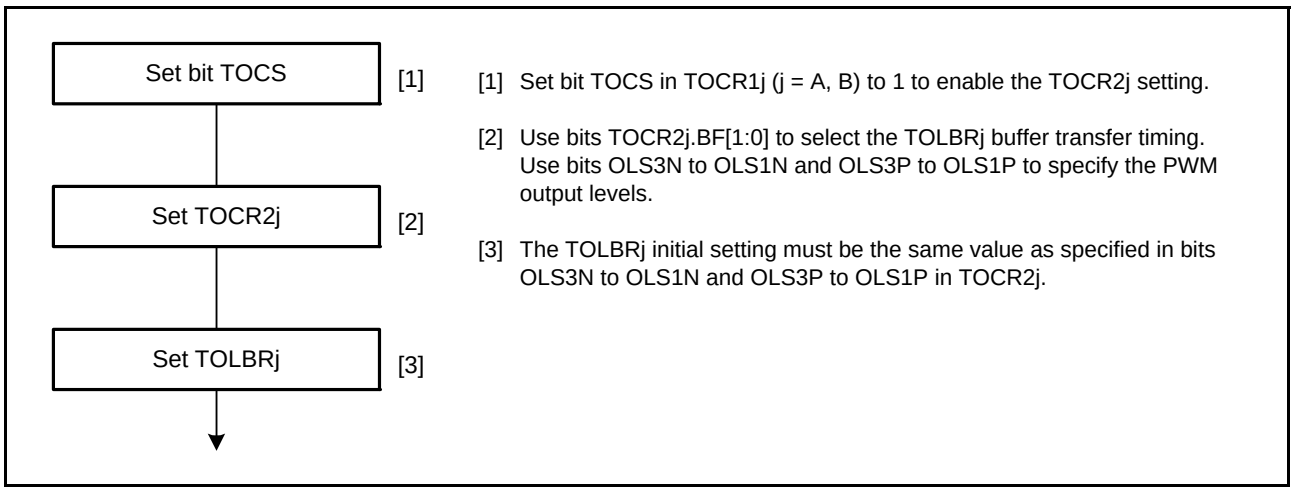


Figure 24.4 Example of PWM Output Level Setting Procedure in Buffer Operation

24.2.25 Timer Gate Control Register A (TGCRA)

Address(es): MTU.TGCRA 000C 120Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	BDC	N	P	FB	WF	VF	UF
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	UF	Output Phase Switch	These bits turn on or off the positive-phase/negative-phase output. The setting of these bits is valid only when the FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. Refer to Table 24.51.	R/W
b1	VF			R/W
b2	WF			R/W
b3	FB	External Feedback Signal Enable	0: Output is switched by external input (input sources are TGRA, TGRB, and TGRC input capture signals in MTU0) 1: Output is switched by software (TGCRA's UF, VF, and WF settings)	R/W
b4	P	Positive-Phase Output (P) Control	0: Level output 1: Reset-synchronized PWM or complementary PWM output	R/W
b5	N	Negative-Phase Output (N) Control	0: Level output 1: Reset-synchronized PWM or complementary PWM output	R/W
b6	BDC	Brushless DC Motor	0: Ordinary output 1: Functions of this register are made effective	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

TGCRA controls the output waveform necessary for brushless DC motor control in reset-synchronized PWM mode and complementary PWM mode. TGCRA register settings are ineffective for anything other than complementary PWM mode and reset-synchronized PWM mode.

UF, VF, and WF Bits (Output Phase Switch)

The setting of these bits is valid only when the FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. Refer to Table 24.51 for details.

FB Bit (External Feedback Signal Enable)

This bit selects whether the positive-/negative-phase output is switched automatically with the TGRA, TGRB, and TGRC input capture signals in MTU0 or by writing 0 or 1 to bits 2 to 0 in TGCRA.
When the TGCRA.FB bit is 0, output of MTU3 and MTU4 can be switched with the TGRA, TGRB, and TGRC input capture signals in MTU0.

P Bit (Positive-Phase Output (P) Control)

This bit selects the level output or the reset-synchronized PWM/complementary PWM output for the positive-phase output pins (MTIOC3B, MTIOC4A, and MTIOC4B pins).

N Bit (Negative-Phase Output (N) Control)

This bit selects the level output or the reset-synchronized PWM/complementary PWM output for the negative-phase output pins (MTIOC3D, MTIOC4C, and MTIOC4D pins).

BDC Bit (Brushless DC Motor)

This bit selects whether to make the functions of TGCRA effective or ineffective.

Table 24.51 Output Level Select Function

Bit 2	Bit 1	Bit 0	Function					
			MTIOC3B	MTIOC4A	MTIOC4B	MTIOC3D	MTIOC4C	MTIOC4D
WF	VF	UF	U Phase	V Phase	W Phase	U Phase	V Phase	W Phase
0	0	0	OFF	OFF	OFF	OFF	OFF	OFF
0	0	1	ON	OFF	OFF	OFF	OFF	ON
0	1	0	OFF	ON	OFF	ON	OFF	OFF
0	1	1	OFF	ON	OFF	OFF	OFF	ON
1	0	0	OFF	OFF	ON	OFF	ON	OFF
1	0	1	ON	OFF	OFF	OFF	ON	OFF
1	1	0	OFF	OFF	ON	ON	OFF	OFF
1	1	1	OFF	OFF	OFF	OFF	OFF	OFF

24.2.26 Timer Subcounters (TCNTSA, TCNTSB)

Address(es): MTU.TCNTSA 000C 1220h, MTU.TCNTSB 000C 1A20h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: TCNTSA and TCNTSB must not be accessed in 8 bits; it should be accessed in 16 bits.

TCNTSA and TCNTSB are 16-bit read-only counters used only in complementary PWM mode.
The initial value of TCNTSA and TCNTSB after a reset is 0000h.

24.2.27 Timer Cycle Data Registers (TCDRA, TCDRB)

Address(es): MTU.TCDRA 000C 1214h, MTU.TCDRB 000C 1A14h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: TCDRA and TCDRB must not be accessed in 8 bits; it should be accessed in 16 bits.

TCDRA and TCDRB are 16-bit readable/writable registers used only in complementary PWM mode. Set half the PWM carrier cycle as the TCDRA and TCDRB values. The TCDRA and TCDRB registers are constantly compared with the TCNTSA and TCNTSB counters in complementary PWM mode, respectively. When a match occurs, the TCNTSA and TCNTSB counters switch the count direction (down-count to up-count). The initial value of TCDRA and TCDRB after a reset is FFFFh.

24.2.28 Timer Cycle Buffer Registers (TCBRA, TCBRB)

Address(es): MTU.TCBRA 000C 1222h, MTU.TCBRB 000C 1A22h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: TCBRA and TCBRB must not be accessed in 8 bits; it should be accessed in 16 bits.

TCBRA and TCBRB are 16-bit readable/writable registers, used only in complementary PWM mode, that function as buffer registers for TCDRA and TCDRB. The TCBRA and TCBRB values are transferred to TCDRA and TCDRB with the transfer timing set in TMDR1. The initial value of TCBRA and TCBRB after a reset is FFFFh.

24.2.29 Timer Dead Time Data Registers (TDDRA, TDDRB)

Address(es): MTU.TDDRA 000C 1216h, MTU.TDDRB 000C 1A16h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: TDDRA and TDDRB must not be accessed in 8 bits; it should be accessed in 16 bits.

TDDRA and TDDRB are 16-bit readable/writable registers, used only in complementary PWM mode, that specify the MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) counter offset value. In complementary PWM mode, when the MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) counters are cleared and then restarted, the TDDRA (TDDRB) value is loaded into the MTU3.TCNT (MTU6.TCNT) counter and the count operation starts. The initial value of TDDRA and TDDRB after a reset is FFFFh.

24.2.30 Timer Dead Time Enable Registers (TDERA, TDERB)

Address(es): MTU.TDERA 000C 1234h, MTU.TDERB 000C 1A34h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TDER
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	TDER	Dead Time Enable	0: No dead time is generated 1: Dead time is generated*1	R/(W)
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. TDDRA and TDDRB must be set to 1 or a larger value.

TDERA and TDERB control dead time generation in complementary PWM mode. The MTU has one TDER each for MTU3 and MTU6. TDERA and TDERB should be modified only while TCNT stops.

TDER Bit (Dead Time Enable)

This bit specifies whether to generate dead time.

[Clearing condition]

- When 0 is written to TDER after reading TDER = 1

24.2.31 Timer Buffer Transfer Set Registers (TBTERA, TBTERB)

Address(es): MTU.TBTERA 000C 1232h, MTU.TBTERB 000C 1A32h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	BTE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	BTE[1:0]	Buffer Transfer Disable and Interrupt Skipping Link Setting	These bits enable or disable transfer from the buffer registers*1 used in complementary PWM mode to the temporary registers, and specify whether to link the transfer with interrupt skipping function 1. For details, refer to Table 24.52.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Applicable buffer registers (TBTERA):
MTU3.TGRC, MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, and TCBRA
Applicable buffer registers (TBTERB):
MTU6.TGRC, MTU6.TGRD, MTU7.TGRC, MTU7.TGRD, and TCBRB

TBTERA and TBTERB enable or disable transfer from the buffer registers used in complementary PWM mode to the temporary registers, and specify whether to link the transfer with interrupt skipping 1 operation.

Table 24.52 Setting of TBTERA.BTE[1:0] Bits and TBTERB.BTE[1:0] Bits

Bit 1	Bit 0	
BTE[1]	BTE[0]	Description
0	0	Enables transfer from the buffer registers to the temporary registers*1 and does not link the transfer with interrupt skipping function 1.
0	1	Disables transfer from the buffer registers to the temporary registers.
1	0	Links transfer from the buffer registers to the temporary registers with interrupt skipping function 1.*2
1	1	Setting prohibited

Note 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR1. For details, refer to section 24.3.8, Complementary PWM Mode.

Note 2. When interrupt skipping is disabled the T3AEN and T4VEN (T6AEN and T7VEN) bits are cleared to 0 in the timer interrupt skipping set register (TITCR1A (TITCR1B)) or the skipping count set bits (T3ACOR and T4VCOR (T6ACOR and T7VCOR)) in TITCR1A (TITCR1B) are cleared to 0), be sure to disable link of buffer transfer with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTERA (TBTERB)) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

24.2.32 Timer Waveform Control Registers (TWCRA, TWCRB)

Address(es): MTU.TWCRA 000C 1260h, MTU.TWCRB 000C 1A60h

	b7	b6	b5	b4	b3	b2	b1	b0
	CCE	—	—	—	—	—	SCC	WRE
Value after reset:	0*2	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WRE	Waveform Retain Enable	0: Initial values specified in TOCR1A and TOCR2A (TOCR1B and TOCR2B) are output 1: Initial output is inhibited	R/(W)*3
b1	SCC *1, *3	Synchronous Clearing Control	(Only valid in TWCRB) 0: Clearing of MTU6.TCNT and MTU7.TCNT in response to synchronous clearing for MTU0, MTU1, MTU2–MTU6, MTU7 is enabled. 1: Clearing of MTU6.TCNT and MTU7.TCNT in response to synchronous clearing for MTU0, MTU1, MTU2–MTU6, MTU7 is disabled.	R/(W)
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CCE*2	Compare Match Clear Enable	0: Counters are not cleared at MTU3.TGRA (MTU6.TGRA) compare match 1: Counters are cleared at MTU3.TGRA (MTU6.TGRA) compare match	R/(W)

Note 1. This bit is only valid in register TWCRB and is a reserved bit in register TWCRA.
Note 2. Do not set to 1 when complementary PWM mode 1 is not selected.
Note 3. Do not set to 1 when complementary PWM mode is not selected.

TWCRA and TWCRB control the output waveform when synchronous counter clearing occurs in MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) in complementary PWM mode and specifies whether to clear the counters at MTU3.TGRA (MTU6.TGRA) compare match.
The CCE bit and WRE bit in TWCRA and TWCRB should be modified only while TCNT stops.

WRE Bit (Waveform Retain Enable)

This bit selects the waveform output when synchronous counter clearing occurs in complementary PWM mode.
The initial output is inhibited with this function only when synchronous clearing occurs within the Tb interval at the trough in complementary PWM mode. When synchronous clearing occurs outside this interval, the initial values specified in TOCR1A and TOCR2A (TOCR1B and TOCR2B) are output regardless of the WRE bit setting. The initial values specified in TOCR1A and TOCR2A (TOCR1B and TOCR2B) are also output when synchronous clearing occurs in the Tb interval at the trough immediately after MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) start operation.
For the Tb interval at the trough in complementary PWM mode, refer to Figure 24.50.
[Setting condition]

- When 1 is written to the WRE bit after reading WRE = 0

SCC Bit (Synchronous Clearing Control)

The setting of this bit selects whether MTU6.TCNT and MTU7.TCNT are or are not cleared when counter-synchronous clearing is generated for MTU0, MTU1, MTU2–MTU6, MTU7 in complementary PWM mode.
Make the complementary PWM mode settings for MTU6 and MTU7 when this function is in use. When writing a new value to the SCC bit while the counter is operating, do so in such a way that the values of the CCE and WRE bits are not changed.
Synchronous clearing from the MTU module only becomes disabled due to the setting of the SCC bit when synchronous clearing is generated outside the Tb interval in the trough. If synchronous clearing is generated within the Tb interval in the trough including immediately after the value at which MTU6.TCNT and MTU7.TCNT start, MTU6.TCNT and MTU7.TCNT are cleared.
Regarding the Tb interval in the trough in complementary PWM mode, refer to Figure 24.50.
[Setting condition]

- Writing of 1 to the SCC bit after reading it as 0

The corresponding bit in register TWCRA is reserved and is read as 0. When writing to TWCRA, write 0 to this bit.

CCE Bit (Compare Match Clear Enable)

This bit specifies whether to clear counters at MTU3.TGRA (MTU6.TGRA) compare match in complementary PWM mode.
[Setting condition]

- When 1 is written to CCE after reading CCE = 0

24.2.33 Noise Filter Control Register n (NFCRn) (n = 0 to 4, 6, 7, 8, C)

- MTU0.NFCR0, MTU1.NFCR1, MTU2.NFCR2, MTU3.NFCR3, MTU4.NFCR4, MTU6.NFCR6, MTU7.NFCR7, MTU8.NFCR8

Address(es): MTU0.NFCR0 000C 1290h, MTU1.NFCR1 000C 1291h, MTU2.NFCR2 000C 1292h, MTU3.NFCR3 000C 1293h, MTU4.NFCR4 000C 1294h, MTU6.NFCR6 000C 1A93h, MTU7.NFCR7 000C 1A94h, MTU8.NFCR8 000C 1298h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFCS[1:0]		NFDEN	NFCEN	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFAEN	Noise Filter A Enable	0: The noise filter for the MTIOCnA pin is disabled. 1: The noise filter for the MTIOCnA pin is enabled.	R/W
b1	NFBEN	Noise Filter B Enable	0: The noise filter for the MTIOCnB pin is disabled. 1: The noise filter for the MTIOCnB pin is enabled.	R/W
b2	NFCEN	Noise Filter C Enable	0: The noise filter for the MTIOCnC pin is disabled. 1: The noise filter for the MTIOCnC pin is enabled.	R/W*1
b3	NFDEN	Noise Filter D Enable	0: The noise filter for the MTIOCnD pin is disabled. 1: The noise filter for the MTIOCnD pin is enabled.	R/W*1
b5, b4	NFCS[1:0]	Noise Filter Clock Select	b5 b4 0 0: PCLKA/1 0 1: PCLKA/8 1 0: PCLKA/32 1 1: Clock source for counting	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits are reserved in the NFCR1 and NFCR2 registers. These bits are read as 0 and writing to them has no effect.

The NFCRn register sets the noise filter function of external clock pins common to each channel.

NFAEN Bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTIOCnA pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFBEN Bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTIOCnB pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFCEN Bit (Noise Filter C Enable)

This bit disables or enables the noise filter for input from the MTIOCnC pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFDEN Bit (Noise Filter D Enable)

This bit disables or enables the noise filter for input from the MTIOCnD pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFCS[1:0] Bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function. When the NFCS[1:0] bits are set to 11b, i.e. selecting the external clock as the source to drive counting, wait for two cycles of the external clock before setting the input capture function.

- MTU0.NFCRC

Address(es): MTU0.NFCRC 000C 1299h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFCSC[1:0]		NFDEN	NFCEN	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFAEN	Noise Filter A Enable	0: The noise filter for the MTCLKA pin is disabled. 1: The noise filter for the MTCLKA pin is enabled.	R/W
b1	NFBEN	Noise Filter B Enable	0: The noise filter for the MTCLKB pin is disabled. 1: The noise filter for the MTCLKB pin is enabled.	R/W
b2	NFCEN	Noise Filter C Enable	0: The noise filter for the MTCLKC pin is disabled. 1: The noise filter for the MTCLKC pin is enabled.	R/W
b3	NFDEN	Noise Filter D Enable	0: The noise filter for the MTCLKD pin is disabled. 1: The noise filter for the MTCLKD pin is enabled.	R/W
b5, b4	NFCSC[1:0]	Noise Filter Clock Select	b5 b4 0 0: PCLKA/1 0 1: PCLKA/2 1 0: PCLKA/8 1 1: PCLKA/32	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

NFAEN Bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTCLKA pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFBEN Bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTCLKB pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFCEN Bit (Noise Filter C Enable)

This bit disables or enables the noise filter for input from the MTCLKC pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFDEN Bit (Noise Filter D Enable)

This bit disables or enables the noise filter for input from the MTCLKD pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFCSC[1:0] Bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. After setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval to set the input capture function.

24.2.34 Noise Filter Control Register 5 (NFCR5)

Address(es): MTU5.NFCR5 000C 1A95h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFCS[1:0]		—	NFWEN	NFVEN	NFUEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFUEN	Noise Filter U Enable	0: The noise filter for the MTIC5U pin is disabled. 1: The noise filter for the MTIC5U pin is enabled.	R/W
b1	NFVEN	Noise Filter V Enable	0: The noise filter for the MTIC5V pin is disabled. 1: The noise filter for the MTIC5V pin is enabled.	R/W
b2	NFWEN	Noise Filter W Enable	0: The noise filter for the MTIC5W pin is disabled. 1: The noise filter for the MTIC5W pin is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	NFCS[1:0]	Noise Filter Clock Select	b5 b4 0 0: PCLKA/1 0 1: PCLKA/8 1 0: PCLKA/32 1 1: Clock source for counting	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

NFUEN Bit (Noise Filter U Enable)

This bit disables or enables the noise filter for input from the MTIC5U pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

NFVEN Bit (Noise Filter V Enable)

This bit disables or enables the noise filter for input from the MTIC5V pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

NFWEN Bit (Noise Filter W Enable)

This bit disables or enables the noise filter for input from the MTIC5W pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

NFCS[1:0] Bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function.

24.2.35 Timer A/D Converter Start Request Control Register (TADCR)

- MTU4.TADCR

Address(es): MTU4.TADCR 000C 1240h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BF[1:0]		—	—	—	—	—	—	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ITB4VE	TCIV4 Interrupt Skipping Link Enable*1, *2, *3	0: A/D converter start request signal TRG4BN and TCI4V interrupt skipping 1 are not linked 1: A/D converter start request signal TRG4BN and TCI4V interrupt skipping 1 are linked	R/W
b1	ITB3AE	TGIA3 Interrupt Skipping Link Enable*1, *2, *3	0: A/D converter start request signal TRG4BN and TGI3A interrupt skipping 1 are not linked 1: A/D converter start request signal TRG4BN and TGI3A interrupt skipping 1 are linked	R/W
b2	ITA4VE	TCIV4 Interrupt Skipping Link Enable*1, *2, *3	0: A/D converter start request signal TRG4AN and TCI4V interrupt skipping 1 are not linked 1: A/D converter start request signal TRG4AN and TCI4V interrupt skipping 1 are linked	R/W
b3	ITA3AE	TGIA3 Interrupt Skipping Link Enable*1, *2, *3	0: A/D converter start request signal TRG4AN and TGI3A interrupt skipping 1 are not linked 1: A/D converter start request signal TRG4AN and TGI3A interrupt skipping 1 are linked	R/W
b4	DT4BE	Down-Count TRG4BN Enable*3	0: A/D converter start requests (TRG4BN) disabled during MTU4.TCNT down-count operation 1: A/D converter start requests (TRG4BN) enabled during MTU4.TCNT down-count operation	R/W
b5	UT4BE	Up-Count TRG4BN Enable	0: A/D converter start requests (TRG4BN) disabled during MTU4.TCNT up-count operation 1: A/D converter start requests (TRG4BN) enabled during MTU4.TCNT up-count operation	R/W
b6	DT4AE	Down-Count TRG4AN Enable*3	0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT down-count operation 1: A/D converter start requests (TRG4AN) enabled during MTU4.TCNT down-count operation	R/W
b7	UT4AE	Up-Count TRG4AN Enable	0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT up-count operation 1: A/D converter up requests (TRG4AN) enabled during MTU4.TCNT down-count operation	R/W
b13 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	BF[1:0]	MTU4.TADCOBRA/TADCOBRB Transfer Timing Select	Refer to Table 24.53 for details. These bits specify the transfer timing from MTU4.TADCOBRA and MTU4.TADCOBRB to MTU4.TADCORA and MTU4.TADCORB.	R/W

Note: MTU4.TADCR must not be accessed in 8 bits; it should be accessed in 16 bits.
Note 1. Set to 0 when interrupt skipping is disabled (the T3AEN and T4VEN bits in TITCR1A are cleared to 0 or the T3ACOR and T4VCOR bits in TITCR1A are cleared to 0).
Note 2. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.
Note 3. Set to 0 when complementary PWM mode is not selected.

TADCR enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping function. The MTU has one TADCR each for MTU4 and MTU7.

Table 24.53 Setting of Transfer Timing by TADCR.BF[1:0] Bits (MTU4)

Bit 15	Bit 14	Description			
BF[1]	**BF[0]**	**In Complementary PWM Mode**	**In Reset-Synchronized PWM Mode**	**In PWM Mode 1**	**In Normal Mode**
0	0	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).
0	1	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) at the crest of the MTU4.TCNT.	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when a compare match occurs between MTU3.TCNT and MTU3.TGRA.	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when a compare match occurs between MTU4.TCNT and MTU4.TGRA.	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when a compare match occurs between MTU4.TCNT and MTU4.TGRA.
1	0	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) at the trough of the MTU4.TCNT.	Setting prohibited	Setting prohibited	Setting prohibited
1	1	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) at the crest and trough of the MTU4.TCNT.	Setting prohibited	Setting prohibited	Setting prohibited

- MTU7.TADCR

Address(es): MTU7.TADCR 000C 1A40h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BF[1:0]		—	—	—	—	—	—	UT7AE	DT7AE	UT7BE	DT7BE	ITA6AE	ITA7VE	ITB6AE	ITB7VE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ITB7VE	TCIV7 Interrupt Skipping Link Enable*1, *2, *3	0: A/D converter start request signal TRG7BN and TCI7V interrupt skipping 1 are not linked 1: A/D converter start request signal TRG7BN and TCI7V interrupt skipping 1 are linked	R/W
b1	ITB6AE	TGIA6 Interrupt Skipping Link Enable*1, *2, *3	0: A/D converter start request signal TRG7BN and TGI6A interrupt skipping 1 are not linked 1: A/D converter start request signal TRG7BN and TGI6A interrupt skipping 1 are linked	R/W
b2	ITA7VE	TCIV7 Interrupt Skipping Link Enable*1, *2, *3	0: A/D converter start request signal TRG7AN and TCI7V interrupt skipping 1 are not linked 1: A/D converter start request signal TRG7AN and TCI7V interrupt skipping 1 are linked	R/W
b3	ITA6AE	TGIA6 Interrupt Skipping Link Enable*1, *2, *3	0: A/D converter start request signal TRG7AN and TGI6A interrupt skipping 1 are not linked 1: A/D converter start request signal TRG7AN and TGI6A interrupt skipping 1 are linked	R/W
b4	DT7BE	Down-Count TRG7BN Enable*3	0: A/D converter start requests (TRG7BN) disabled during MTU7.TCNT down-count operation 1: A/D converter start requests (TRG7BN) enabled during MTU7.TCNT down-count operation	R/W
b5	UT7BE	Up-Count TRG7BN Enable	0: A/D converter start requests (TRG7BN) disabled during MTU7.TCNT up-count operation 1: A/D converter start requests (TRG7BN) enabled during MTU7.TCNT up-count operation	R/W
b6	DT7AE	Down-Count TRG7AN Enable*3	0: A/D converter start requests (TRG7AN) disabled during MTU7.TCNT down-count operation 1: A/D converter start requests (TRG7AN) enabled during MTU7.TCNT down-count operation	R/W
b7	UT7AE	Up-Count TRG7AN Enable	0: A/D converter start requests (TRG7AN) disabled during MTU7.TCNT up-count operation 1: A/D converter up requests (TRG7AN) enabled during MTU7.TCNT down-count operation	R/W
b13 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	BF[1:0]	MTU7.TADCOBRA/TADCOBRB Transfer Timing Select	Refer to Table 24.54 for details. These bits specify the transfer timing from MTU7.TADCOBRA and MTU7.TADCOBRB to MTU7.TADCORA and MTU7.TADCORB.	R/W

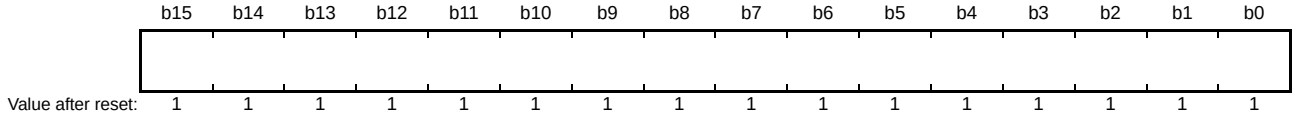
Note: MTU7.TADCR must not be accessed in 8 bits; it should be accessed in 16 bits.
Note 1. Set to 0 when interrupt skipping is disabled (the T6AEN and T7VEN bits in TITCR1B are cleared to 0 or the T6ACOR and T7VCOR bits in TITCR1B are cleared to 0).
Note 2. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.
Note 3. Set to 0 when complementary PWM mode is not selected.

Table 24.54 Setting of Transfer Timing by TADCR.BF[1:0] Bits (MTU7)

Bit 15	Bit 14	Description			
BF[1]	**BF[0]**	**In Complementary PWM Mode**	**In Reset-Synchronized PWM Mode**	**In PWM Mode 1**	**In Normal Mode**
0	0	Data is not transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB).	Data is not transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB).	Data is not transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB).	Data is not transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB).
0	1	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) at the crest of the MTU7.TCNT.	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) when a compare match occurs between MTU6.TCNT and MTU6.TGRA.	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) when a compare match occurs between MTU7.TCNT and MTU7.TGRA.	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) when a compare match occurs between MTU7.TCNT and MTU7.TGRA.
1	0	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) at the trough of the MTU7.TCNT.	Setting prohibited	Setting prohibited	Setting prohibited
1	1	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) at the crest and trough of the MTU7.TCNT.	Setting prohibited	Setting prohibited	Setting prohibited

24.2.36 Timer A/D Converter Start Request Cycle Set Registers (TADCORA, TADCORB)

Address(es): MTU4.TADCORA 000C 1244h, MTU4.TADCORB 000C 1246h, MTU7.TADCORA 000C 1A44h, MTU7.TADCORB 000C 1A46h



Note: TADCORA and TADCORB must not be accessed in 8 bits; it should be accessed in 16 bits.

Note 1. When the A/D converter start request delaying function linked with skipping function 1 (for details, refer to section 24.3.9 (4), A/D Converter Start Request Delaying Function Linked with Interrupt Skipping Function 1) is used, the value of this register should be 0002h to TCDRA setting – 2 in MTU4 and 0002h to TCDRB setting – 2 in MTU7.

Note 2. When interrupt skipping function 2 is used and the difference between the TADCORA value and the TADCORB value is small, the skipping count may not be counted correctly and the A/D converter start request may not be generated with the expected timing in some cases. The TADCORA and TADCORB values should satisfy the following conditions.

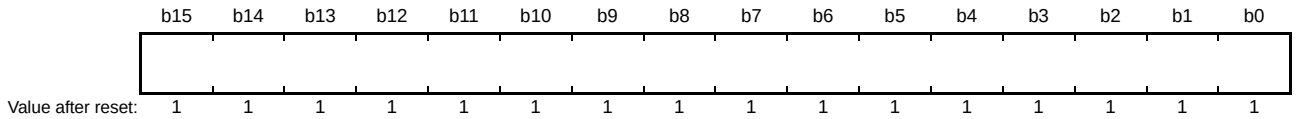
(1) When skipping function 2 is specified with the skipping count set to 0
- The difference between the TADCORA and TADCORB values should be equal to or greater than 4.
- The TADCORA compare interval should be equal to or greater than 4 PCLKA cycles (the TADCORA update value should be the previous value + 4 or greater, or previous value – 4 or smaller).
- The TADCORB compare interval should be equal to or greater than 4 PCLKA cycles (the TADCORB update value should be the previous value + 4 or greater, or previous value – 4 or smaller).

(2) When skipping function 2 is specified with the skipping count set to 1 or greater
- The difference between the TADCORA and TADCORB values should be equal to or greater than 2.
- The TADCORB compare interval should be equal to or greater than 2 PCLKA cycles (the TADCORB update value should be the previous value + 2 or greater, or previous value – 2 or smaller)

TADCORA and TADCORB are 16-bit readable/writable registers that issue a corresponding A/D converter start request when the MTUn.TCNT (n = 4, 7) count reaches the value in TADCORA or TADCORB.
MTUn.TADCORA and TADCORB are initialized to FFFFh by a reset.

24.2.37 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOBRA, TADCOBRB)

Address(es): MTU4.TADCOBRA 000C 1248h, MTU4.TADCOBRB 000C 124Ah, MTU7.TADCOBRA 000C 1A48h, MTU7.TADCOBRB 000C 1A4Ah



Note: TADCOBRA and TADCOBRB must not be accessed in 8 bits; it should be accessed in 16 bits.

TADCOBRA and TADCOBRB are 16-bit readable/writable registers whose values are transferred to TADCORA and TADCORB, respectively, when the crest or trough of the MTUn.TCNT count is reached.
TADCOBRA and TADCOBRB are initialized to FFFFh by a reset.

24.2.38 Timer Interrupt Skipping Mode Registers (TITMRA, TITMRB)

Address(es): MTU.TITMRA 000C 123Ah, MTU.TITMRB 000C 1A3Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TITM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TITM	Interrupt Skipping Function Select	Selects one of the two types of interrupt skipping functions shown in Table 24.55.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TITMRA and TITMRB are used to select either of two skipping functions for the TITMRA and TITMRB registers.

Table 24.55 Interrupt Skipping Function Selected through TITM Bit

Bit 0	
TITM	**Description**
0	Selects interrupt skipping function 1*1
1	Selects interrupt skipping function 2*2

Note 1. TITCR1A or TITCR1B enables interrupt skipping function 1.
Note 2. TITCR2A or TITCR2B enables interrupt skipping function 2.

24.2.39 Timer Interrupt Skipping Set Registers 1 (TITCR1A, TITCR1B)

- MTU.TITCR1A

Address(es): MTU.TITCR1A 000C 1230h

	b7	b6	b5	b4	b3	b2	b1	b0
	T3AEN	T3ACOR[2:0]			T4VEN	T4VCOR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T4VCOR[2:0]	TCIV4 Interrupt Skipping Count Setting	These bits specify the TCIV4 interrupt skipping count within the range from 0 to 7.*1 For details, refer to Table 24.56.	R/W
b3	T4VEN	T4VEN	0: TCIV4 interrupt skipping disabled 1: TCIV4 interrupt skipping enabled	R/W
b6 to b4	T3ACOR[2:0]	TGIA3 Interrupt Skipping Count Setting	These bits specify the TGIA3 interrupt skipping count within the range from 0 to 7.*1 For details, refer to Table 24.57.	R/W
b7	T3AEN	T3AEN	0: TGIA3 interrupt skipping disabled 1: TGIA3 interrupt skipping enabled	R/W

Note 1. When 0 is specified for the interrupt skipping count, no interrupt skipping will be performed.
Before changing the interrupt skipping count, be sure to clear the TITCR1A.T3AEN and TITCR1A.T4VEN bits to 0 to clear the skipping counter (TITCNT1A).

- MTU.TITCR1B

Address(es): MTU.TITCR1B 000C 1A30h

	b7	b6	b5	b4	b3	b2	b1	b0
	T6AEN	T6ACOR[2:0]			T7VEN	T7VCOR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T7VCOR[2:0]	TCIV7 Interrupt Skipping Count Setting	These bits specify the TCIV7 interrupt skipping count within the range from 0 to 7.*1 For details, refer to Table 24.58.	R/W
b3	T7VEN	T7VEN	0: TCIV7 interrupt skipping disabled 1: TCIV7 interrupt skipping enabled	R/W
b6 to b4	T6ACOR[2:0]	TGIA6 Interrupt Skipping Count Setting	These bits specify the TGIA6 interrupt skipping count within the range from 0 to 7.*1 For details, refer to Table 24.59.	R/W
b7	T6AEN	T6AEN	0: TGIA6 interrupt skipping disabled 1: TGIA6 interrupt skipping enabled	R/W

Note 1. When 0 is specified for the interrupt skipping count, no interrupt skipping will be performed.
Before changing the interrupt skipping count, be sure to clear the TITCR1B.T6AEN and TITCR1B.T7VEN bits to 0 to clear the skipping counter (TITCNT1B).

Registers TITCR1A and TITCR1B enable or disable interrupt skipping and specify the interrupt skipping count. This setting is valid only while the TITMRA.TITM or TITMRB.TITM bit is set to 0; when the TITMRA.TITM (TITMRB.TITM) bit is set to 1, the setting in the TITCR1A (TITCR1B) register is cleared.

Table 24.56 Setting of Interrupt Skipping Count by T4VCOR[2:0] Bits

Bit 2	Bit 1	Bit 0	
T4VCOR[2]	T4VCOR[1]	T4VCOR[0]	Description
0	0	0	Does not skip TCIV4 interrupts.
0	0	1	Sets the TCIV4 interrupt skipping count to 1.
0	1	0	Sets the TCIV4 interrupt skipping count to 2.
0	1	1	Sets the TCIV4 interrupt skipping count to 3.
1	0	0	Sets the TCIV4 interrupt skipping count to 4.
1	0	1	Sets the TCIV4 interrupt skipping count to 5.
1	1	0	Sets the TCIV4 interrupt skipping count to 6.
1	1	1	Sets the TCIV4 interrupt skipping count to 7.

Table 24.57 Setting of Interrupt Skipping Count by T3ACOR[2:0] Bits

Bit 6	Bit 5	Bit 4	
T3ACOR[2]	T3ACOR[1]	T3ACOR[0]	Description
0	0	0	Does not skip TGIA3 interrupts.
0	0	1	Sets the TGIA3 interrupt skipping count to 1.
0	1	0	Sets the TGIA3 interrupt skipping count to 2.
0	1	1	Sets the TGIA3 interrupt skipping count to 3.
1	0	0	Sets the TGIA3 interrupt skipping count to 4.
1	0	1	Sets the TGIA3 interrupt skipping count to 5.
1	1	0	Sets the TGIA3 interrupt skipping count to 6.
1	1	1	Sets the TGIA3 interrupt skipping count to 7.

Table 24.58 Setting of Interrupt Skipping Count by T7VCOR[2:0] Bits

Bit 2	Bit 1	Bit 0	
T7VCOR[2]	T7VCOR[1]	T7VCOR[0]	Description
0	0	0	Does not skip TCIV7 interrupts.
0	0	1	Sets the TCIV7 interrupt skipping count to 1.
0	1	0	Sets the TCIV7 interrupt skipping count to 2.
0	1	1	Sets the TCIV7 interrupt skipping count to 3.
1	0	0	Sets the TCIV7 interrupt skipping count to 4.
1	0	1	Sets the TCIV7 interrupt skipping count to 5.
1	1	0	Sets the TCIV7 interrupt skipping count to 6.
1	1	1	Sets the TCIV7 interrupt skipping count to 7.

Table 24.59 Setting of Interrupt Skipping Count by T6ACOR[2:0] Bits

Bit 6	Bit 5	Bit 4	
T6ACOR[2]	T6ACOR[1]	T6ACOR[0]	Description
0	0	0	Does not skip TGIA6 interrupts.
0	0	1	Sets the TGIA6 interrupt skipping count to 1.
0	1	0	Sets the TGIA6 interrupt skipping count to 2.
0	1	1	Sets the TGIA6 interrupt skipping count to 3.
1	0	0	Sets the TGIA6 interrupt skipping count to 4.
1	0	1	Sets the TGIA6 interrupt skipping count to 5.
1	1	0	Sets the TGIA6 interrupt skipping count to 6.
1	1	1	Sets the TGIA6 interrupt skipping count to 7.

24.2.40 Timer Interrupt Skipping Counters 1 (TITCNT1A, TITCNT1B)

- MTU.TITCNT1A

Address(es): MTU.TITCNT1A 000C 1231h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	T3ACNT[2:0]			—	T4VCNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T4VCNT[2:0]	TCIV4 Interrupt Counter	While the T4VEN bit in TITCR1A is set to 1, the count in these bits is incremented every time a TCIV4 interrupt occurs.	R
b3	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R
b6 to b4	T3ACNT[2:0]	TGIA3 Interrupt Counter	While the T3AEN bit in TITCR1A is set to 1, the count in these bits is incremented every time a TGIA3 interrupt occurs.	R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

Note 1. To clear the TITCNT1A, clear the TITCR1A.T3AEN and TITCR1A.T4VEN bits to 0.

TITCNT1A and TITCNT1B are 8-bit readable/writable counters. TITCNT1A and TITCNT1B retain their values even after stopping the count operation of MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT).

T4VCNT[2:0] Bits (TCIV4 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRA is 1
- When the T4VEN bit in TITCR1A is cleared to 0
- When the T4VCOR[2:0] bits in TITCR1A are cleared to 000b
- When the T4VCNT[2:0] bits in TITCNT1A match the T4VCOR[2:0] bits in TITCR1A

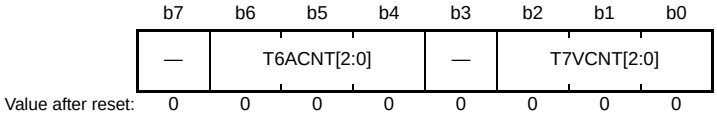
T3ACNT[2:0] Bits (TGIA3 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRA is 1
- When the T3AEN bit in TITCR1A is cleared to 0
- When the T3ACOR[2:0] bits in TITCR1A are cleared to 000b
- When the T3ACNT[2:0] bits in TITCNT1A match the T3ACOR[2:0] bits in TITCR1A

- MTU.TITCNT1B

Address(es): MTU.TITCNT1B 000C 1A31h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T7VCNT[2:0]	TCIV7 Interrupt Counter	While the T7VEN bit in TITCR1B is set to 1, the count in these bits is incremented every time a TCIV7 interrupt occurs.	R
b3	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R
b6 to b4	T6ACNT[2:0]	TGIA6 Interrupt Counter	While the T6AEN bit in TITCR1B is set to 1, the count in these bits is incremented every time a TGIA6 interrupt occurs.	R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

Note 1. To clear the TITCNT1B, clear the TITCR1B.T6AEN and TITCR1B.T7VEN bits to 0.

T7VCNT[2:0] Bits (TCIV7 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRB is 1
- When the T7VEN bit in TITCR1B is cleared to 0
- When the T7VCOR[2:0] bits in TITCR1B are cleared to 000b
- When the T7VCNT[2:0] bits in TITCNT1B match the T7VCOR[2:0] bits in TITCR1B

T6ACNT[2:0] Bits (TGIA6 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRB is 1
- When the T6AEN bit in TITCR1B is cleared to 0
- When the T6ACOR[2:0] bits in TITCR1B are cleared to 000b
- When the T6ACNT[2:0] bits in TITCNT1B match the T6ACOR[2:0] bits in TITCR1B

24.2.41 Timer Interrupt Skipping Set Registers 2 (TITCR2A, TITCR2B)

- MTU.TITCR2A

Address(es): MTU.TITCR2A 000C 123Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TRG4COR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TRG4COR[2:0]	TRG4AN/TRG4BN Interrupt Skipping Count Setting	These bits specify the TRG4AN/TRG4BN interrupt skipping count within the range from 0 to 7. For details, refer to Table 24.60.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TITCR2A and TITCR2B specify the interrupt skipping count for TRG4AN and TRG4BN (TRG7AN and TRG7BN). This setting is valid only while TITMRA or TITMRB is set to 1.

Table 24.60 Setting of Interrupt Skipping Count by TRG4COR[2:0] Bits

Bit 2	Bit 1	Bit 0	
TRG4COR[2]	TRG4COR[1]	TRG4COR[0]	Description
0	0	0	Does not skip TRG4AN and TRG4BN interrupts.
0	0	1	Sets the TRG4AN and TRG4BN interrupt skipping count to 1.
0	1	0	Sets the TRG4AN and TRG4BN interrupt skipping count to 2.
0	1	1	Sets the TRG4AN and TRG4BN interrupt skipping count to 3.
1	0	0	Sets the TRG4AN and TRG4BN interrupt skipping count to 4.
1	0	1	Sets the TRG4AN and TRG4BN interrupt skipping count to 5.
1	1	0	Sets the TRG4AN and TRG4BN interrupt skipping count to 6.
1	1	1	Sets the TRG4AN and TRG4BN interrupt skipping count to 7.

- MTU.TITCR2B

Address(es): MTU.TITCR2B 000C 1A3Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TRG7COR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TRG7COR[2:0]	TRG7AN/TRG7BN Interrupt Skipping Count Setting	These bits specify the TRG7AN/TRG7BN interrupt skipping count within the range from 0 to 7. For details, refer to Table 24.61.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 24.61 Setting of Interrupt Skipping Count by TRG7COR[2:0] Bits

Bit 2	Bit 1	Bit 0	
TRG7COR[2]	TRG7COR[1]	TRG7COR[0]	Description
0	0	0	Does not skip TRG7AN and TRG7BN interrupts.
0	0	1	Sets the TRG7AN and TRG7BN interrupt skipping count to 1.
0	1	0	Sets the TRG7AN and TRG7BN interrupt skipping count to 2.
0	1	1	Sets the TRG7AN and TRG7BN interrupt skipping count to 3.
1	0	0	Sets the TRG7AN and TRG7BN interrupt skipping count to 4.
1	0	1	Sets the TRG7AN and TRG7BN interrupt skipping count to 5.
1	1	0	Sets the TRG7AN and TRG7BN interrupt skipping count to 6.
1	1	1	Sets the TRG7AN and TRG7BN interrupt skipping count to 7.

24.2.42 Timer Interrupt Skipping Counters 2 (TITCNT2A, TITCNT2B)

- MTU.TITCNT2A

Address(es): MTU.TITCNT2A 000C 123Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TRG4CNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TRG4CNT[2:0]	TRG4AN/TRG4BN Interrupt Counter	These bits start counting from the value set in TRG4COR[2:0] and the count decrements every time TRG4AN or TRG4BN is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid.	R
b7 to b3	—	Reserved	These bits are read as 0. Writing to these bit has no effect.	R

TITCNT2A and TITCNT2B start counting from the values set in the TRG4COR[2:0] and TRG7COR[2:0] bits and the count decrements every time TRG4AN or TRG4BN (TITCNT2A) is generated or TRG7AN or TRG7BN (TITCNT2B) is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts or the TRG7AN and TRG7BN interrupts become valid.

TRG4CNT[2:0] Bits (TRG4AN/TRG4BN Interrupt Counter)

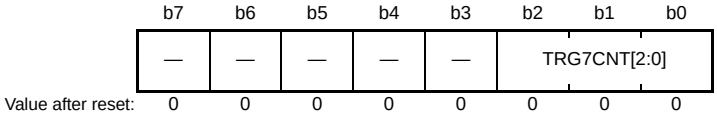
These bits start counting from the value set in the TRG4COR[2:0] bits and the count decrements every time a TRG4AN or TRG4BN interrupt is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid.

[Clearing conditions]

- When the TITM bit in TITMRA is 0
- When the TRG4COR[2:0] bits in TITCR2A are cleared to 000b
- When the count of TRG4AN and TRG4BN occurrence matches the TRG4COR[2:0] value in TITCR2A

- MTU.TITCNT2B

Address(es): MTU.TITCNT2B 000C 1A3Ch



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TRG7CNT[2:0]	TRG7AN/TRG7BN Interrupt Counter	These bits start counting from the value set in TRG7COR[2:0] and the count decrements every time TRG7AN or TRG7BN is generated. When the count reaches 0 and is reloaded, the TRG7AN and TRG7BN interrupts become valid.	R
b7 to b3	—	Reserved	These bits are read as 0. Writing to these bit has no effect.	R

TRG7CNT[2:0] Bits (TRG7AN/TRG7BN Interrupt Counter)

These bits start counting from the value set in the TRG7COR[2:0] bits and the count decrements every time a TRG7AN or TRG7BN interrupt is generated. When the count reaches 0 and is reloaded, the TRG7AN and TRG7BN interrupts become valid.

[Clearing conditions]

- When the TITM bit in TITMRB is 0
- When the TRG7COR[2:0] bits in TITCR2B are cleared to 000b
- When the count of TRG7AN and TRG7BN occurrence matches the TRG7COR[2:0] value in TITCR2B

24.3 Operation

24.3.1 Basic Functions

Each channel has TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.
Each TGR register can be used as an input capture register or an output compare register.

(1) Counter Operation

When one of bits CST0 to CST4 and CST8 in the TSTRA register, bits CST6 and CST7 in the TSTRB register, and bits CSTU5, CSTV5, and CSTW5 in the MTU5.TSTR register is set to 1, TCNT for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example.

(a) Example of Count Operation Setting Procedure

Figure 24.5 shows an example of the count operation setting procedure.

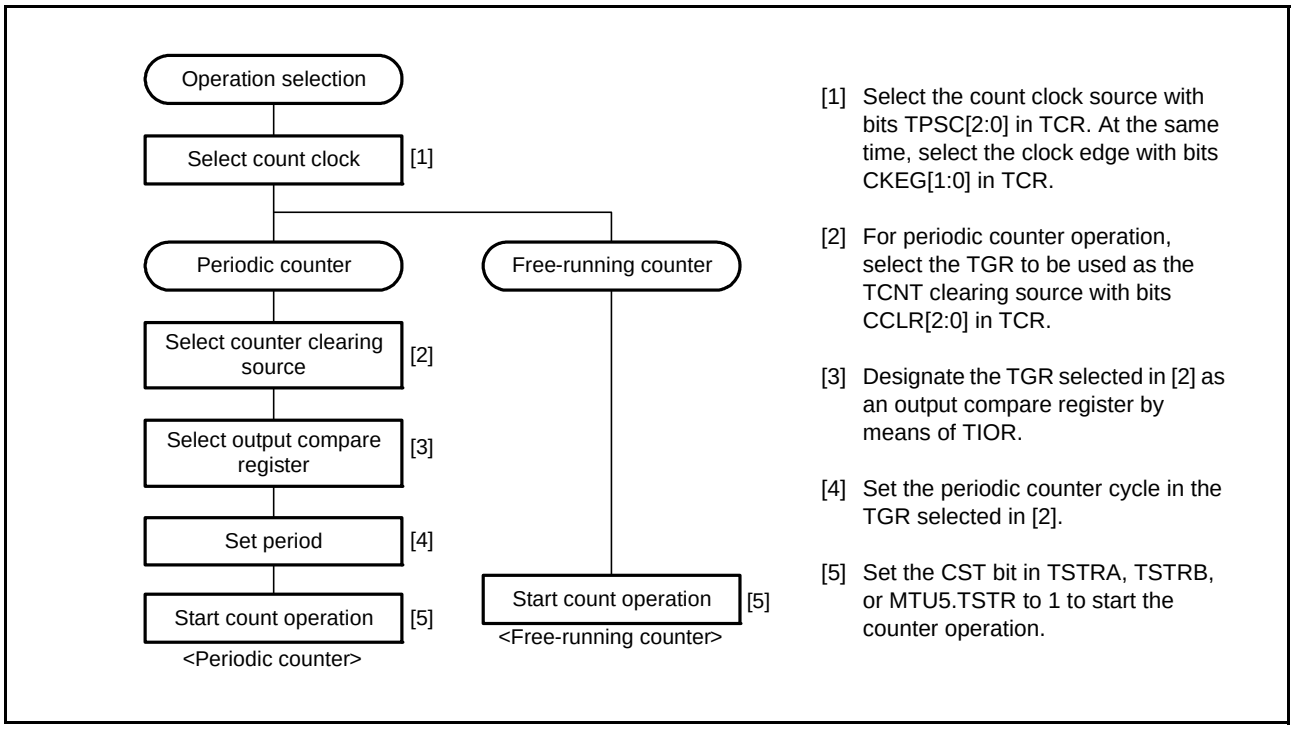


Figure 24.5 Example of Counter Operation Setting Procedure

(b) Free-Running Count Operation and Periodic Count Operation

Immediately after a reset, the TCNT counters are all designated as free-running counters. When the relevant bit in TSTRA, TSTRB, or MTU5.TSTR is set to 1, the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from FFFFh to 0000h), an interrupt request is issued to the CPU if the corresponding TIER.TCIEV bit is 1. After an overflow, TCNT starts counting up again from 0000h.
Figure 24.6 illustrates free-running counter operation.

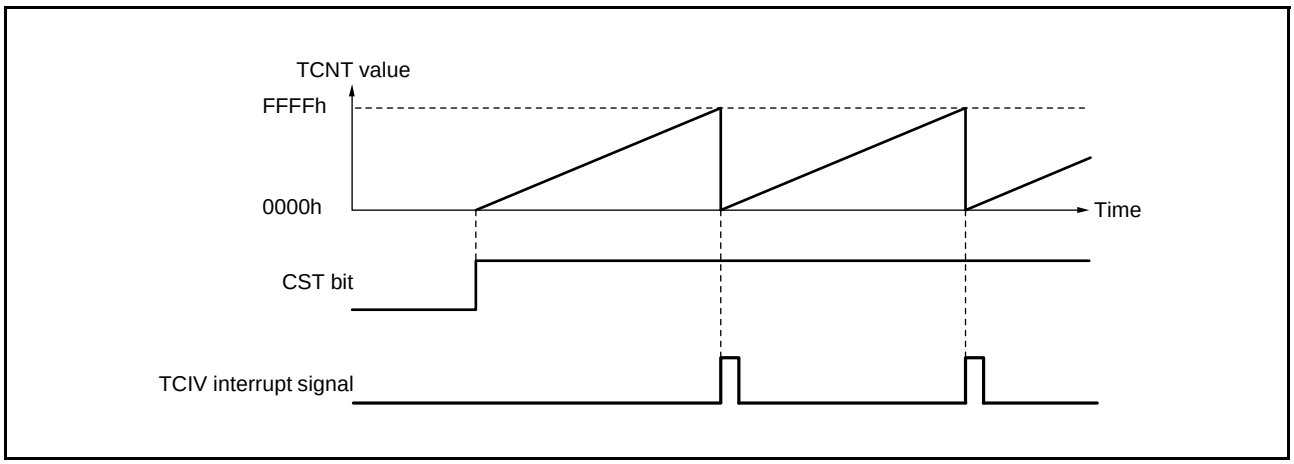


Figure 24.6 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, TCNT for the relevant channel performs periodic count operation. TGR for setting the cycle is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR[2:0] in TCR. After the settings have been made, TCNT starts up-count operation as a periodic counter when the corresponding bit in TSTRA, TSTRB or MTU5.TSTR is set to 1. When the count matches the value in TGR, TCNT is cleared to 0000h.
If the value of the corresponding TIER.TGIE bit is 1 at this point, an interrupt request is issued to the CPU. After a compare match, TCNT starts counting up again from 0000h.
Figure 24.7 illustrates periodic counter operation.

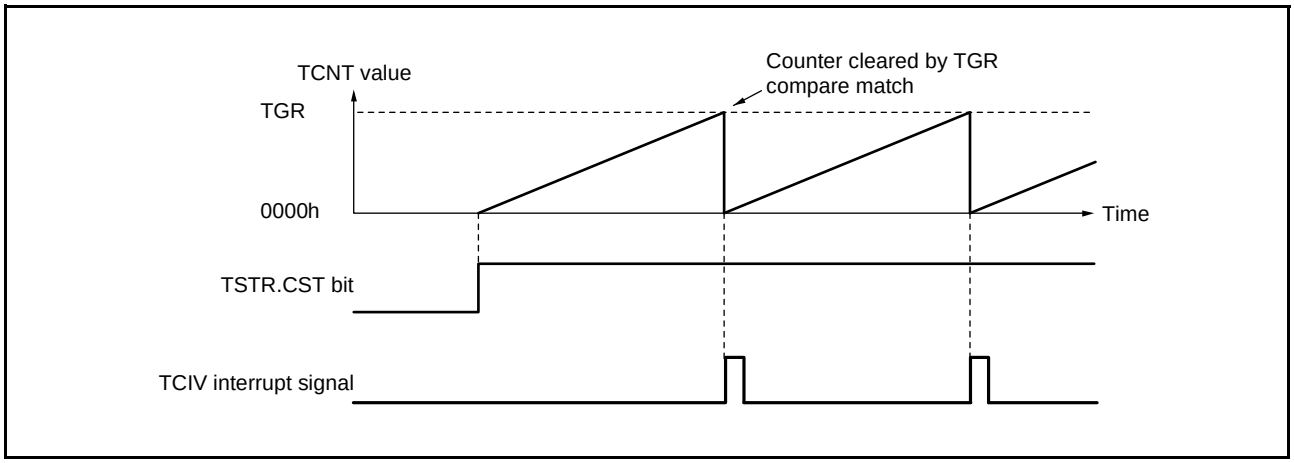


Figure 24.7 Periodic Counter Operation

(2) Waveform Output by Compare Match

Upon compare match, low, high, or toggle output from the corresponding pin can be performed.

(a) Example of Procedure for Setting Waveform Output by Compare Match

Figure 24.8 shows an example of the procedure for setting waveform output by compare match

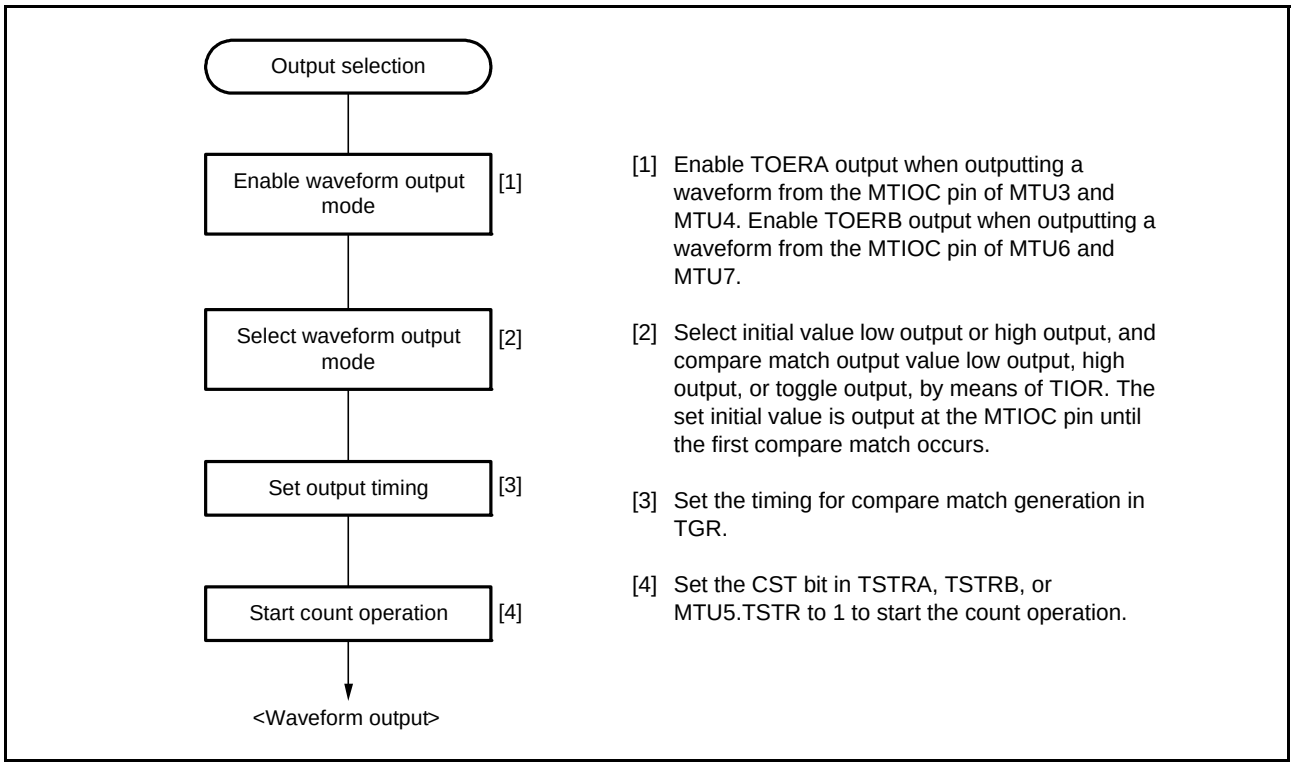


Figure 24.8 Example of Procedure for Setting Waveform Output by Compare Match

(b) Examples of Waveform Output Operation

Figure 24.9 shows an example of low output and high output.
In this example, TCNT has been designated as a free-running counter, and settings have been made so that high is output by compare match A and low is output by compare match B. When the pin level is the same as the specified level, the pin level does not change.

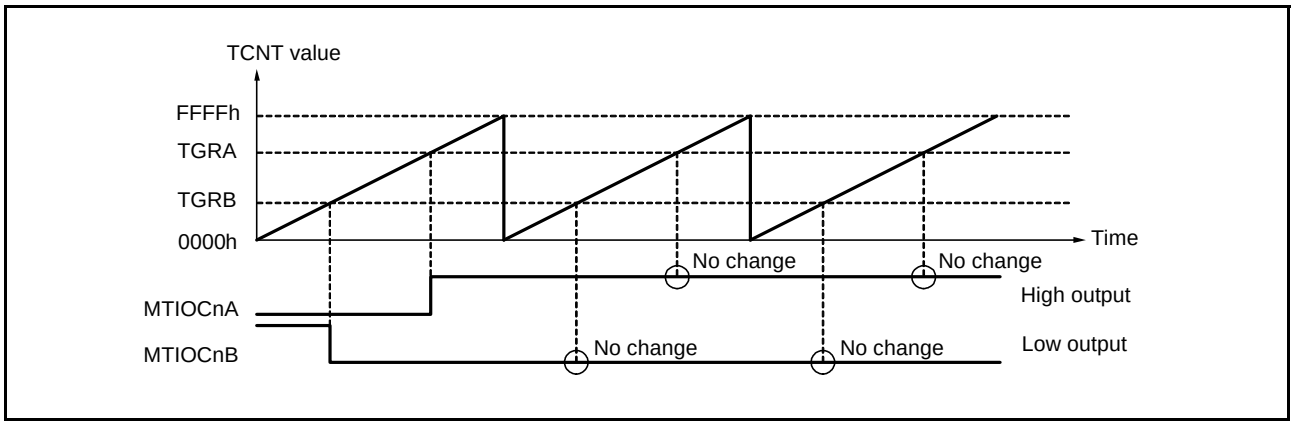


Figure 24.9 Example of low output and high output Operation (n = 0 to 4, 6, 7, 8)

Figure 24.10 shows an example of toggle output.
In this example, TCNT has been designated as a periodic counter (with counter clearing on compare match B), and settings have been made so that the output is toggled by both compare match A and compare match B.

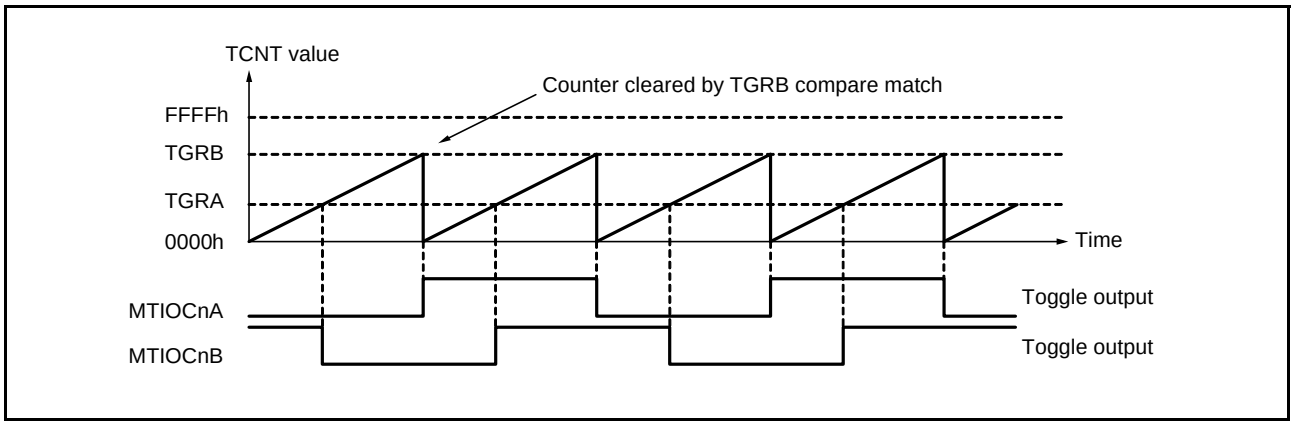


Figure 24.10 Example of Toggle Output Operation (n = 0 to 4, 6, 7, 8)

(3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the MTIOCnm pin (n = 0 to 4, 6, 7, 8; m = A to D) input edge.
The rising edge, falling edge, or both edges can be selected as the detection edge. For MTU0 and MTU1, another channel's count clock or compare match signal can also be specified as the input capture source.

Note: When another channel's count clock is used as the input capture input for MTU0 and MTU1, PCLKA/1 should not be selected as the count clock used for input capture input. Input capture will not be generated if PCLKA/1 is selected.

(a) Example of Input Capture Operation Setting Procedure

Figure 24.11 shows an example of the input capture operation setting procedure.

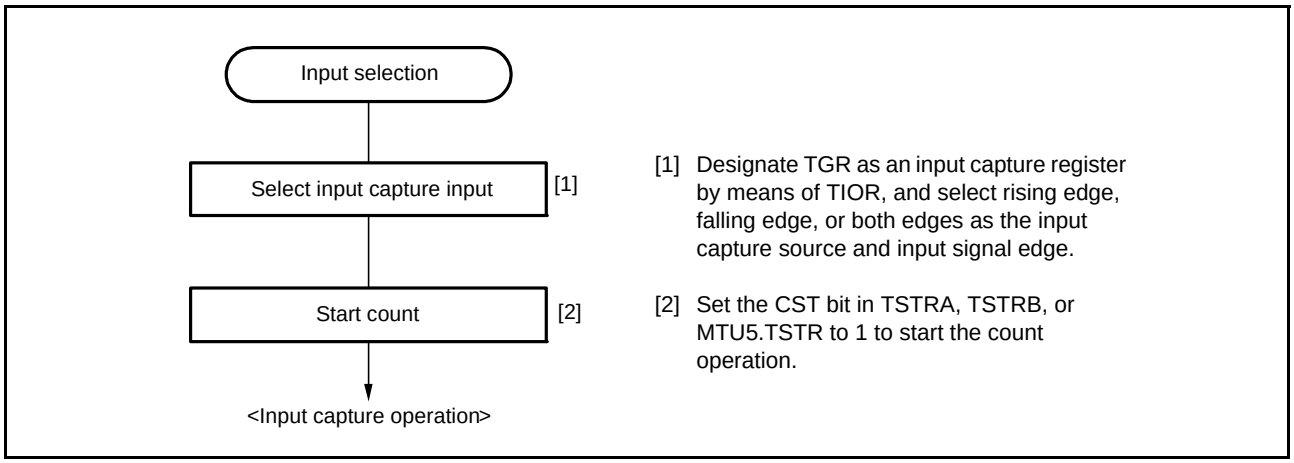


Figure 24.11 Example of Input Capture Operation Setting Procedure

(b) Example of Input Capture Operation

Figure 24.12 shows an example of input capture operation.
In this example, both rising and falling edges have been selected as the MTIOCnA pin input capture input edge, the falling edge has been selected as the MTIOCnB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT. (n = 0 to 4, 6, 7, 8)

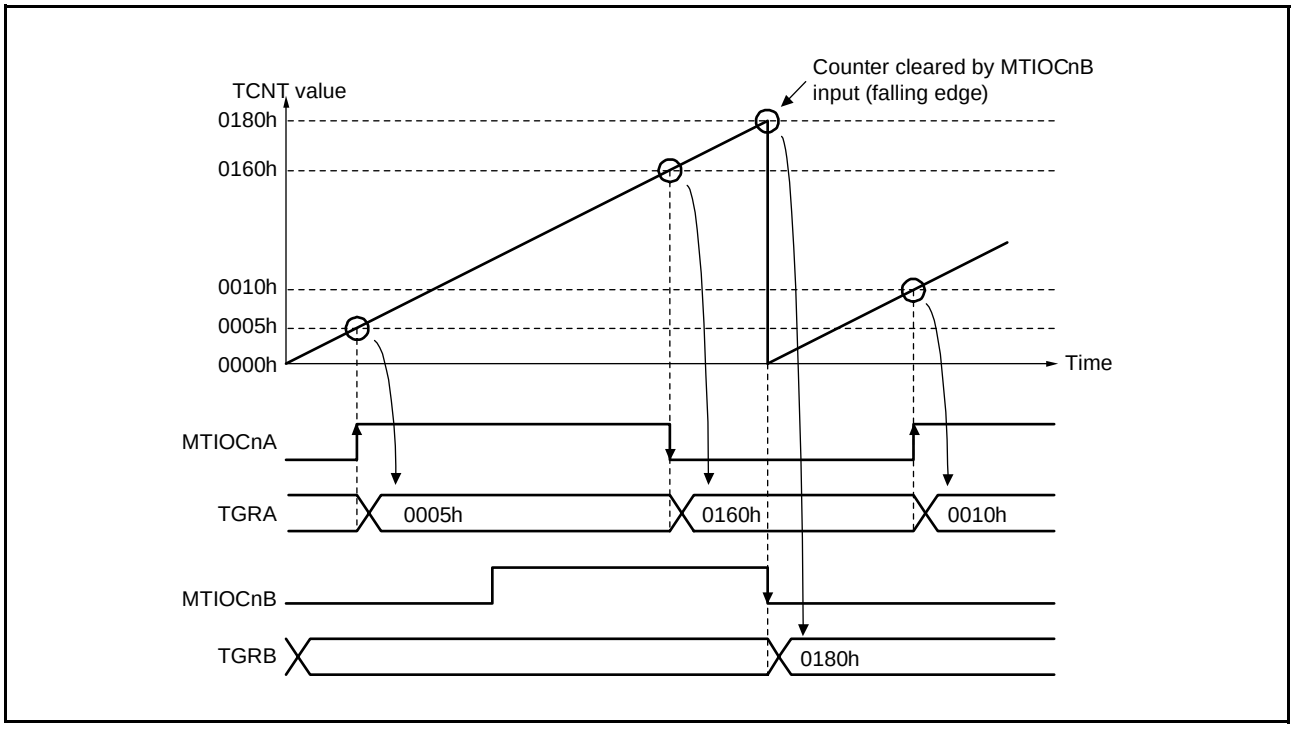


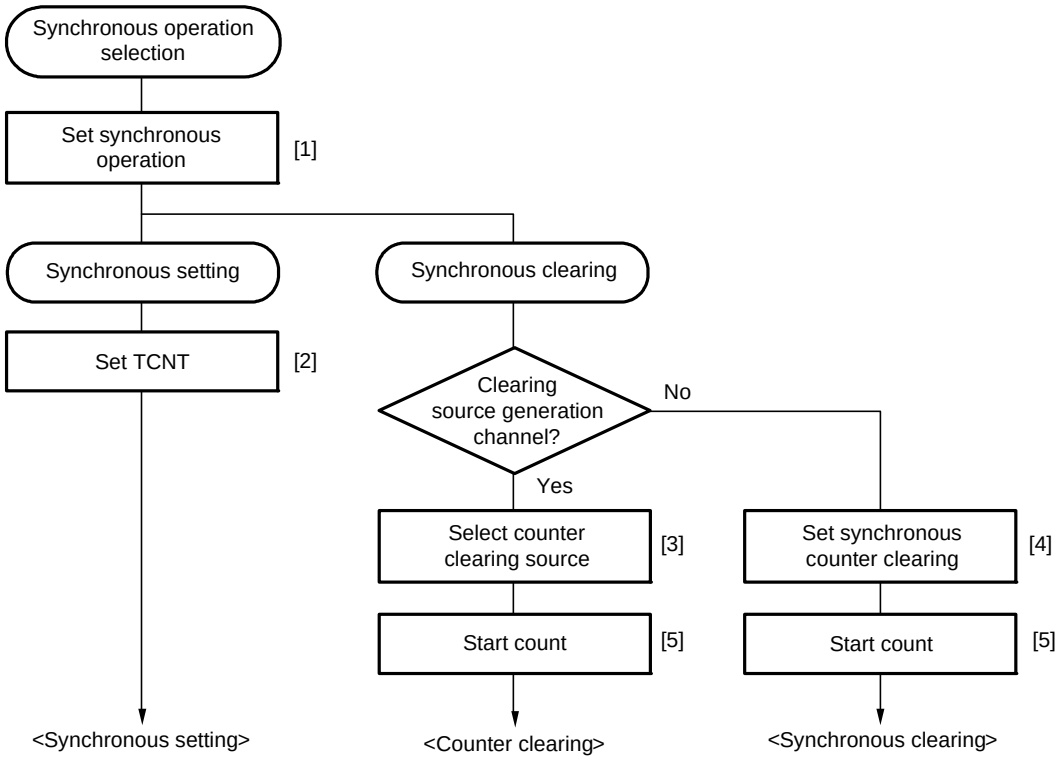
Figure 24.12 Example of Input Capture Operation (n = 0 to 4, 6, 7, 8)

24.3.2 Synchronous Operation

In synchronous operation, the values in multiple TCNT counters can be modified simultaneously (synchronous setting). In addition, multiple TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TCR.
Synchronous operation can increase the number of TGR registers assigned to the same time base.
MTU0 to MTU4, MTU6, and MTU7 can all be designated for synchronous operation. MTU5 and MTU8 cannot be used for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 24.13 shows an example of the synchronous operation setting procedure.



[1] Set to 1 in the SYNC bits in TSYRA or TSYRB corresponding to the channels to be designated for synchronous operation.
[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.
[3] Use the CCLR[2:0] bits in TCR to specify TCNT clearing by input capture/output compare, etc.
[4] Use the CCLR[2:0] bits in TCR to designate synchronous clearing for the counter clearing source.
[5] Set to 1 in the CST bits in TSTRA or TSTRB for the relevant channels, to start the count operation.

Figure 24.13 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 24.14 shows an example of synchronous operation.
In this example, synchronous operation and PWM mode 1 have been designated for MTU0 to MTU 2, MTU0.TGRB compare match has been set as the counter clearing source in MTU0, and synchronous clearing has been set for the counter clearing source in MTU1 and MTU2.
Three-phase PWM waveforms are output from pins MTIOC0A, MTIOC1A, and MTIOC2A. At this time, synchronous setting and synchronous clearing by MTU0.TGRB compare match are performed for the TCNT counters in MTU0 to MTU2, and the data set in MTU0.TGRB is used as the PWM cycle.
For details of PWM modes, refer to section 24.3.5, PWM Modes.

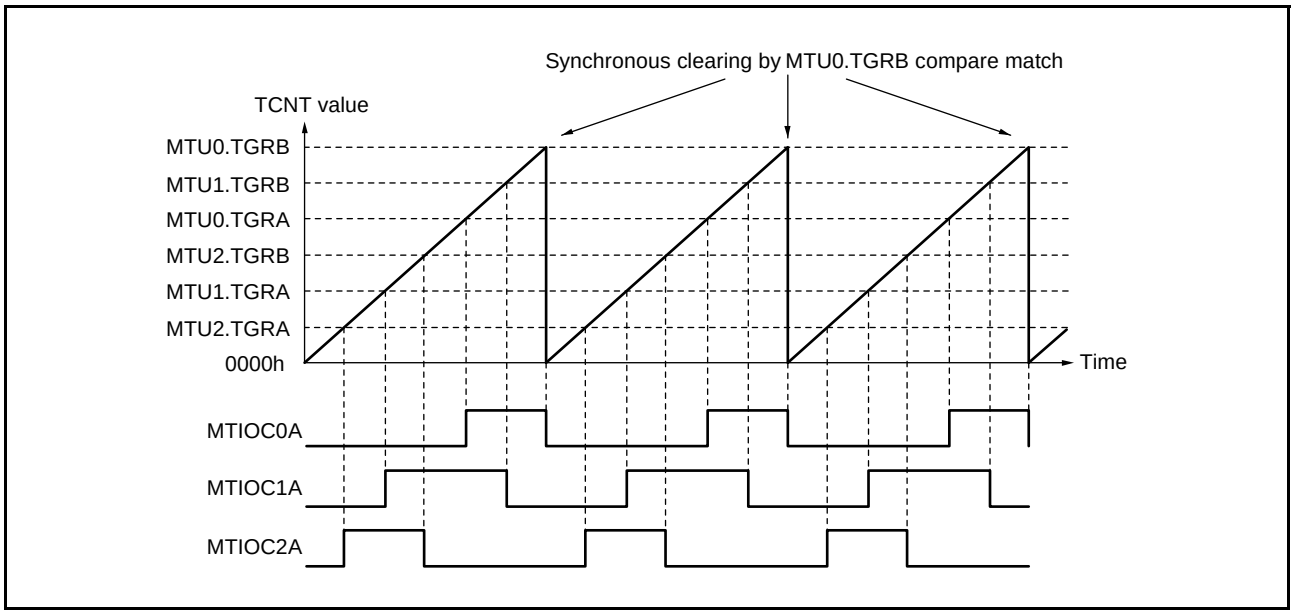


Figure 24.14 Example of Synchronous Operation

24.3.3 Buffer Operation

Buffer operation, provided for MTU0, MTU3, MTU4, MTU6, MTU7, and MTU8, enables TGRC and TGRD to be used as buffer registers. In MTU0, TGRF can also be used as a buffer register.
Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register.

Note: MTU0.TGRE cannot be designated as an input capture register and can only operate as a compare match register.

Table 24.62 shows the register combinations used in buffer operation.

Table 24.62 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
MTU0	TGRA	TGRC
	TGRB	TGRD
	TGRE	TGRF
MTU3	TGRA	TGRC
	TGRB	TGRD
MTU4	TGRA	TGRC
	TGRB	TGRD
MTU6	TGRA	TGRC
	TGRB	TGRD
MTU7	TGRA	TGRC
	TGRB	TGRD
MTU8	TGRA	TGRC
	TGRB	TGRD

- When TGR is an output compare register

When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.
This operation is illustrated in Figure 24.15.

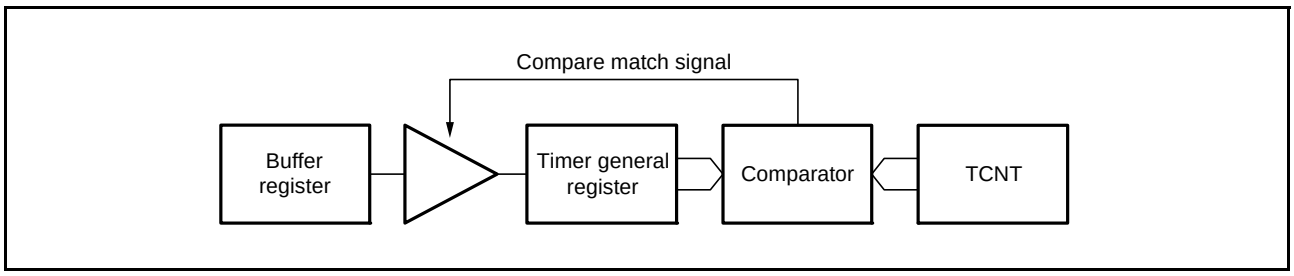


Figure 24.15 Compare Match Buffer Operation

- When TGR is an input capture register

When an input capture occurs, the value in TCNT is transferred to TGR and the value previously held in TGR is transferred to the buffer register.
This operation is illustrated in Figure 24.16.

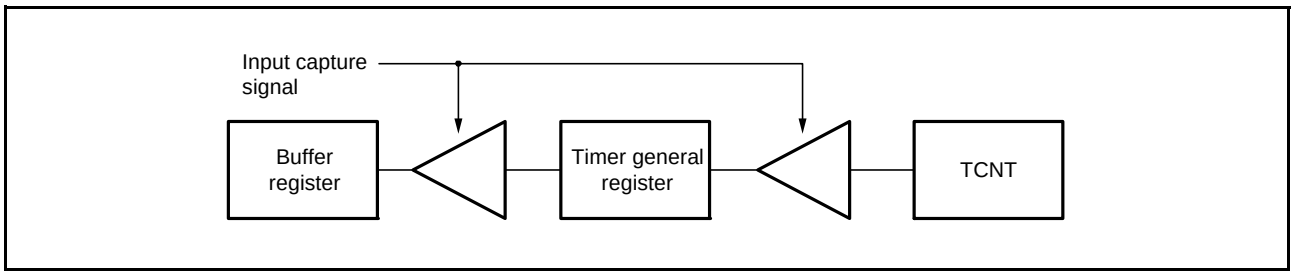


Figure 24.16 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 24.17 shows an example of the buffer operation setting procedure.

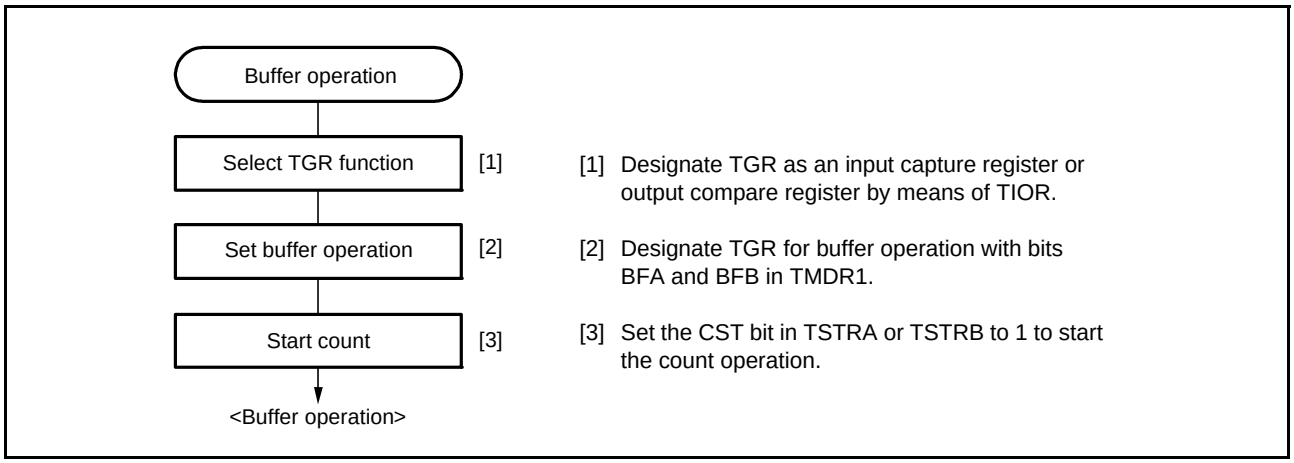


Figure 24.17 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an Output Compare Register

Figure 24.18 shows an operation example in which PWM mode 1 has been designated for MTU0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. In this example, the TTSA bit in TBTM is cleared to 0.
As buffer operation has been set, when compare match A occurs, the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.
For details of PWM modes, refer to section 24.3.5, PWM Modes.

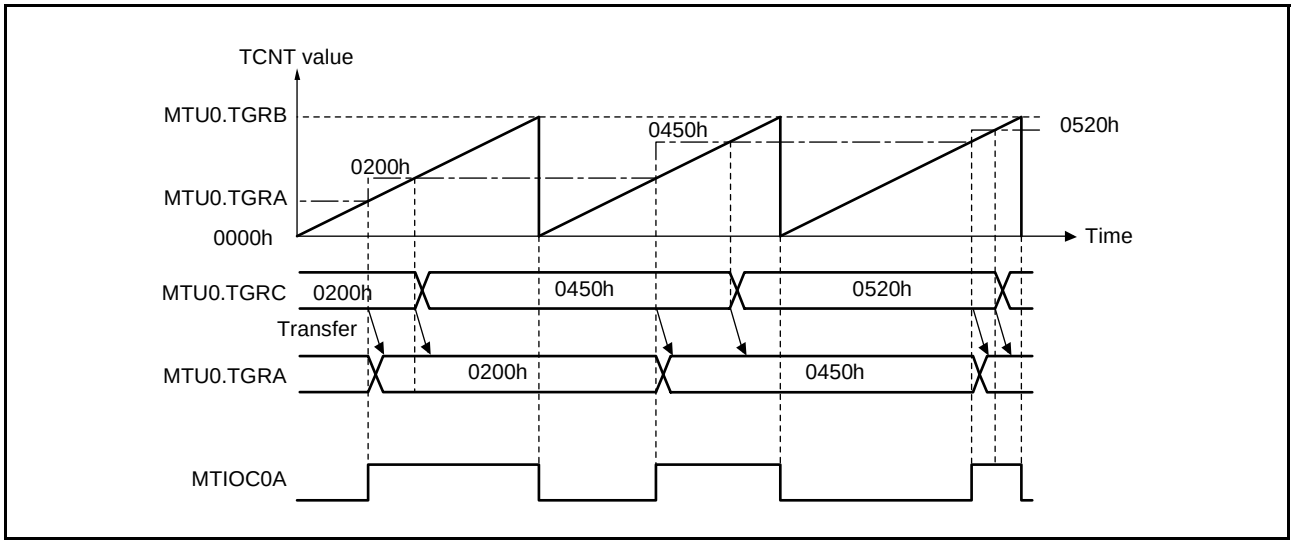


Figure 24.18 Example of Buffer Operation (1)

(b) When TGR is an Input Capture Register

Figure 24.19 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.

Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the MTIOCnA pin input capture input edge. (n = 0 to 4, 6, 7, 8)

As buffer operation has been set, when the TCNT value is transferred to TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

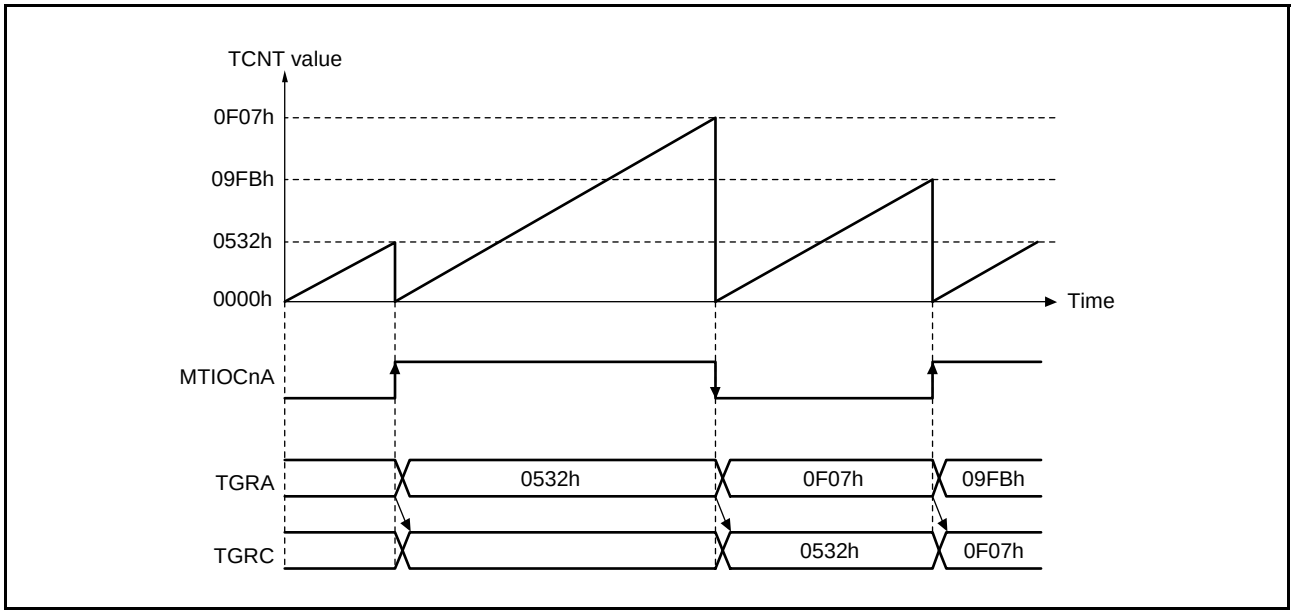


Figure 24.19 Example of Buffer Operation (2) (n = 0 to 4, 6, 7, 8)

(3) Selecting Timing for Transfer from Buffer Registers to Timer General Registers in Buffer Operation

The timing for transfer from buffer registers to timer general registers can be selected in PWM mode 1 or 2 for MTU0 or in PWM mode 1 for MTU3, MTU4, MTU6, and MTU7 by setting the buffer operation transfer mode registers (MTUn.TBTM (n = 0, 3, 4, 6, 7)). Either compare match (value after reset) or TCNT clearing can be selected for the transfer timing. TCNT clearing as transfer timing is one of the following cases.

- When TCNT overflows (FFFFh to 0000h)
- When 0000h is written to TCNT during counting
- When TCNT is cleared to 0000h under the condition specified in the CCLR[2:0] bits in TCR

Note: TBTM must be modified only while TCNT stops.

Figure 24.20 shows an operation example in which PWM mode 1 is designated for MTU0 and buffer operation is designated for MTU0.TGRA and MTU0.TGRC. The settings used in this example are MTU0.TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. The TTSA bit in MTU0.TBTM is set to 1.

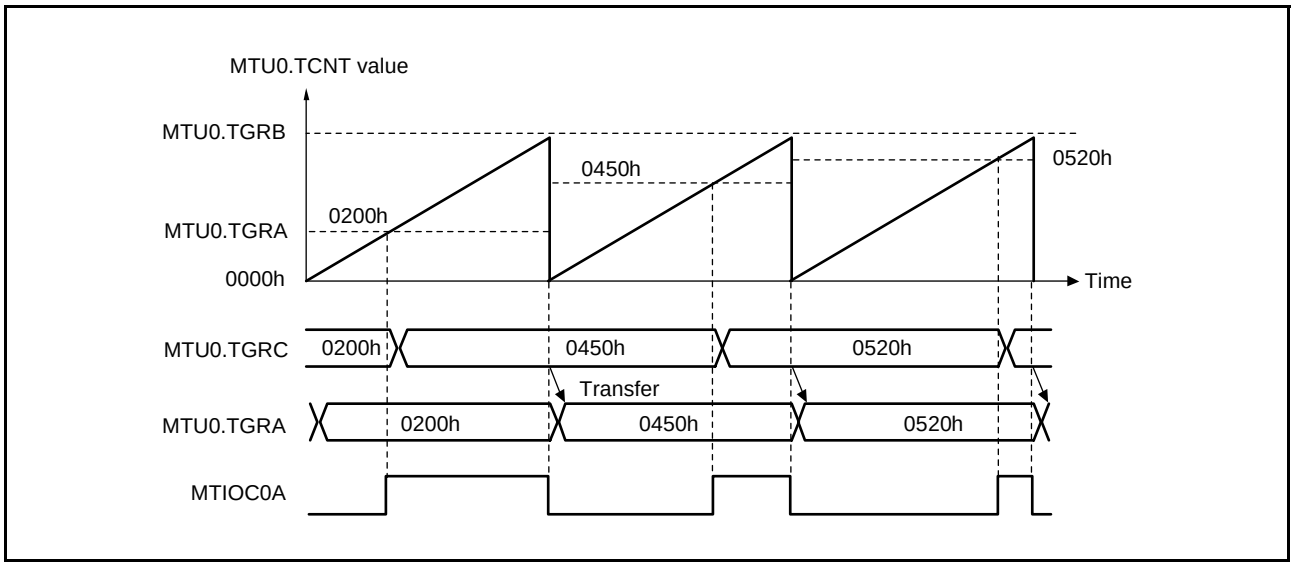


Figure 24.20 Example of Buffer Operation When MTU0.TCNT Clearing is Selected for MTU0.TGRC to MTU0.TGRA Transfer Timing

24.3.4 Cascaded Operation

In cascaded operation, two 16-bit counters in different channels are used together as a 32-bit counter.
There are two functions for connecting MTU1 and MTU2 to use as a 32-bit counter: cascade connection to be set when the MTU1.TMDR3.LWA bit is 0, and cascade connection 32-bit phase counting mode to be set when the MTU1.TMDR3.LWA bit is 1. For details on cascade connection 32-bit phase counting mode, refer to section 24.3.6.2, Cascade Connection 32-Bit Phase Counting Mode. This section describes the cascade connection function to be set when the MTU1.TMDR3.LWA bit is 0.
This function operates when the MTU1.TMDR3.LWA bit is set to 0 and the MTU1.TCR.TPSC[2:0] bits are set so that MTU1.TCNT counts at an overflow/underflow of MTU2.TCNT. Underflow occurs only when the MTU2 to which the lower 16 bits allocated is in phase counting mode.
Table 24.63 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for MTU1, the count clock setting is invalid and the counters operate independently in phase counting mode.

Table 24.63 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
MTU1 and MTU2	MTU1.TCNT	MTU2.TCNT

For simultaneous input capture of MTU1.TCNT and MTU2.TCNT during cascaded operation, additional input capture input pins can be specified by the input capture control register (TICCR). The input-capture condition is of edges in the signal produced by taking the logical OR of the input level on the main input pin and the input level on the added input pin. Accordingly, if either is at the high level, a change in the level of the other will not produce an edge for detection. For details, refer to (4), Cascaded Operation Example (c). For input capture in cascade connection, refer to section 24.6.21, Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection.
Table 24.64 shows the TICCR setting and input capture input pins.

Table 24.64 TICCR Setting and Input Capture Input Pins

Target Input Capture	TICCR Setting	Input Capture Input Pin
Input capture from MTU1.TCNT to MTU1.TGRA	I2AE bit = 0 (Initial value)	MTIOC1A
	I2AE bit = 1	MTIOC1A, MTIOC2A
Input capture from MTU1.TCNT to MTU1.TGRB	I2BE bit = 0 (Initial value)	MTIOC1B
	I2BE bit = 1	MTIOC1B, MTIOC2B
Input capture from MTU2.TCNT to MTU2.TGRA	I1AE bit = 0 (Initial value)	MTIOC2A
	I1AE bit = 1	MTIOC2A, MTIOC1A
Input capture from MTU2.TCNT to MTU2.TGRB	I1BE bit = 0 (Initial value)	MTIOC2B
	I1BE bit = 1	MTIOC2B, MTIOC1B

(1) Example of Cascaded Operation Setting Procedure

Figure 24.21 shows an example of the cascaded operation setting procedure.

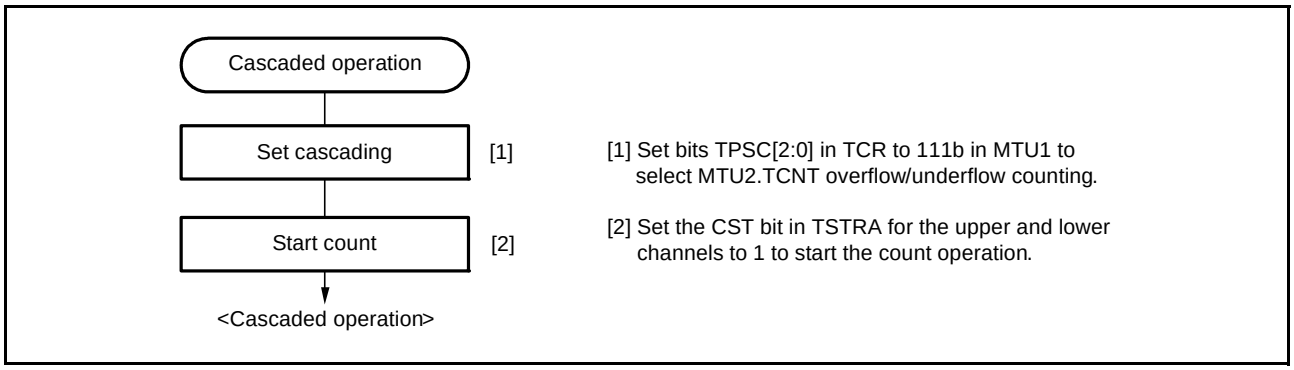


Figure 24.21 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 24.22 shows the operation when MTU1.TCNT is set for counting at MTU2.TCNT overflow/underflow and MTU2 is set for phase counting mode 1 while MTU1.TCNT and MTU2.TCNT are cascaded.
MTU1.TCNT is incremented by MTU2.TCNT overflow and decremented by MTU2.TCNT underflow.

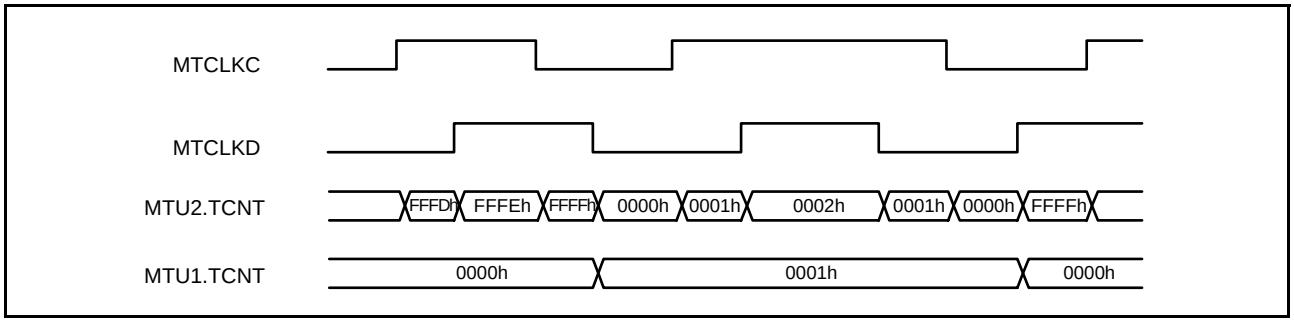


Figure 24.22 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 24.23 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE bit in TICCR has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the MTU1.TIOR.IOA[3:0] bits have selected the MTIOC1A rising edge for the input capture timing while the MTU2.TIOR.IOA[3:0] bits have selected the MTIOC2A rising edge for the input capture timing.
Under these conditions, the rising edge of both MTIOC1A and MTIOC2A is used for the MTU1.TGRA input capture condition. For the MTU2.TGRA input capture condition, the MTIOC2A rising edge is used.

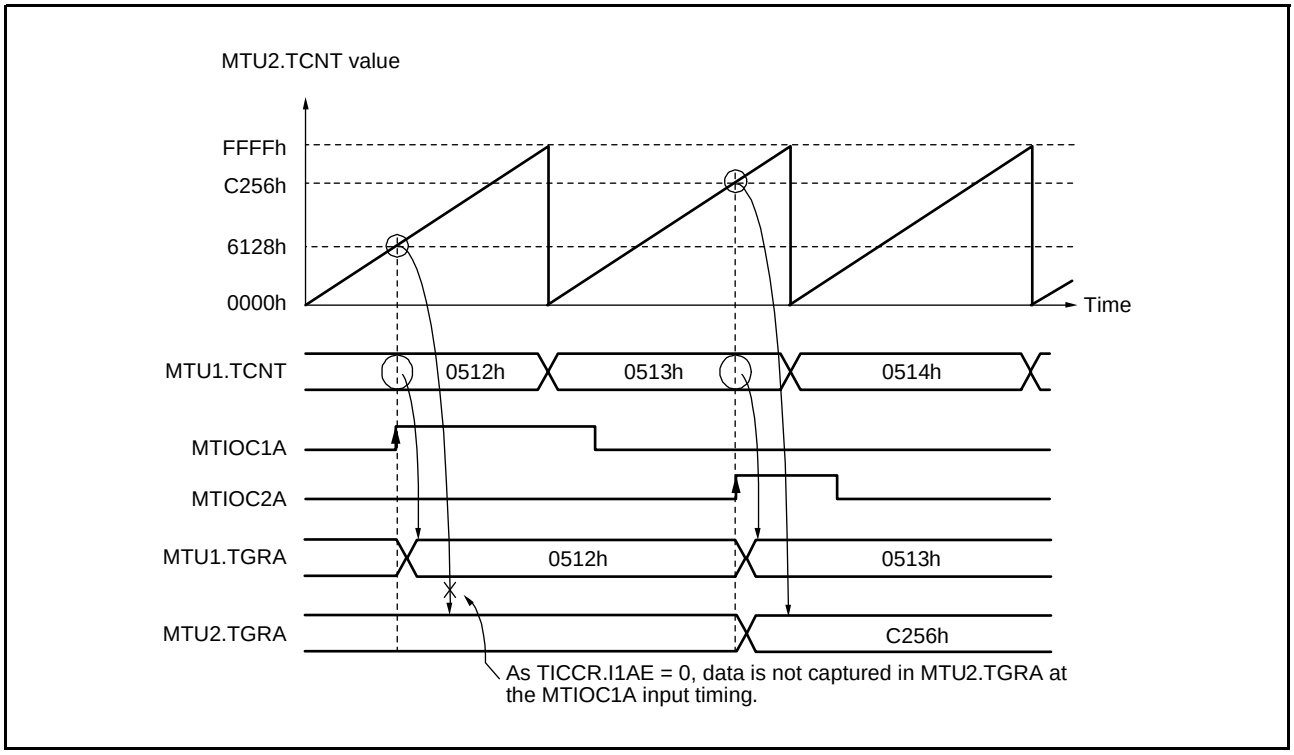
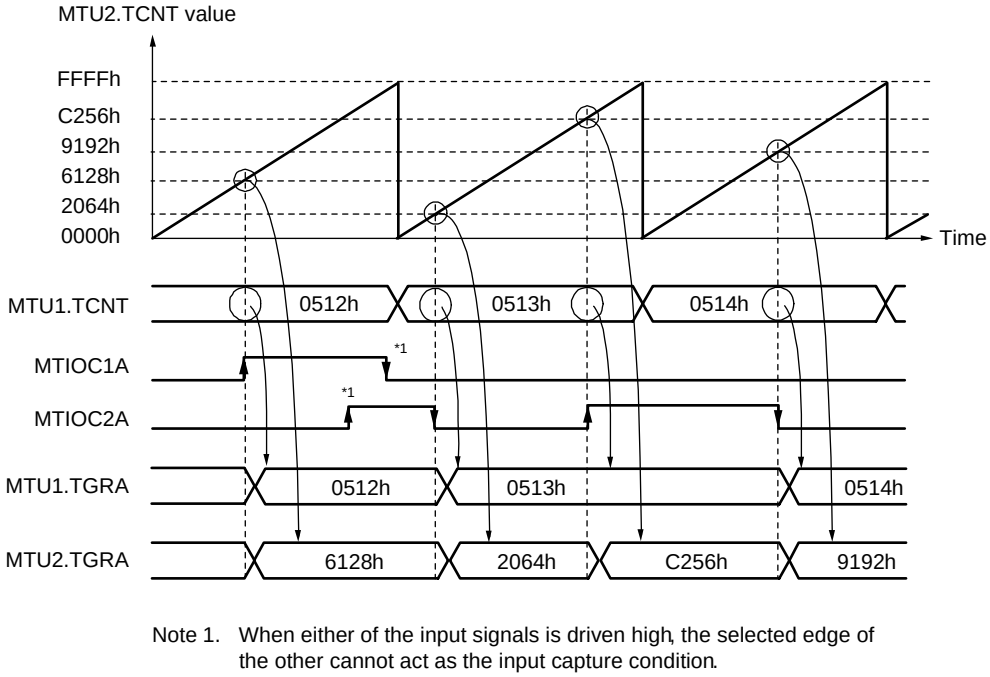


Figure 24.23 Cascaded Operation Example (b)

(4) Cascaded Operation Example (c)

Figure 24.24 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE and I1AE bits have been set to 1 to include the MTIOC2A and MTIOC1A pins in the MTU1.TGRA and MTU2.TGRA input capture conditions, respectively. In this example, the IOA[3:0] bits in both MTU1.TIOR and MTU2.TIOR have selected both the rising and falling edges for the input capture timing. Under these conditions, the ORed result of MTIOC1A and MTIOC2A input is used for the MTU1.TGRA and MTU2.TGRA input capture conditions.



Note 1. When either of the input signals is driven high, the selected edge of the other cannot act as the input capture condition.

Figure 24.24 Cascaded Operation Example (c)

(5) Cascaded Operation Example (d)

Figure 24.25 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE bit has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the IOA[3:0] bits in MTU1.TIOR have selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing while the IOA[3:0] bits in MTU2.TIOR have selected the MTIOC2A rising edge for the input capture timing.

Under these conditions, as MTU1.TIOR has selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing, the MTIOC2A edge is not used for MTU1.TGRA input capture condition although the I2AE bit in TICCR has been set to 1.

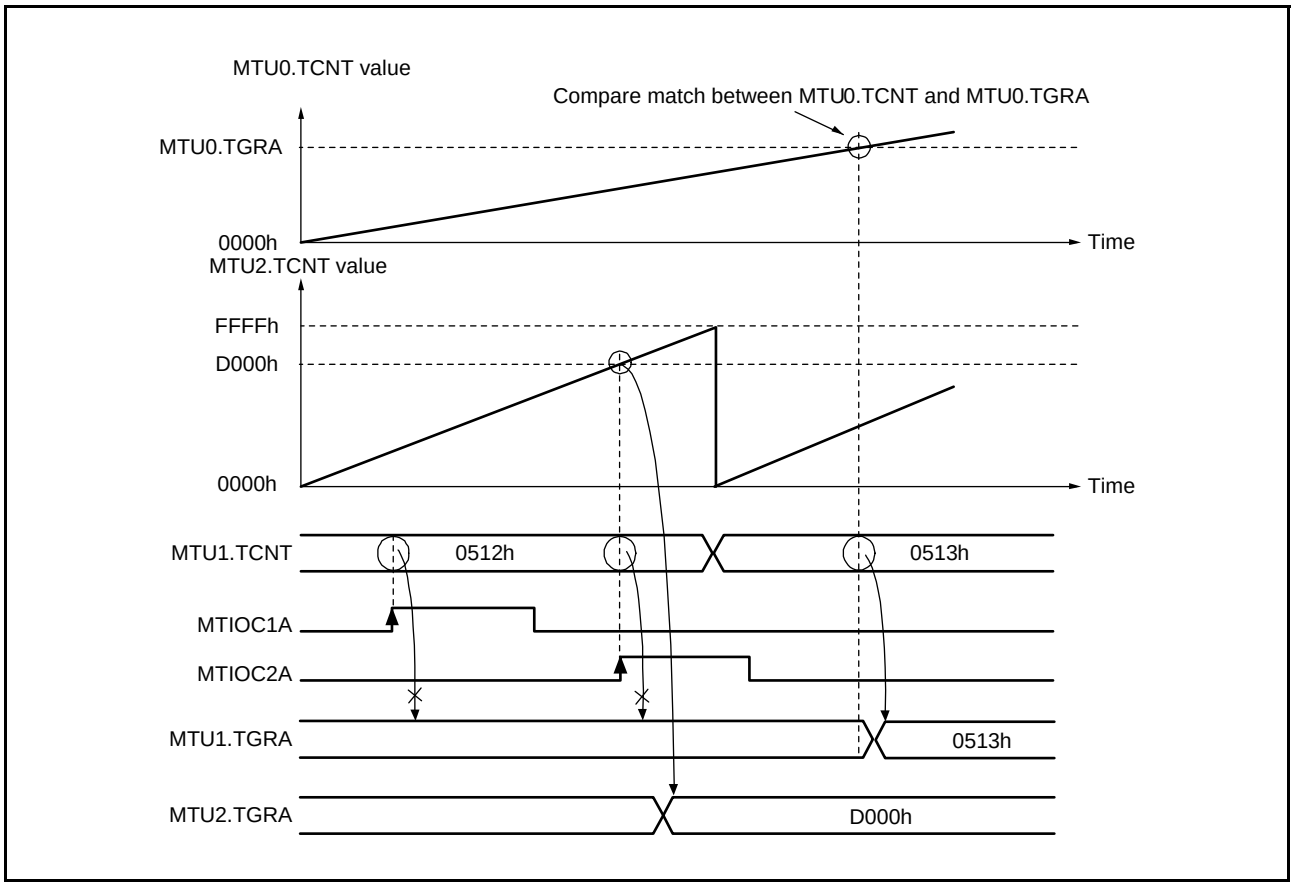


Figure 24.25 Cascaded Operation Example (d)

24.3.5 PWM Modes

PWM modes are provided to output PWM waveforms from the external pins. The output level can be selected as low, high, or toggle output in response to a compare match of each TGR.
PWM waveforms in the range of 0% to 100% duty cycle can be output according to the TGR settings.
By designating TGR compare match as the counter clearing source, the PWM cycle can be specified in that register.
Every channel except MTU5 and MTU8 can be set to PWM mode independently. Channels set to PWM mode can perform synchronous operation with each other or other channels set to any other mode.
There are two PWM modes as described below.

(a) PWM Mode 1

PWM waveforms are output from the MTIOCnA and MTIOCnC pins by pairing TGRA with TGRB and TGRC with TGRD. The levels specified by the TIOR.IOA[3:0] and IOC[3:0] bits are output from the MTIOCnA and MTIOCnC pins at compare matches A and C, and the level specified by the TIOR.IOB[3:0] and IOD[3:0] bits are output at compare matches B and D (n = 0 to 4, 6, 7). The initial output value is set in TGRA or TGRC. If the values set in paired TGRs are identical, the output value does not change even when a compare match occurs.
In PWM mode 1, PWM waveforms in up to 12 phases can be output.

(b) PWM Mode 2

PWM waveform output is generated using one TGR as the cycle register and the others as duty registers. The level specified in TIOR is output at compare matches. Upon counter clearing by a cycle register compare match, the initial value set in TIOR is output from each pin. If the values set in the cycle and duty registers are identical, the output value does not change even when a compare match occurs.

Up to eight phases of PWM waveforms can be output by combining synchronous clearing of channels that cannot be set to PWM mode 2 as synchronous operation.

The correspondence between PWM output pins and registers is shown in Table 24.65.

Table 24.65 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Register**	**PWM Mode 1**	**PWM Mode 2**
MTU0	TGRA	MTIOC0A	MTIOC0A
	TGRB		MTIOC0B
	TGRC	MTIOC0C	MTIOC0C
	TGRD		MTIOC0D
MTU1	TGRA	MTIOC1A	MTIOC1A
	TGRB		MTIOC1B
MTU2	TGRA	MTIOC2A	MTIOC2A
	TGRB		MTIOC2B
MTU3	TGRA	MTIOC3A	Setting prohibited
	TGRB		
	TGRC	MTIOC3C	
	TGRD		
MTU4	TGRA	MTIOC4A	
	TGRB		
	TGRC	MTIOC4C	
	TGRD		
MTU6	TGRA	MTIOC6A	
	TGRB		
	TGRC	MTIOC6C	
	TGRD		
MTU7	TGRA	MTIOC7A	
	TGRB		
	TGRC	MTIOC7C	
	TGRD		

Note: In PWM mode 2, PWM waveform output is not possible for the TGR register in which the PWM cycle is set.

(1) Example of PWM Mode Setting Procedure

Figure 24.26 shows an example of the PWM mode setting procedure.

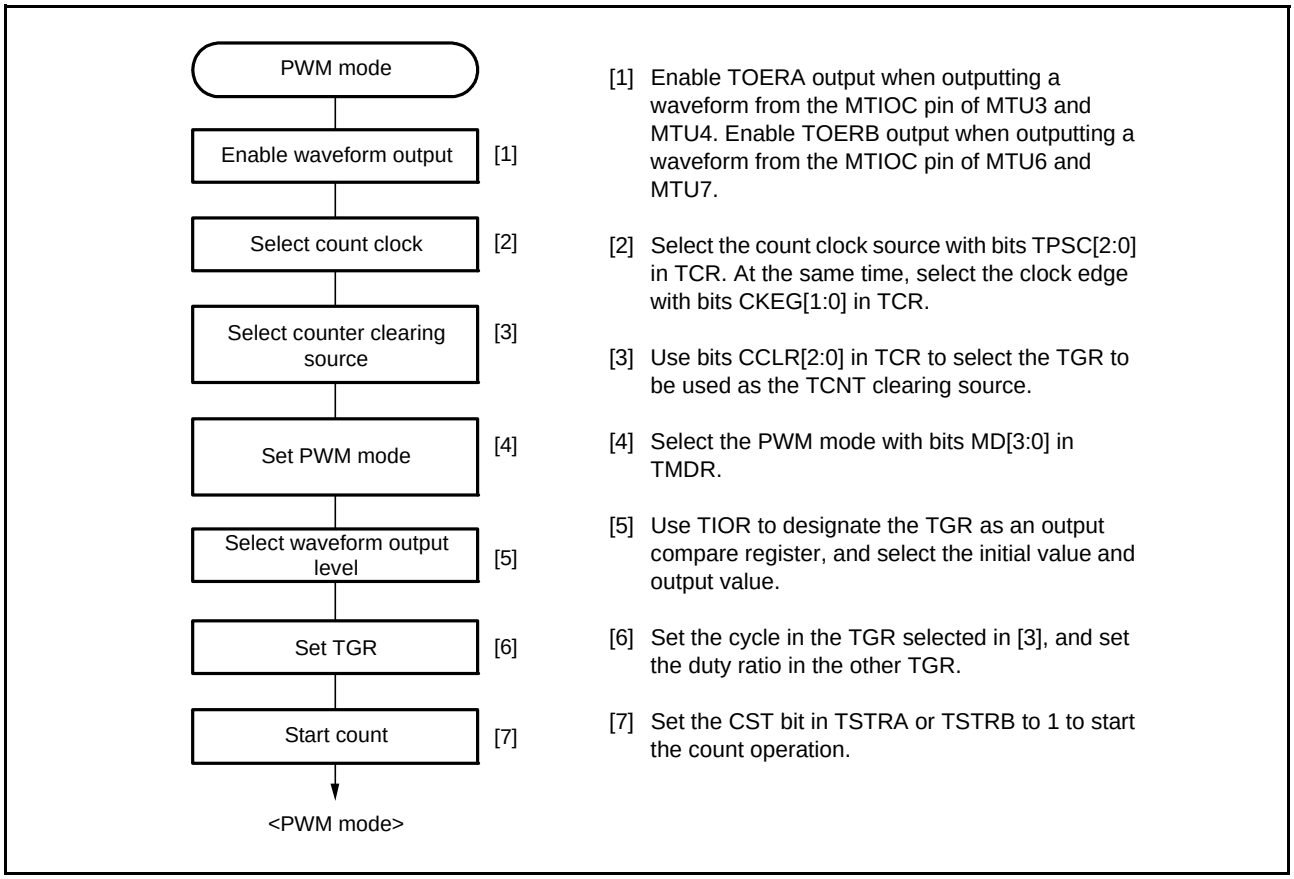


Figure 24.26 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 24.27 shows an example of operation in PWM mode 1.
In this example, TGRA compare match is set as the TCNT clearing source, 0 is set as the initial output value and output value for TGRA, and 1 is set as the output value for TGRB.
In this case, the value set in TGRA is used as the cycle, and the value set in TGRB is used as the duty ratio.

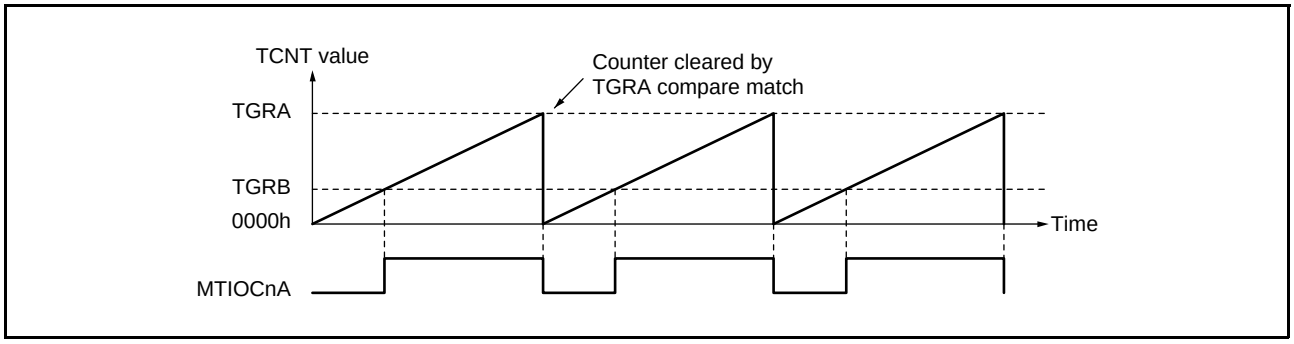


Figure 24.27 Example of PWM Mode 1 Operation (n = 0 to 4, 6, 7)

Figure 24.28 shows an example of operation in PWM mode 2.
In this example, synchronous operation is designated for MTU0 and MTU1, MTU1.TGRB compare match is set as the TCNT clearing source, and low is set as the initial output value and high as the output value for the other TGR registers (MTU0.TGRA to MTU0.TGRD and MTU1.TGRA), outputting 5-phase PWM waveforms.
In this case, the value set in MTU1.TGRB is used as the cycle, and the values set in the other TGRs are used as the duty ratio.

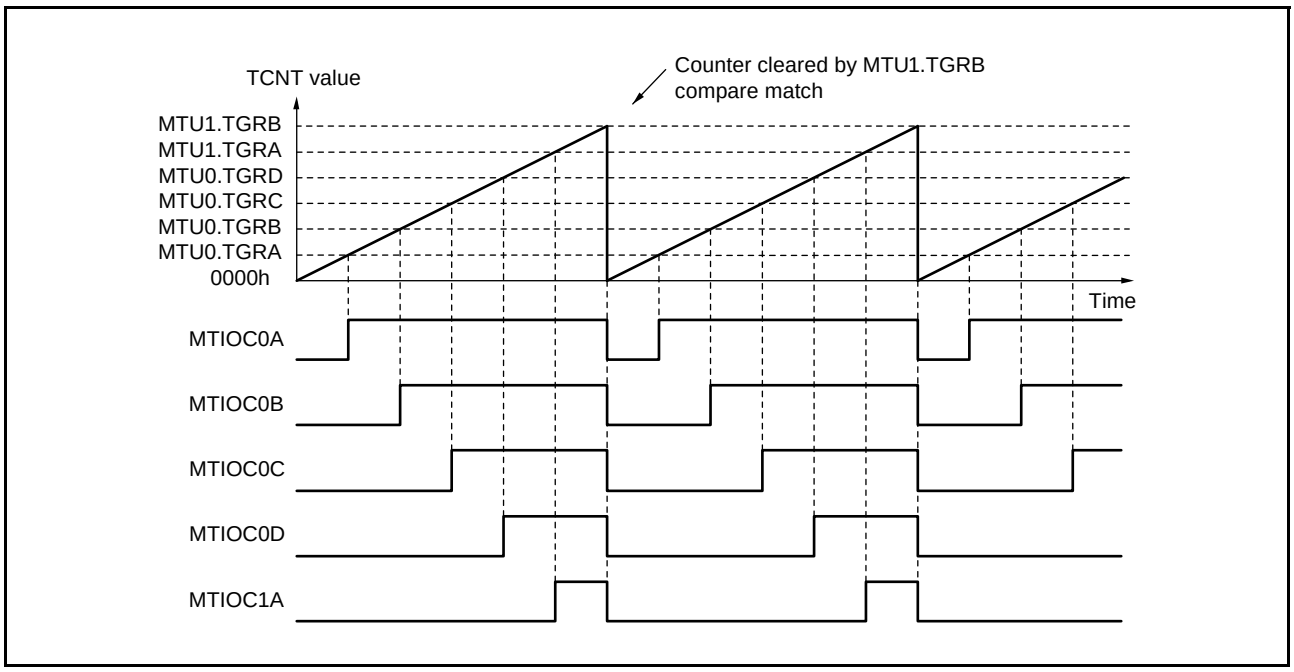


Figure 24.28 Example of PWM Mode 2 Operation

Figure 24.29 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode 1.
In this example, TGRA compare match is set as the TCNT clearing source, a low level is set as the initial output value and output value for TGRA, and a high level is set as the output value for TGRB.

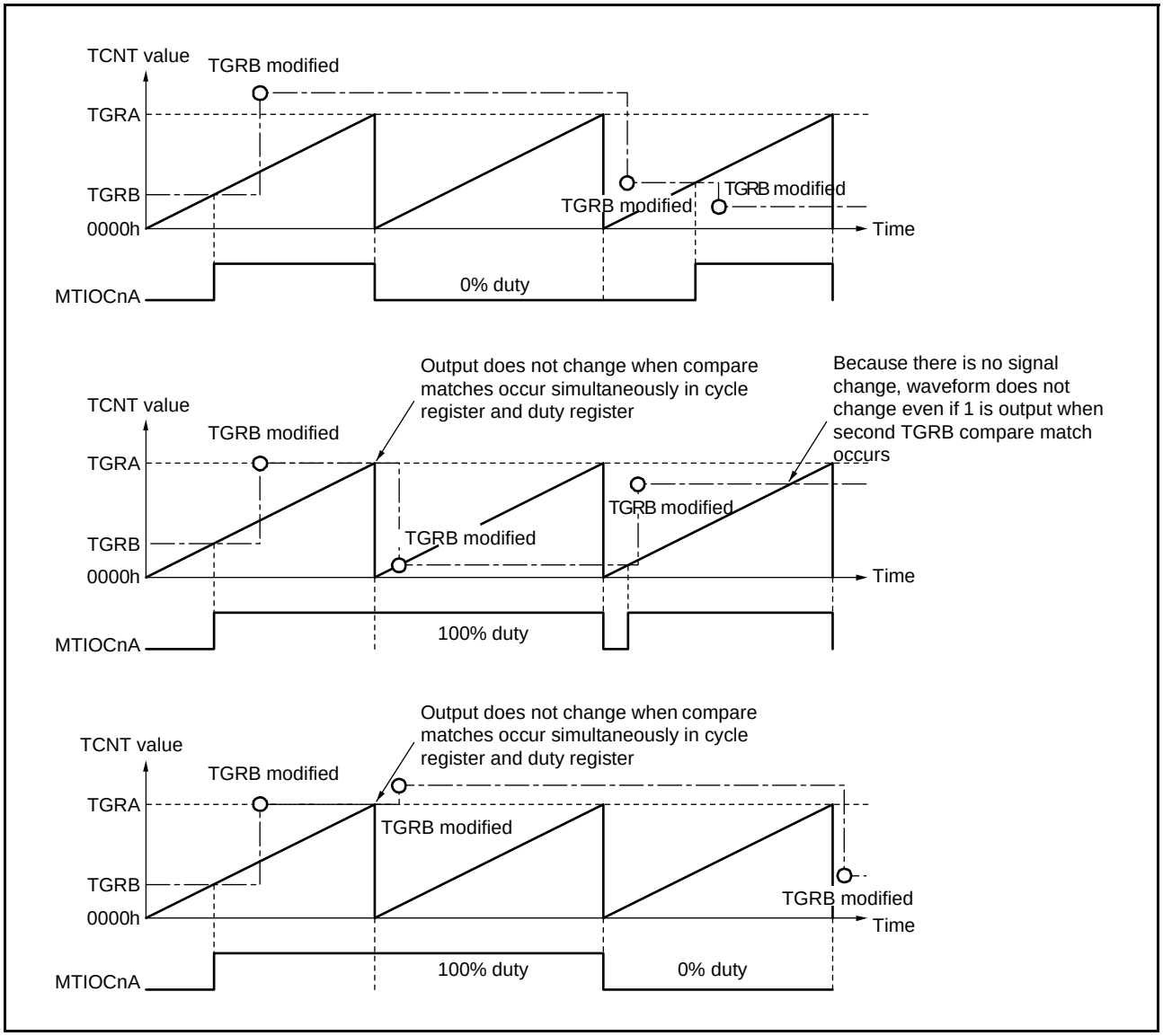


Figure 24.29 Examples of PWM Mode Operation (PWM Waveform Output with 0% Duty and 100% Duty) (n = 0 to 4, 6, 7)

24.3.6 Phase Counting Mode

There are two phase counting modes: 16-bit phase counting mode in which MTU1 and MTU2 operate independently, and cascade connection 32-bit phase counting mode in which MTU1 and MTU2 are cascaded.
In phase counting mode, the phase difference between two external input clocks is detected and the corresponding TCNT is incremented or decremented.
Two external clock input pins for each phase counting mode are not affected by the settings of TCR.TPSC[2:0], TCR2.TPSC2[2:0], and CKEG[1:0]. The two external clock input pins used in 16-bit phase counting mode and cascade connection 32-bit phase counting mode of MTU2 can be selected by MTU1.TMDR3.PHCKSEL. In a phase counting mode other than 16-bit phase counting mode and cascade connection 32-bit phase counting mode of MTU2, MTCLKA and MTCLKB are selected for A-phase and B-phase, respectively. In phase counting mode, the external clock pins MTCLKA, MTCLKB, MTCLKC, and MTCLKD are used for two-phase encoder pulse input.
Table 24.66 lists the external clock input pins to be connected in each phase counting mode.

Table 24.66 Clock Input Pins in Phase Counting Mode

		External Clock Input Pins	
Phase Counting Mode	**TMDR3.PHCKSEL bit**	**A-Phase**	**B-Phase**
MTU1 16-bit phase counting mode	x (Don't care)	MTCLKA	MTCLKB
MTU2 16-bit phase counting mode	0	MTCLKA	MTCLKB
	1 (initial value)	MTCLKC	MTCLKD
Cascade connection 32-bit phase counting mode	0	MTCLKA	MTCLKB
	1 (initial value)	MTCLKC	MTCLKD

24.3.6.1 16-Bit Phase Counting Mode

When the MTU1.TMDR3.LWA is 0, 16-bit phase counting mode can be set individually for MTU1 and MTU2.
In 16-bit phase counting mode, the phase difference between two external input clocks is detected and the 16-bit counter TCNT of the corresponding channel is incremented or decremented.
When 16-bit phase counting mode is specified, an external clock is selected as the count clock and TCNT operates as an up-counter/down-counter regardless of the setting of TCR.TPSC[2:0], TCR2.TPSC2[2:0], and CKEG[1:0]. However, the functions of TCR.CCLR[1:0],TIOR, TIER, and TGR are enabled, and input capture/compare match and interrupt functions can be used.
These external input pins can be used for two-phase encoder pulse input.
When an overflow occurs while TCNT is counting up and the corresponding TIER.TCIEV bit is 1, a TCIV interrupt is generated. When an underflow occurs while TCNT is counting down and the corresponding TIER.TCIEU bit is 1, a TCIU interrupt is generated.
The TSR.TCFD flag is the count direction flag. Read the TCFD flag to check whether TCNT is counting up and down.

(1) Example of 16-Bit Phase Counting Mode Setting Procedure

Figure 24.30 shows an example of the phase counting mode setting procedure.

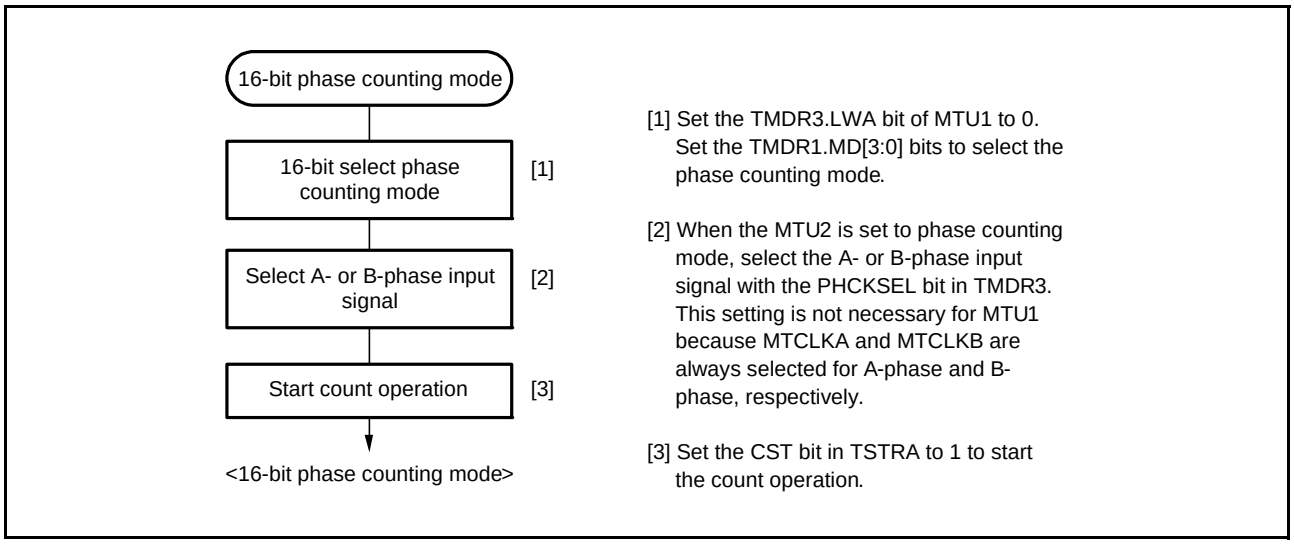


Figure 24.30 Example of 16-Bit Phase Counting Mode Setting Procedure

(2) Examples of 16-Bit Phase Counting Mode Operation

In phase counting mode, TCNT is incremented or decremented according to the phase difference between two external clocks. There are five modes according to the count conditions. Each mode operates under the condition PHCKSEL = 1, which means the phase clock for MTU1 is input from MTCLKA or MTCLKB and that for MTU2 is input from MTCLKC or MTCLKD.

(a) Phase Counting Mode 1

Figure 24.31 shows an example of operation in phase counting mode 1, and Table 24.67 summarizes the TCNT up-counting and down-counting conditions.

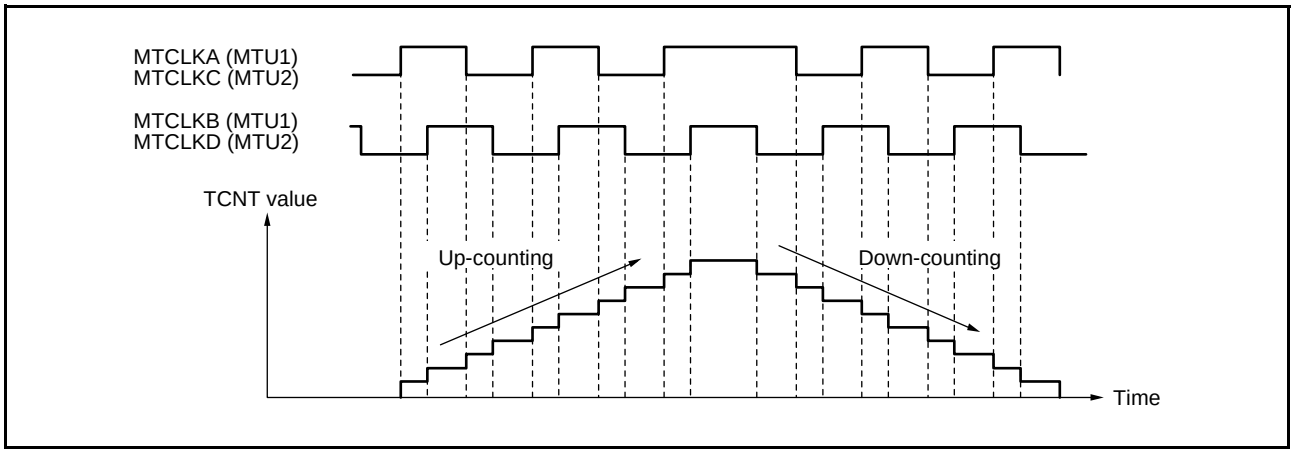


Figure 24.31 Example of Operation in Phase Counting Mode 1

Table 24.67 Up-Counting and Down-Counting Conditions in Phase Counting Mode 1

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	Up-counting
Low	↓	
↑	Low	
↓	High	
High	↓	Down-counting
Low	↑	
↑	High	
↓	Low	

↑ : Rising edge

↓ : Falling edge

(b) Phase Counting Mode 2

Figure 24.32 to Figure 24.34 show the examples of operation in phase counting mode 2 and Table 24.68 summarizes the TCNT up-counting and down-counting conditions.

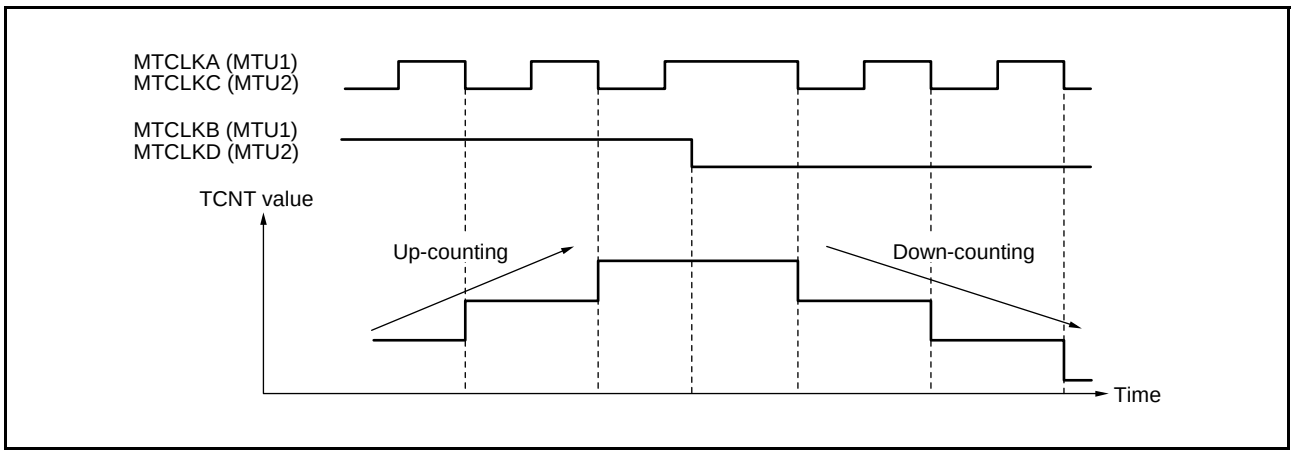


Figure 24.32 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] = 00b (n = 1, 2))

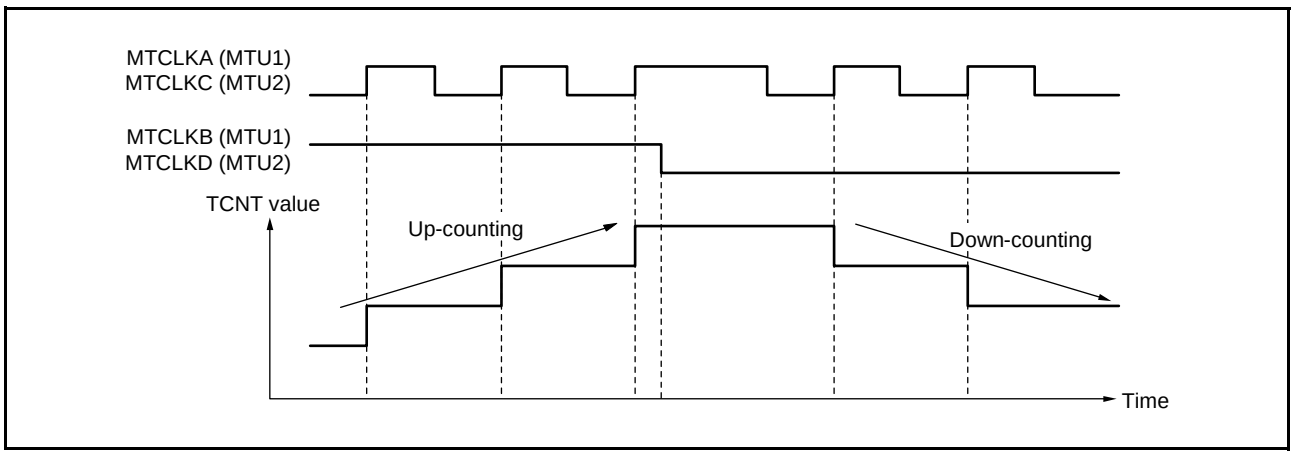


Figure 24.33 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] = 01b (n = 1, 2))

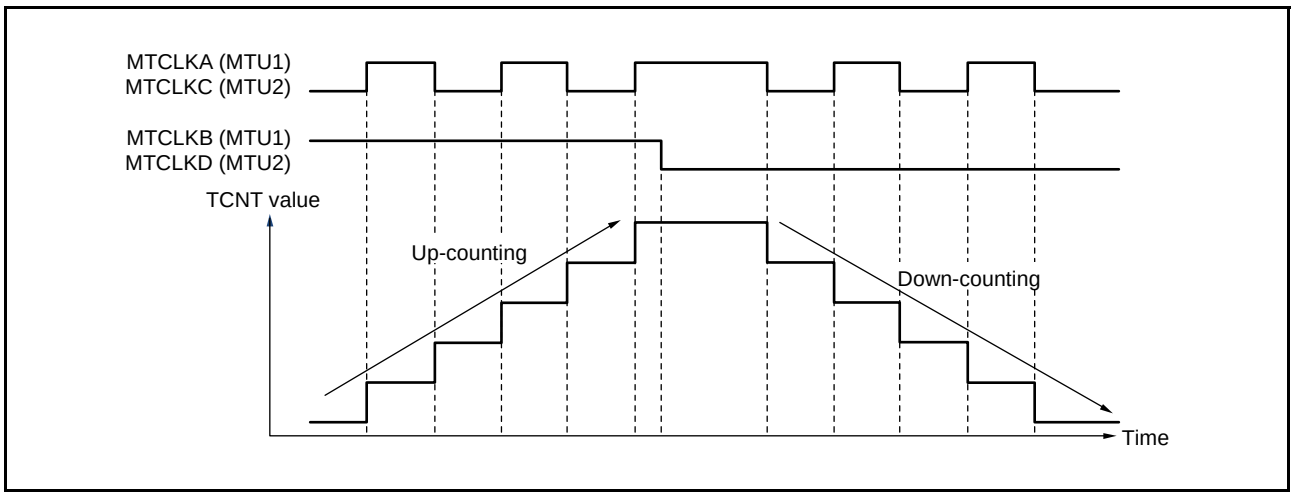


Figure 24.34 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 24.68 Up-Counting and Down-Counting Conditions in Phase Counting Mode 2

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
00b	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	
	↓	High	Up-counting
	High	↓	
	Low	↑	
	↑	High	
	↓	Low	Down-counting
01b	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	Down-counting
	↓	High	Not counted (Don't care)
	High	↓	
	Low	↑	
	↑	High	Up-counting
	↓	Low	Not counted (Don't care)
1xb	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	Down-counting
	↓	High	Up-counting
	High	↓	Not counted (Don't care)
	Low	↑	
	↑	High	Up-counting
	↓	Low	Down-counting

↑ : Rising edge

↓ : Falling edge

(c) Phase Counting Mode 3

Figure 24.35 to Figure 24.37 show the examples of operation in phase counting mode 3 and Table 24.69 summarizes the TCNT up-counting and down-counting conditions.

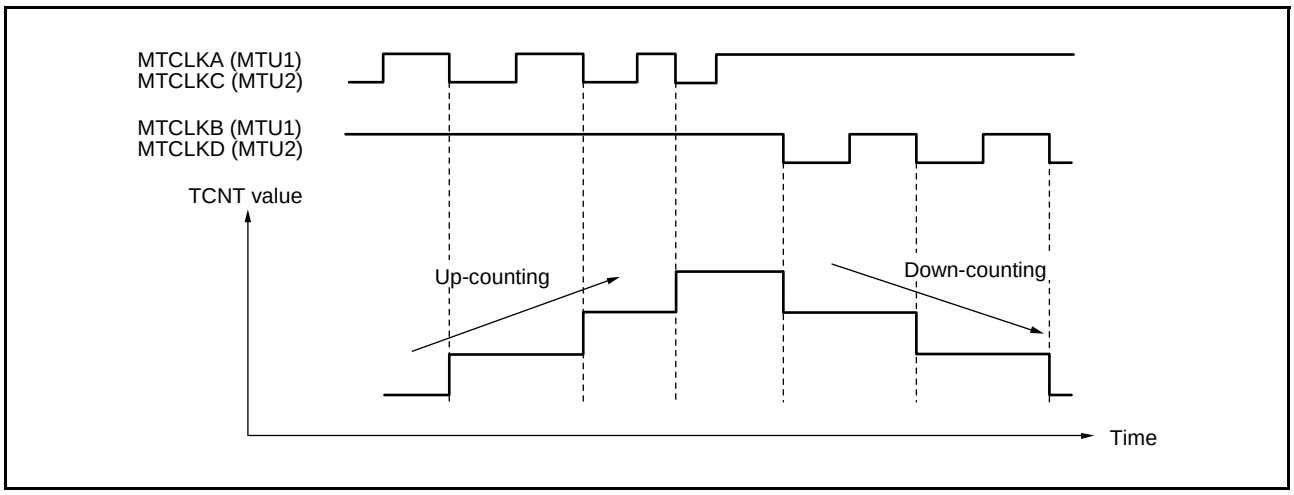


Figure 24.35 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] = 00b (n = 1, 2))

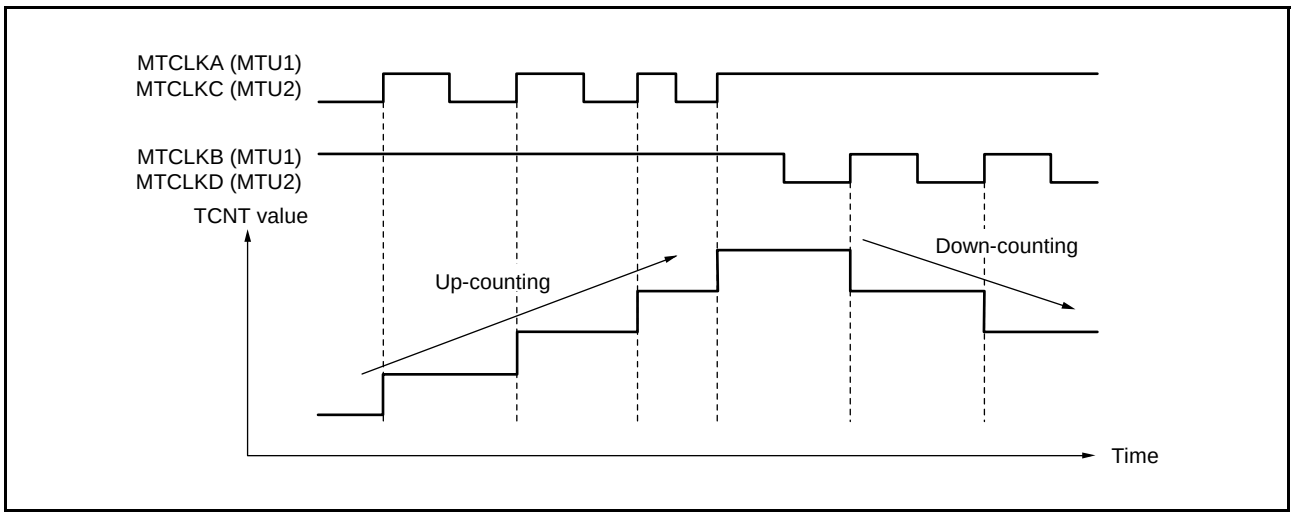


Figure 24.36 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] = 01b (n = 1, 2))

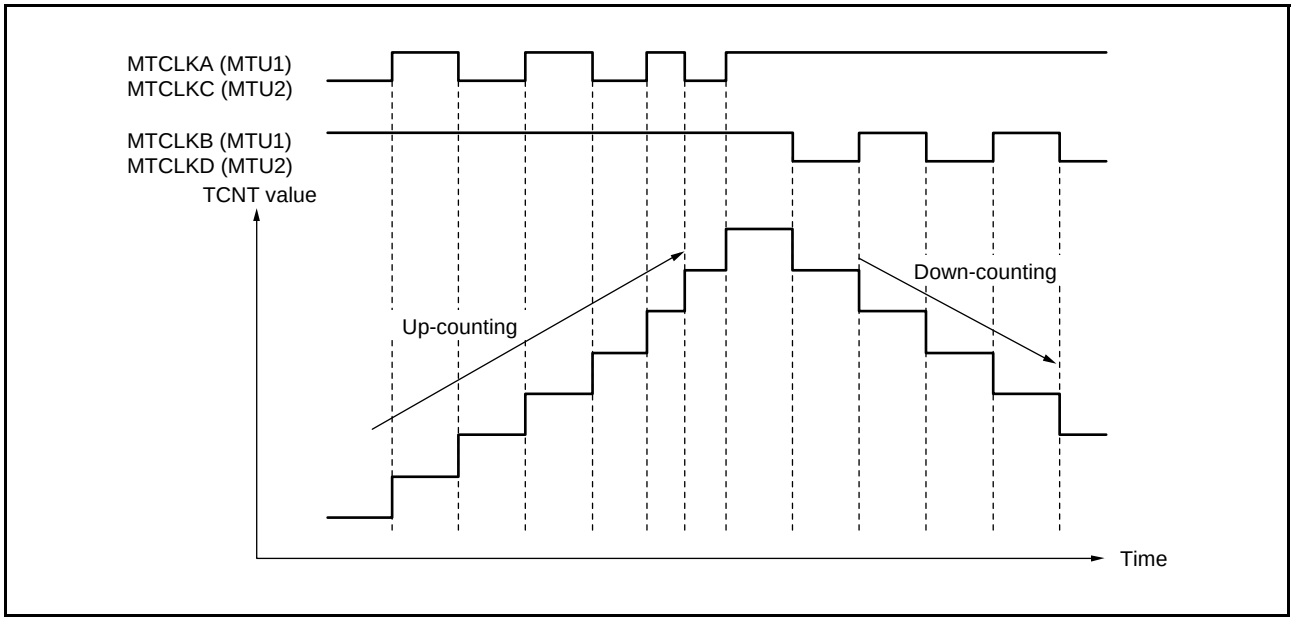


Figure 24.37 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 24.69 Up-Counting and Down-Counting Conditions in Phase Counting Mode 3

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
00b	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	
	↓	High	Up-counting
	High	↓	Down-counting
	Low	↑	Not counted (Don't care)
	↑	High	
	↓	Low	
01b	High	↑	Down-counting
	Low	↓	Not counted (Don't care)
	↑	Low	
	↓	High	
	High	↓	
	Low	↑	
	↑	High	Up-counting
	↓	Low	Not counted (Don't care)
1xb	High	↑	Down-counting
	Low	↓	Not counted (Don't care)
	↑	Low	
	↓	High	Up-counting
	High	↓	Down-counting
	Low	↑	Not counted (Don't care)
	↑	High	Up-counting
	↓	Low	Not counted (Don't care)

↑ : Rising edge

↓ : Falling edge

(d) Phase Counting Mode 4

Figure 24.38 shows an example of operation in phase counting mode 4, and Table 24.70 summarizes the TCNT up-counting and down-counting conditions.

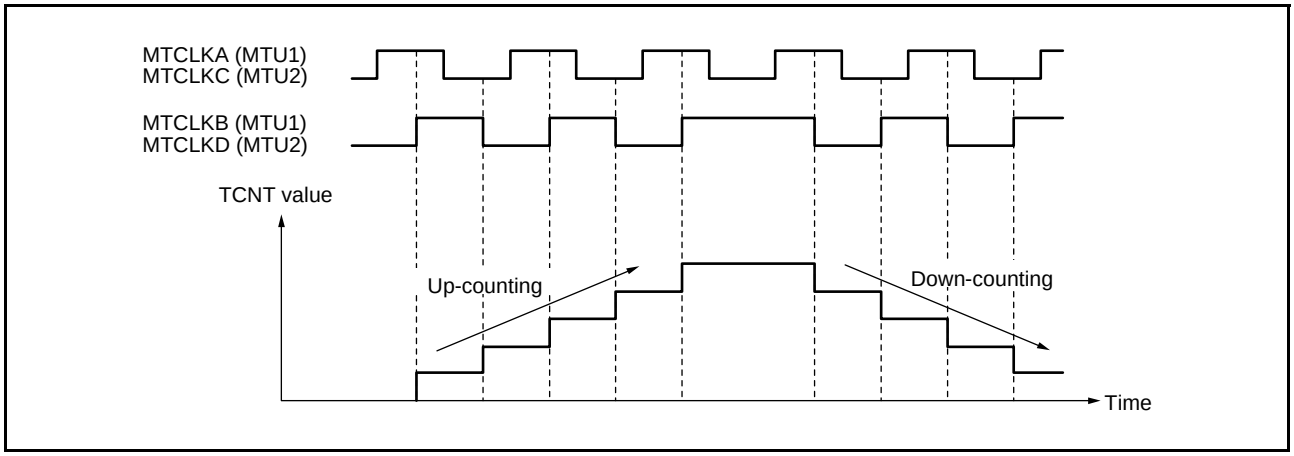


Figure 24.38 Example of Operation in Phase Counting Mode 4

Table 24.70 Up-Counting and Down-Counting Conditions in Phase Counting Mode 4

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	Up-counting
Low	↓	
↑	Low	Not counted (Don't care)
↓	High	
High	↓	Down-counting
Low	↑	
↑	High	Not counted (Don't care)
↓	Low	

↑ : Rising edge

↓ : Falling edge

(e) Phase Counting Mode 5

Figure 24.39 and Figure 24.40 show the examples of operation in phase counting mode 5 and Table 24.71 summarizes the TCNT up-counting and down-counting conditions.

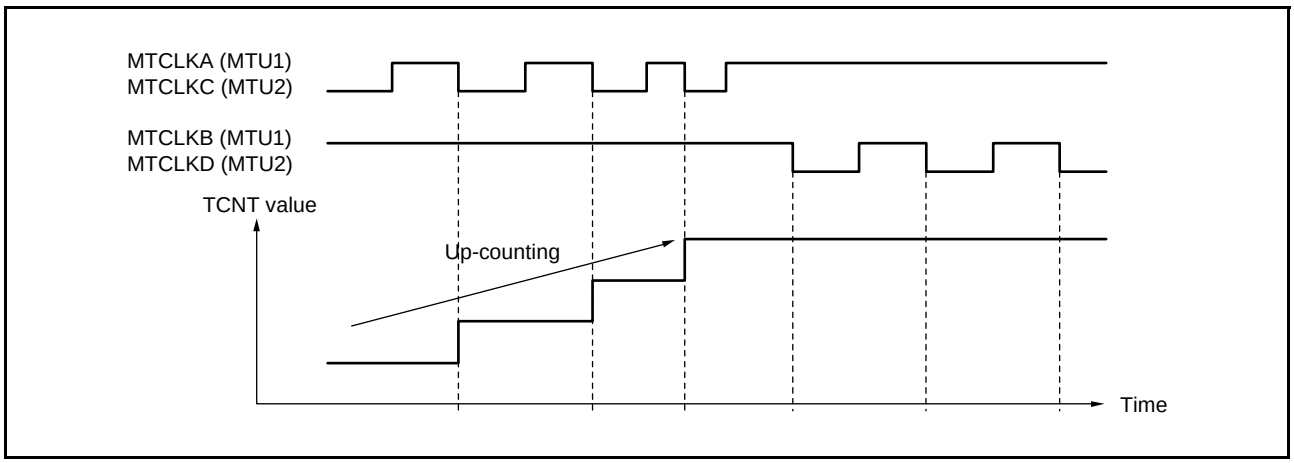


Figure 24.39 Example of Operation in Phase Counting Mode 5 (When MTUn.TCR2.PCB[1:0] = 0xb (n = 1, 2))

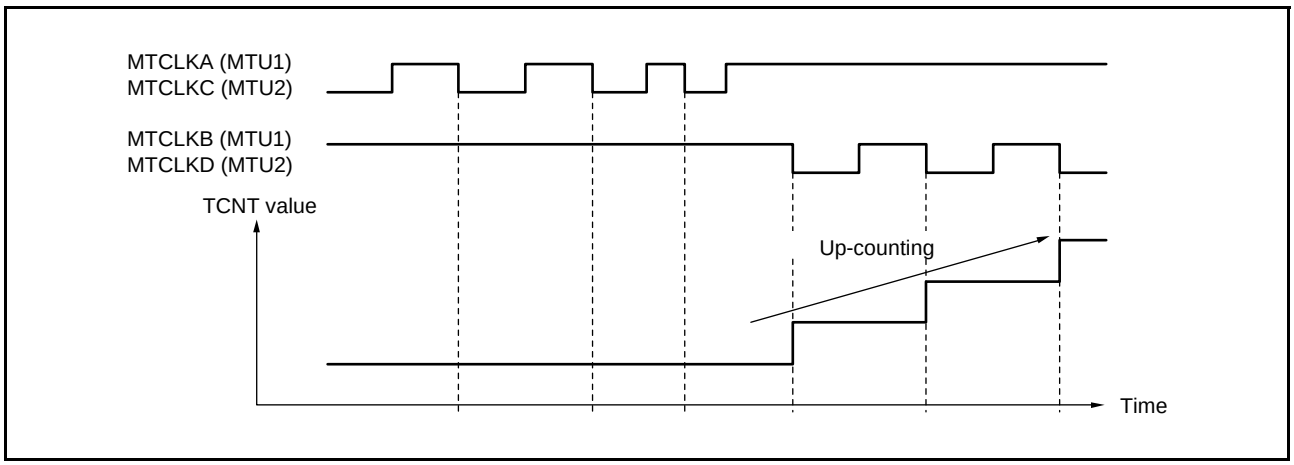


Figure 24.40 Example of Operation in Phase Counting Mode 5 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 24.71 Up-Counting and Down-Counting Conditions in Phase Counting Mode 5

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
0xb	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	
	↓	High	Up-counting
	High	↓	Not counted (Don't care)
	Low	↑	
	↑	High	
	↓	Low	Up-counting
1xb	High	↑	Not counted (Don't care)
	Low	↓	Up-counting
	↑	Low	Not counted (Don't care)
	↓	High	
	High	↓	Up-counting
	Low	↑	Not counted (Don't care)
	↑	High	
	↓	Low	

↑ : Rising edge

↓ : Falling edge

(3) 16-Bit Phase Counting Mode Application Example

Figure 24.41 shows an example in which MTU1 is in phase counting mode, and MTU1 is coupled with MTU0 to input 2-phase encoder pulses of a servo motor in order to detect position or speed.

MTU1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to MTCLKA and MTCLKB.

In MTU0, MTU0.TGRC compare match is specified as the TCNT clearing source and MTU0.TGRA and MTU0.TGRC are used for the compare match function and are set with the speed control cycle and position control cycle.

MTU0.TGRB is used for input capture, with MTU0.TGRB and MTU0.TGRD operating in buffer mode. The MTU1 count clock is designated as the MTU0.TGRB input capture source, and the widths of 2-phase encoder 4-multiplication pulses are detected.

MTU1.TGRA and MTU1.TGRB for MTU1 are designated for the input capture function and MTU0.TGRA and MTU0.TGRC compare matches in MTU0 are selected as the input capture sources to store the up/down-counter values for the control cycles.

This procedure enables the accurate detection of position and speed.

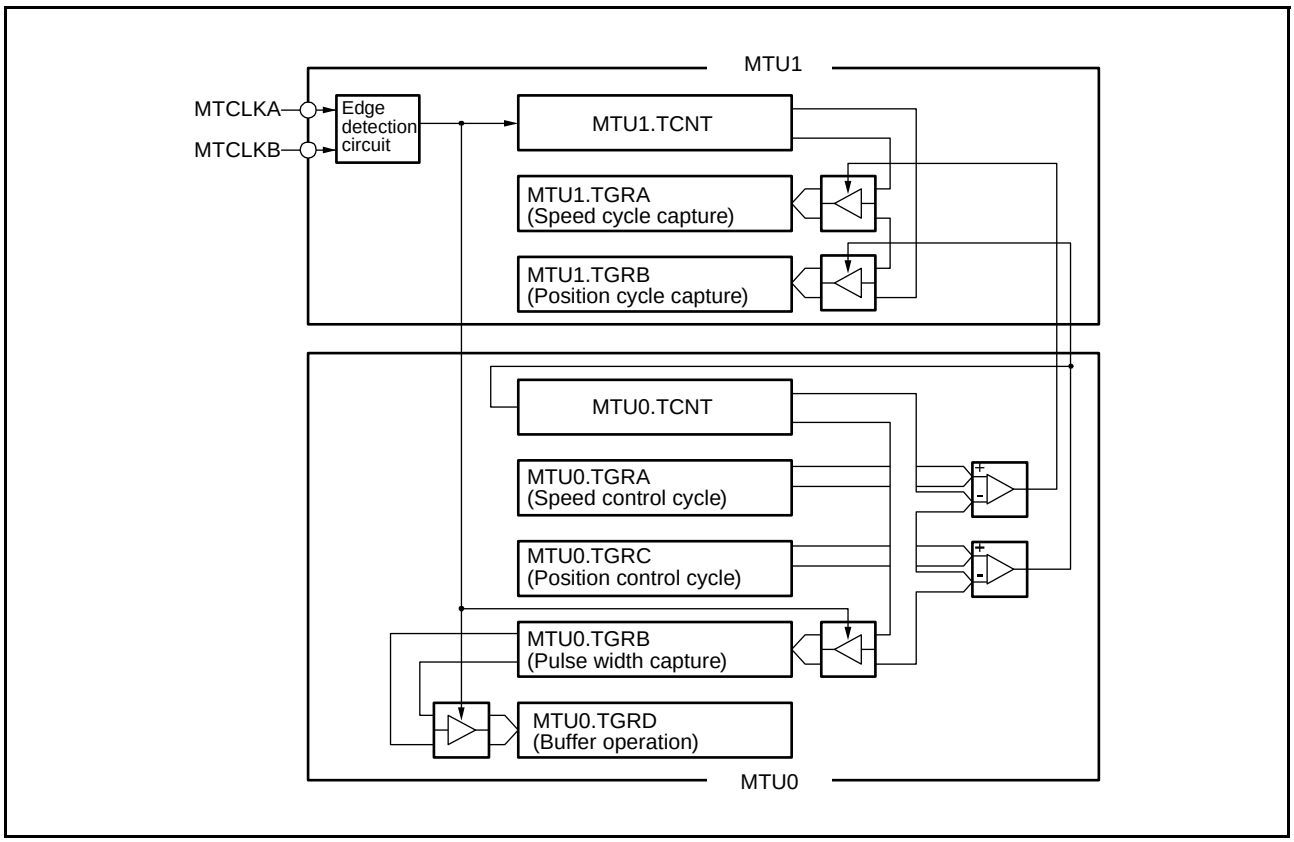


Figure 24.41 16-Bit Phase Counting Mode Application Example

24.3.6.2 Cascade Connection 32-Bit Phase Counting Mode

When MTU1 is set to phase counting mode by setting MTU1.TMDR3.LWA = 1, MTU1 and MTU2 are connected to operate in cascade connection 32-bit phase counting mode as shown in Figure 24.42. When this mode is used, the TCR, TCR2, TIOR, TIER, TGR, and TSR registers are controlled by MTU1 and the settings of MTU2 are disabled. Refer to Figure 24.43 for the procedure for setting cascade connection 32-bit phase counting mode.

In this mode, three-phase (A, B, and Z) signals can be input. As an encoder pulse signal, the external input phase clocks MTCLKA and MTCLKB or MTCLKC and MTCLKD can be selected for A-phase and B-phase, and MTIOC1A can be selected for Z-phase, respectively. Refer to Table 24.70 for selecting external clock input of A-phase and B-phase. A counter event is generated using an A-phase or B-phase pulse and counted by the 32-bit counter MTU1.TCNTLW.

An input capture can also be generated using a Z-phase signal; thus each speed can be measured using the captured value in the general register.

Furthermore, MTU8 can be used as a channel for measuring a 1-ms interval, and a compare match signal can be output at a 1-ms interval to the MTU1 and MTU2, which operate in cascade connection 32-bit phase counting mode. That is, a compare match signal of MTU8 is used as a capture or clear signal of MTU1 and MTU2, and the number of A-phase and B-phase pulses for a 1-ms period can be measured.

When MTU0 or MTU5 is specified as the channel for measuring a Z-phase signal pulse, this compare match signal can be output as a capture signal or clear signal to MTU0 or MTU5, thus the Z-phase count at a 1-ms interval can be measured.

In addition, a counter event signal of combined MTU1 and MTU2 can be used as a capture signal of MTU8, and measurement can be performed including the intervals of A, B, or both phase pulses. In this case, the general register for measurement should be set to a register for buffer operation.

Refer to section 24.3.4, Cascaded Operation, for details on the cascade connection function for connecting MTU1 and MTU2 in a mode other than cascade connection 32-bit phase counting mode.

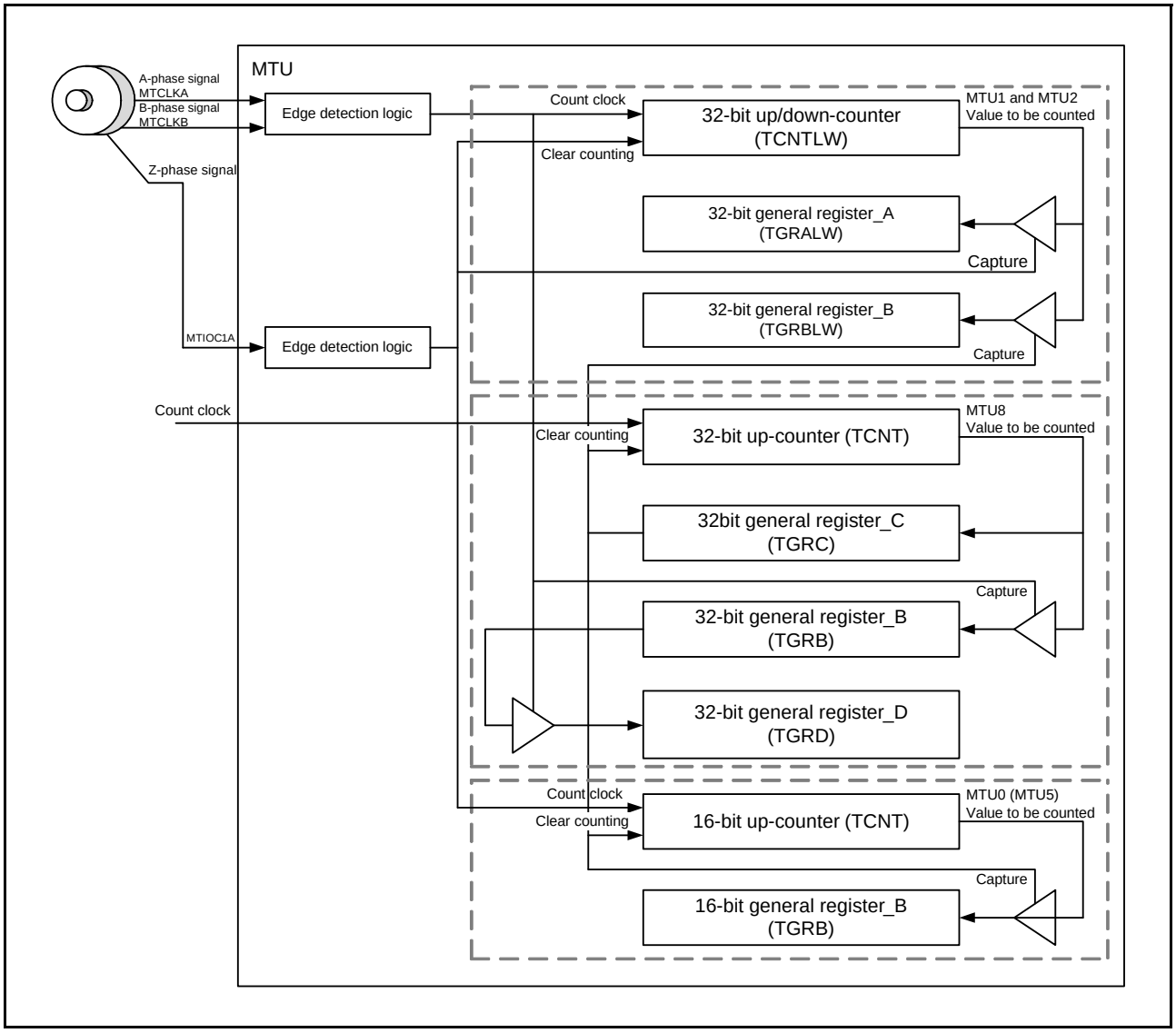


Figure 24.42 Block Diagram for Operation in Cascade Connection 32-Bit Phase Counting Mode

(1) Example of Setting Cascade Connection 32-Bit Phase Counting Mode

Figure 24.43 shows an example of the procedure for setting cascade connection 32-bit phase counting mode.

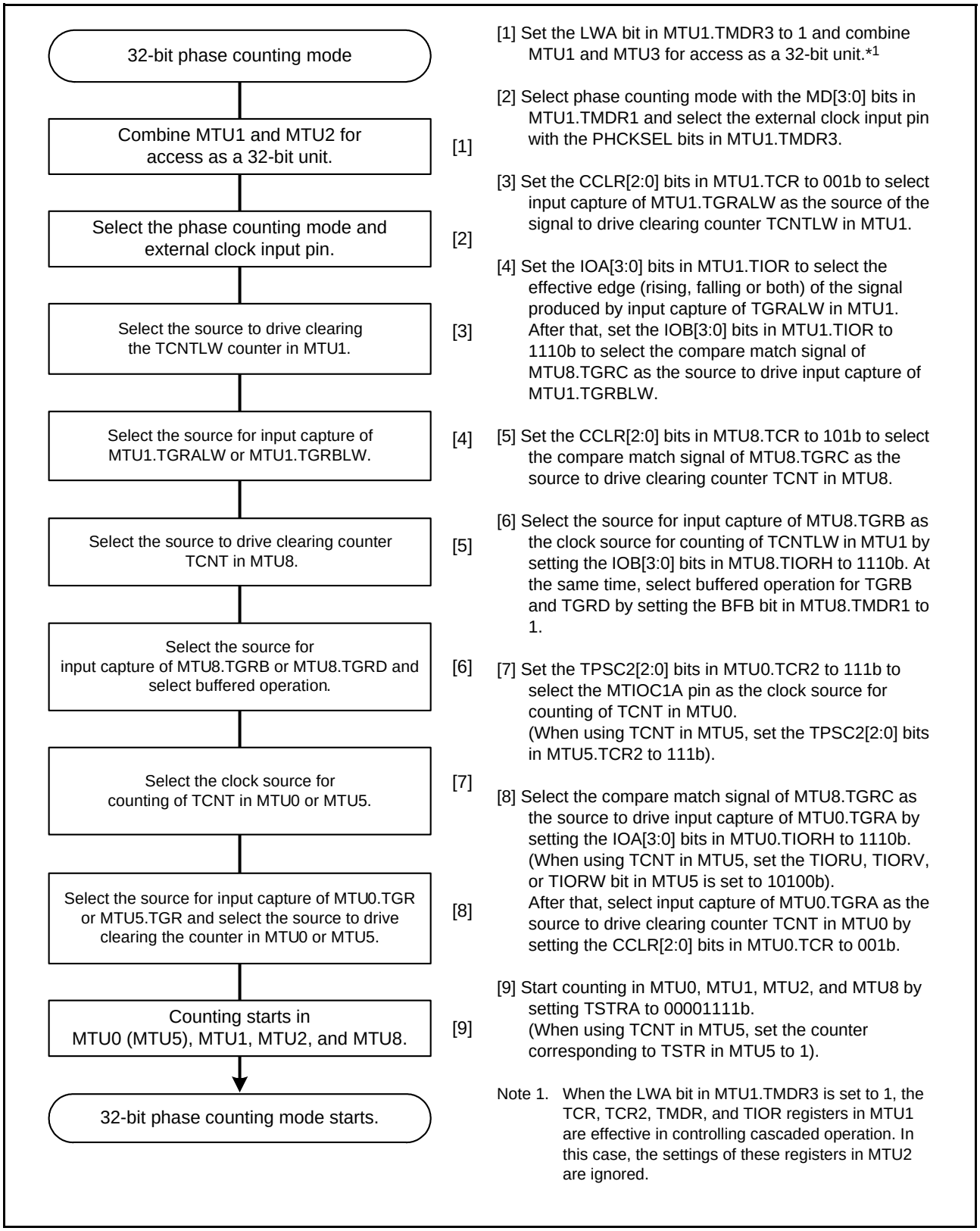


Figure 24.43 Procedure for Setting Cascade Connection 32-Bit Phase Counting Mode

24.3.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, six phases of positive and negative PWM waveforms (12 phases in total) that share a common wave transition point can be output by combining MTU3 and MTU4 and MTU6 and MTU7.
When set for reset-synchronized PWM mode, the MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, and MTIOC7D pins function as PWM output pins and timer counters 3 and 6 (MTU3.TCNT and MTU6.TCNT) functions as an up-counter.
Table 24.72 shows the PWM output pins used. Table 24.73 shows the settings of the registers.

Table 24.72 Output Pins for Reset-Synchronized PWM Mode

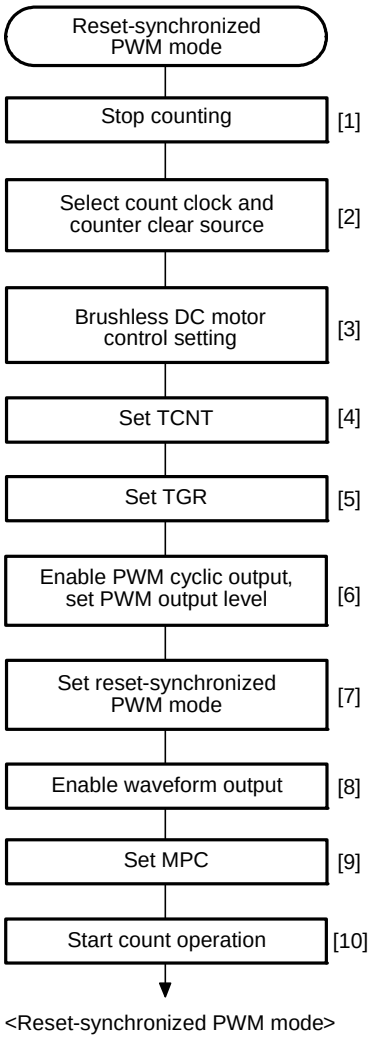
Channel	Output Pin	Description
MTU3	MTIOC3B	PWM output pin 1
	MTIOC3D	PWM output pin 1' (negative-phase waveform of PWM output 1)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (negative-phase waveform of PWM output 2)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (negative-phase waveform of PWM output 3)
MTU6	MTIOC6B	PWM output pin 4
	MTIOC6D	PWM output pin 4' (negative-phase waveform of PWM output 4)
MTU7	MTIOC7A	PWM output pin 5
	MTIOC7C	PWM output pin 5' (negative-phase waveform of PWM output 5)
	MTIOC7B	PWM output pin 6
	MTIOC7D	PWM output pin 6' (negative-phase waveform of PWM output 6)

Table 24.73 Register Settings for Reset-Synchronized PWM Mode

Register	Setting
MTU3.TCNT	Initial setting (0000h)
MTU4.TCNT	Initial setting (0000h)
MTU3.TGRA	Set the count cycle for MTU3.TCNT
MTU3.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC3B and MTIOC3D pins
MTU4.TGRA	Set the transition point of the PWM waveform to be output from the MTIOC4A and MTIOC4C pins
MTU4.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC4B and MTIOC4D pins
MTU6.TCNT	Initial setting (0000h)
MTU7.TCNT	Initial setting (0000h)
MTU6.TGRA	Set the count cycle for MTU6.TCNT
MTU6.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC6B and MTIOC6D pins
MTU7.TGRA	Set the transition point of the PWM waveform to be output from the MTIOC7A and MTIOC7C pins
MTU7.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC7B and MTIOC7D pins

(1) Example of Procedure for Setting Reset-Synchronized PWM Mode

Figure 24.44 shows an example of procedure for setting the reset-synchronized PWM mode.



[1] Clear the TSTRA.CST3 (TSTRB.CST6) and TSTRA.CST4 (TSTRB.CST7) bits to 0 to stop the TCNT count operation. Specify the reset-synchronized PWM mode while MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) are stopped.

[2] Set the TCR.TPSC[2:0] and TCR.CKEG[1:0] bits of MTU3 (MTU6) to select the count clock source and clock edge for MTU3 (MTU6). Set the TCR.CCLR[2:0] bits to select TGRA compare match as the counter clear source.

[3] To control a brushless DC motor, set the TGCRA.BDC bit and set the feedback signal input source and output chopping or gate signal direct output (only when MTU3 and MTU4 are used).

[4] Set MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) to 0000h.

[5] MTU3.TGRA (MTU6.TGRA) is the cycle register. Set the PWM cycle in MTU3.TGRA (MTU6.TGRA).
Set the PWM output transition timing in MTU3.TGRB (MTU6.TGRB), MTU4.TGRA (MTU7.TGRA), and MTU4.TGRB (MTU7.TGRB).
Note that the setting values should be within the range of compare match with MTU3.TCNT (MTU6.TCNT).

$X \leq$ MTU3.TGRA (X: Set value)

[6] Set the PSYE bit in the TOCR1A (TOCR1B) register to enable or disable toggle output synchronized with the PWM cycle, and set the OLSP and OLSN bits to select the PWM output level.
When specifying the PWM output level using the TOCR2A (TOCR2B) and TOLBRA (TOLBRB) in buffer operation, refer to Timer Output Level Buffer Registers (TOLBRA and TOLBRB).

[7] Set the MD[3:0] bits in MTU3.TMDR1 (MTU6.TMDR1) register to 1000b to select the reset-synchronized PWM mode. Specify the buffer operation (BFA = 1) of the MTU3.TGRA (MTU6.TGRA) register and the buffer operation (BFB = 1) of the MTU3.TGRB (MTU6.TGRB) register when necessary. Set the MTU4.TMDR1 (MTU7.TMDR1) register to the initial value (normal operation; BFA = 0 and BFB = 0).

[8] Set the TOERA (TOERB) register to enable or disable output from the PWM output pin. Then, set the MTU3.TIORH (MTU6.TIORH), MTU3.TIORL (MTU6.TIORL), MTU4.TIORH (MTU7.TIORH), and MTU4.TIORL (MTU7.TIORL) registers to 00h.

[9] Set the MPU and the port mode register (PMR) of the I/O ports.

[10] Set the TSTRA.CST3 (TSTRB.CST6) bit to 1 to start count operation.

Note: The output waveform starts to toggle at the point of MTU3.TCNT (MTU6.TCNT) = MTU3.TGRA (MTU6.TGRA) = X by setting X = MTU3.TGRA (MTU6.TGRA), i.e., cycle = duty ratio.

Figure 24.44 Procedure for Selecting Reset-Synchronized PWM Mode

(2) Example of Reset-Synchronized PWM Mode Operation

Figure 24.45 shows an example of operation in the reset-synchronized PWM mode.
MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) operate as up-counters. The counters are cleared when a compare match occurs between MTU3.TCNT (MTU6.TCNT) and MTU3.TGRA (MTU6.TGRA), and then begin incrementing from 0000h. The output from the PWM pins toggles every time a compare match occurs in MTU3.TGRB (MTU6.TGRB), MTU4.TGRA (MTU7.TGRA), and MTU4.TGRB (MTU7.TGRB) and the counters are cleared.

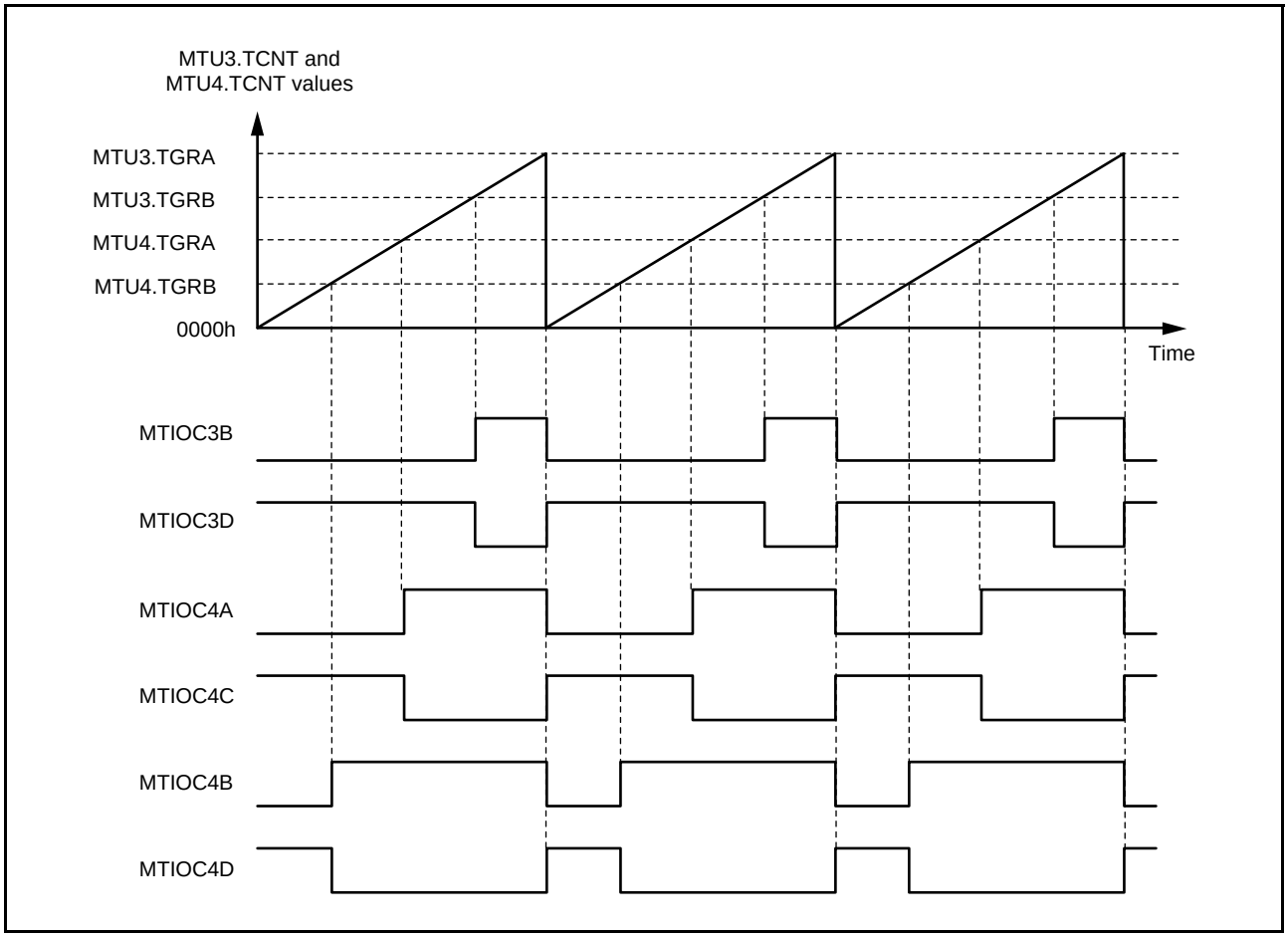


Figure 24.45 Example of Reset-Synchronized PWM Mode Operation (When OLSN = 1 and OLSP = 1 in MTU3.TOCR1 and MTU4.TOCR1)

24.3.8 Complementary PWM Mode

In complementary PWM mode, dead time can be set for PWM waveforms to be output. The dead time is the period during which the upper and lower arm transistors are set to the inactive level in order to prevent short-circuiting of the arms.
Six positive-phase and six negative-phase PWM waveforms (12 phases in total) with dead time can be output by combining MTU3/ MTU4 and MTU6/MTU7. PWM waveforms without dead time can also be output.
In complementary PWM mode, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC7C, and MTIOC7D pins function as PWM output pins, and the MTIOC3A and MTIOC6A pins can be set for toggle output synchronized with the PWM cycle.
MTU3.TCNT, MTU4.TCNT, MTU6.TCNT, and MTU7.TCNT function as up/down-counters.
Table 24.74 shows the PWM output pins used. Table 24.75 shows the settings of the registers used.
A function to directly cut off the PWM output by using an external signal is supported as a port function.

Table 24.74 Output Pins for Complementary PWM Mode

Channel	Output Pin	Description
MTU3	MTIOC3A	Toggle output synchronized with PWM cycle (or I/O port)
	MTIOC3B	PWM output pin 1
	MTIOC3C	I/O port*1
	MTIOC3D	PWM output pin 1' (negative-phase waveform output of PWM output 1)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (negative-phase waveform output of PWM output 1)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (negative-phase waveform output of PWM output 1)
MTU6	MTIOC6A	Toggle output synchronized with PWM cycle (or I/O port)
	MTIOC6B	PWM output pin 4
	MTIOC6C	I/O port*1
	MTIOC6D	PWM output pin 4' (negative-phase waveform output of PWM output 4)
MTU7	MTIOC7A	PWM output pin 5
	MTIOC7C	PWM output pin 5' (negative-phase waveform output of PWM output 5)
	MTIOC7B	PWM output pin 6
	MTIOC7D	PWM output pin 6' (negative-phase waveform output of PWM output 6)

Note 1. Avoid setting the MTIOC3C and MTIOC6C pins as timer I/O pins in complementary PWM mode.

Table 24.75 Register Settings for Complementary PWM Mode (1/2)

Channel	Counter/ Register	Description	Read/Write from CPU
MTU3	TCNT	Starts up-counting from the value set in the dead time register	Maskable by TRWERA setting*1
	TGRA	Set MTU3.TCNT upper limit value (1/2 carrier cycle + dead time)	Maskable by TRWERA setting*1
	TGRB	PWM output 1 compare register	Maskable by TRWERA setting*1
	TGRC	MTU3.TGRA buffer register	Readable/writable
	TGRD	PWM output 1/MTU3.TGRB buffer register	Readable/writable
	TGRE	MTU3.TGRB buffer register B (when double buffer function is used)	Readable/writable
MTU4	TCNT	Starts up-counting after being initialized to 0000h	Maskable by TRWERA setting*1
	TGRA	PWM output 2 compare register	Maskable by TRWERA setting*1
	TGRB	PWM output 3 compare register	Maskable by TRWERA setting*1
	TGRC	PWM output 2/MTU4.TGRA buffer register	Readable/writable
	TGRD	PWM output 3/MTU4.TGRB buffer register	Readable/writable
	TGRE	MTU4.TGRA buffer register B (when double buffer function is used)	Readable/writable
	TGRF	MTU4.TGRB buffer register B (when double buffer function is used)	Readable/writable
MTU6	TCNT	Starts up-counting from the value set in the dead time register	Maskable by TRWERB setting*2
	TGRA	Set MTU6.TCNT upper limit value (1/2 carrier cycle + dead time)	Maskable by TRWERB setting*2
	TGRB	PWM output 4 compare register	Maskable by TRWERB setting*2
	TGRC	MTU6.TGRA buffer register	Readable/writable
	TGRD	PWM output 4/MTU6.TGRB buffer register	Readable/writable
	TGRE	MTU6.TGRB buffer register B (when double buffer function is used)	Readable/writable
MTU7	TCNT	Starts up-counting after being initialized to 0000h	Maskable by TRWERB setting*2
	TGRA	PWM output 5 compare register	Maskable by TRWERB setting*2
	TGRB	PWM output 6 compare register	Maskable by TRWERB setting*2
	TGRC	PWM output 5/MTU7.TGRA buffer register	Readable/writable
	TGRD	PWM output 6/MTU7.TGRB buffer register	Readable/writable
	TGRE	MTU7.TGRA buffer register B (when double buffer function is used)	Readable/writable
	TGRF	MTU7.TGRB buffer register B (when double buffer function is used)	Readable/writable

Note 1. Access can be enabled or disabled according to the setting in TRWERA (timer read/write enable register A).
Note 2. Access can be enabled or disabled according to the setting in TRWERB (timer read/write enable register B).

Table 24.76 Register Settings for Complementary PWM Mode (2/2)

Channel	Counter/ Register	Description	Read/Write from CPU
Timer dead time data register A (TDDRA)		Set MTU4.TCNT and MTU3.TCNT offset value (dead time value)	Maskable by TRWERA setting*1
Timer dead time data register B (TDDRB)		Set MTU7.TCNT and MTU6.TCNT offset value (dead time value)	Maskable by TRWERB setting *2
Timer cycle data register A (TCDRA)		Set MTU4.TCNT upper limit value (1/2 carrier cycle)	Maskable by TRWERA setting*1
Timer cycle data register B (TCDRB)		Set MTU7.TCNT upper limit value (1/2 carrier cycle)	Maskable by TRWERB setting*2
Timer cycle buffer register A (TCBRA)		TCDRA buffer register	Readable/writable
Timer cycle buffer register B (TCBRB)		TCDRB buffer register	Readable/writable
Subcounter A (TCNTSA)		Subcounter A for dead time generation	Read-only
Subcounter B (TCNTSB)		Subcounter B for dead time generation	Read-only
Temporary register 1A (TEMP1A)		PWM output 1/MTU3.TGRB temporary register A	Not readable/writable
Temporary register 1B (TEMP1B)		PWM output 1/MTU3.TGRB temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 2A (TEMP2A)		PWM output 2/MTU4.TGRA temporary register A	Not readable/writable
Temporary register 2B (TEMP2B)		PWM output 2/MTU4.TGRA temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 3A (TEMP3A)		PWM output 3/MTU4.TGRB temporary register A	Not readable/writable
Temporary register 3B (TEMP3B)		PWM output 3/MTU4.TGRB temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 4A (TEMP4A)		PWM output 4/MTU6.TGRB temporary register A	Not readable/writable
Temporary register 4B (TEMP4B)		PWM output 4/MTU6.TGRB temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 5A (TEMP5A)		PWM output 5/MTU7.TGRA temporary register A	Not readable/writable
Temporary register 5B (TEMP5B)		PWM output 5/MTU7.TGRA temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 6A (TEMP6A)		PWM output 6/MTU7.TGRB temporary register A	Not readable/writable
Temporary register 6B (TEMP6B)		PWM output 6/MTU7.TGRB temporary register B (when double buffer function is used)	Not readable/writable

Note 1. Access can be enabled or disabled according to the setting in TRWERA (timer read/write enable register A).
Note 2. Access can be enabled or disabled according to the setting in TRWERB (timer read/write enable register B).

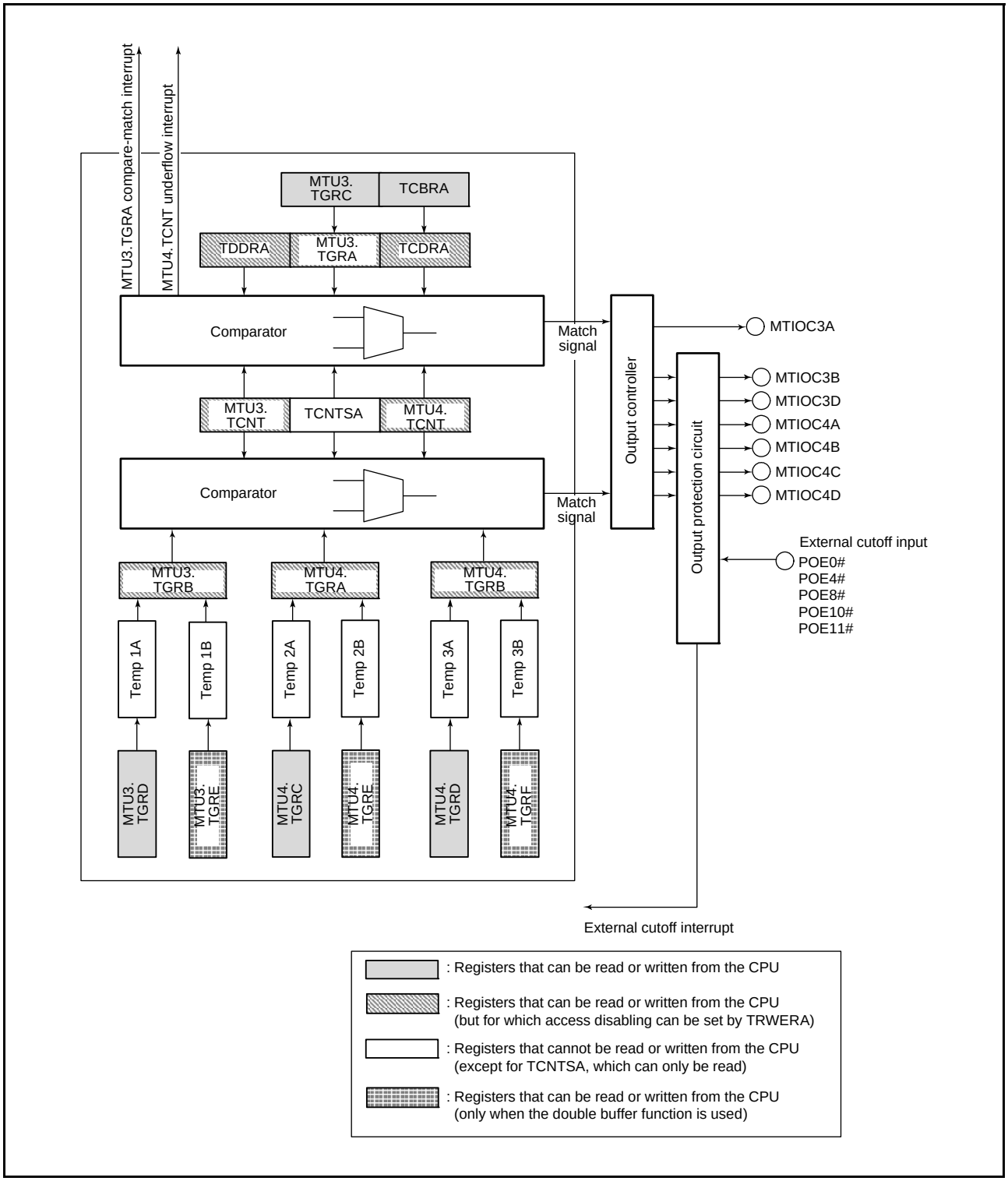


Figure 24.46 Block Diagram of MTU3 and MTU4 in Complementary PWM Mode

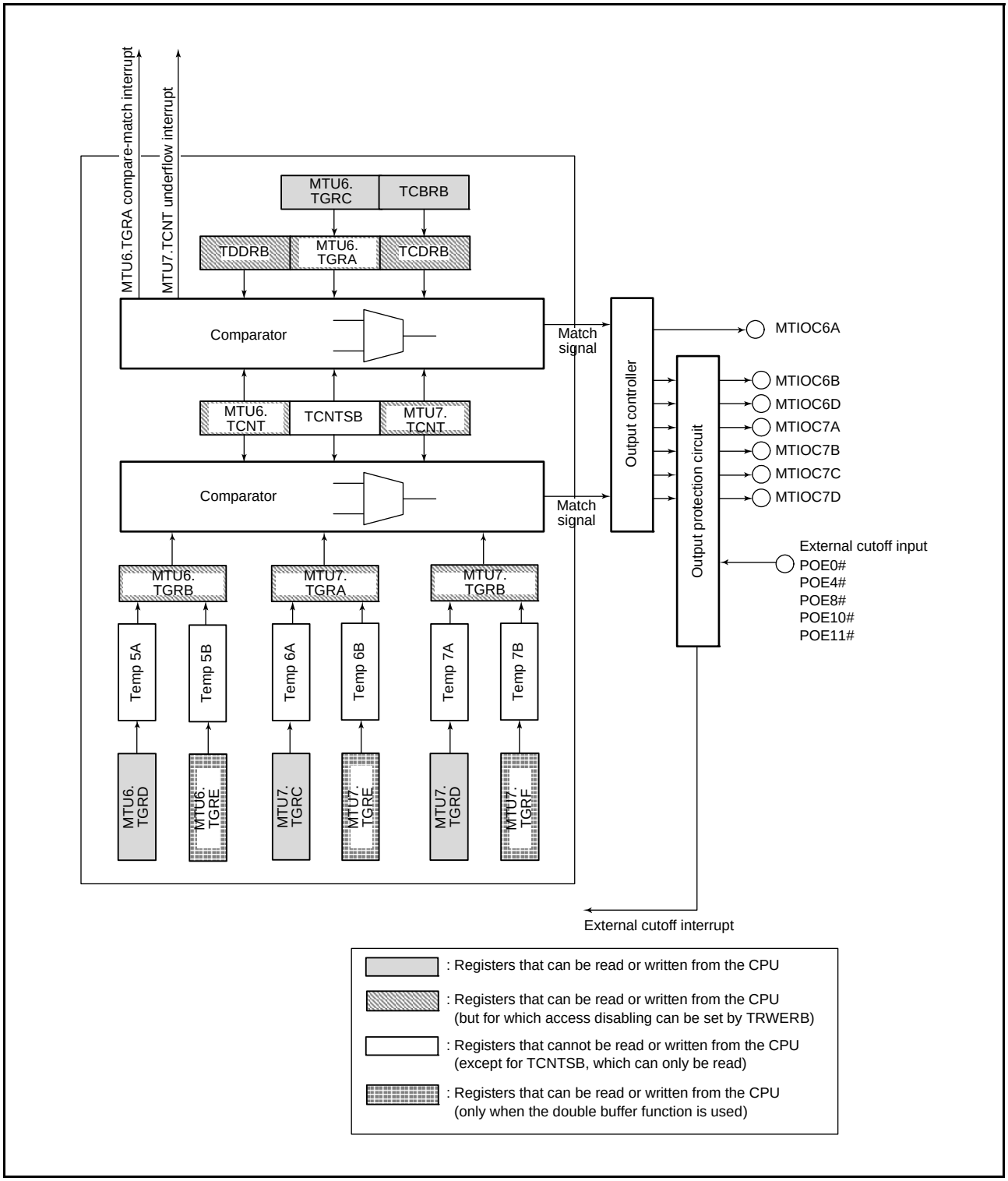
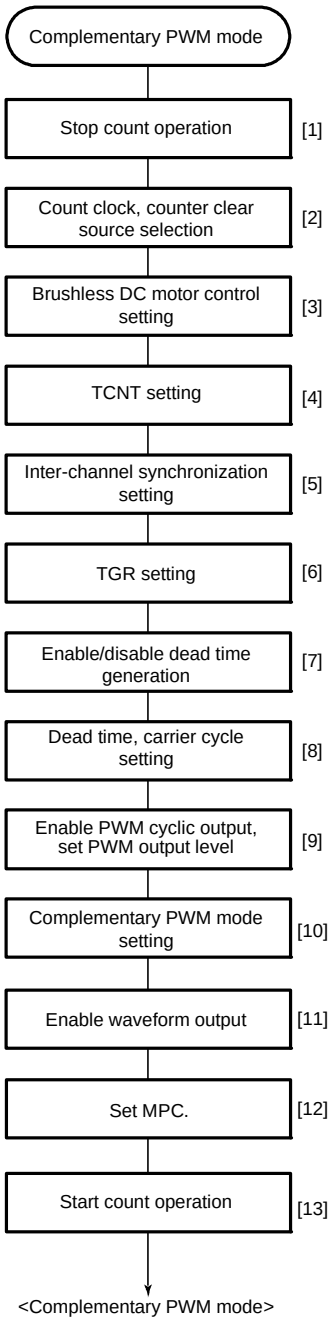


Figure 24.47 Block Diagram of MTU6 and MTU7 in Complementary PWM Mode

(1) Example of Complementary PWM Mode Setting Procedure

Figure 24.48 shows an example of the complementary PWM mode setting procedure.



[1] Clear the TSTRA.CST3 (TSTRB.CST6) and TSTRA.CST4 (TSTRB.CST7) bits to 0 to stop the TCNT count operation.
Specify complementary PWM mode while MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) are stopped.

[2] Set the TCR.TPSC[2:0] and TCR.CKEG[1:0] bits of MTU3 and MTU4 (MTU6 and MTU7) to select the same count clock source and clock edge for MTU3 and MTU4 (MTU6 and MTU7).
Set the TCR.CCLR[2:0] bits only when restarting by synchronous clearing with another channel during complementary PWM operation.

[3] To control a brushless DC motor, set the TGCRA.BDC bit and set the feedback signal input source and output chopping or gate signal direct output (only when MTU3 and MTU4 are used).

[4] Set the dead time in MTU3.TCNT (MTU6.TCNT). Set MTU4.TCNT (MTU7.TCNT) to 0000h.

[5] Make this setting only when restarting by synchronous clearing with another channel during complementary PWM mode operation. In this case, set the TSYRA register (TSYRB) to synchronize the channel generating the synchronous clear with MTU3 and MTU4 (MTU6 and MTU7).

[6] When MTU3 and MTU6 are used, set the PWM output duty ratio in the duty registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) and buffer registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD).
Be sure to set the same value in each pair of the duty registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD) and buffer registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB).
Only when the double buffer function is used, set the PWM output duty value - 1 in the buffer registers (MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF).
When MTU6 and MTU7 are used, set the value to the following registers:
Duty registers (MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB)
Buffer registers (MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD)
Buffer registers when the double buffer function is used (MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF)

[7] This setting is necessary only when no dead time should be generated. Make appropriate settings in the TDERA (TDERB) register so that no dead time generated.

[8] Set the dead time in the TDDRA (TDDRB) register, 1/2 the PWM cycle in the TCDRA (TCDRB) register and TCBRA (TCBRB) register, and 1/2 the PWM cycle plus the dead time in MTU3.TGRA (MTU6.TGRA) and MTU3.TGRC (MTU6.TGRC).
When no dead time generation is selected, set 1 in the TDDRA (TDDRB) register and 1/2 the PWM cycle + 1 in MTU3.TGRA (MTU6.TGRA) and MTU3.TGRC (MTU6.TGRC).

[9] Set the PSYE bit in the TOCR1A (TOCR1B) register to enable or disable toggle output synchronized with the PWM cycle, and set the OLSP and OLSN bits to select the PWM output level.
When specifying the PWM output level using the TOCR2A (TOCR2B) and TOLBRA (TOLBRA) in buffer operation, refer to Timer Output Level Buffer Registers (TOLBRA and TOLBRB).

[10] Select complementary PWM mode in MTU3.TMDR1 (MTU6.TMDR1). Specify the buffer operation (BFA = 1) of MTU3.TGRA (MTU6.TGRA) and the buffer operation (BFB = 1) of MTU3.TGRB (MTU6.TGRB).
Only when the double buffer function is used, set the TMDR2A (TMDR2B) register to enable the double buffer function.

[11] Set the TOERA (TOERB) register to enable or disable output from the PWM output pin. Then, set the MTU3.TIORH (MTU6.TIORH), MTU3.TIORL (MTU6.TIORL), MTU4.TIORH (MTU7.TIORH), and MTU4.TIORL (MTU7.TIORL) to 00h.

[12] Set the MPU and the port mode register (PMR) of the I/O ports.

[13] Set the TSTRA.CST3 (TSTRB.CST6) and TSTRA.CST4 (TSTRB.CST7) bits to 1 simultaneously to start count operation.

Figure 24.48 Example of Complementary PWM Mode Setting Procedure

(2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, six phases (three positive and three negative) PWM waveforms can be output. Figure 24.49 illustrates counter operation in complementary PWM mode (MTU3 and MTU4), and Figure 24.50 shows an example of operation in complementary PWM mode.

(a) Counter Operation

In complementary PWM mode, three counters—MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) —in each unit perform up-/down-count operations.
MTU3.TCNT (MTU6.TCNT) is automatically initialized to the value set in TDDRA (TDDRB) when complementary PWM mode is selected and the CST bit in TSTRA (TSTRB) is 0. When the CST bit is set to 1, MTU3.TCNT (MTU6.TCNT) counts up to the value set in MTU3.TGRA (MTU6.TGRA), then switches to down-counting when it matches MTU3.TGRA (MTU6.TGRA). When the MTU4.TCNT (MTU7.TCNT) value matches 0000h, MTU3.TCNT (MTU6.TCNT) switches to up-counting, and the operation is repeated in this way.
MTU4.TCNT (MTU7.TCNT) should be initialized to 0000h after a reset. When the CST bit is set to 1, MTU4.TCNT (MTU7.TCNT) counts up in synchronization with MTU3.TCNT (MTU6.TCNT), and switches to down-counting when MTU3.TCNT (MTU6.TCNT) matches MTU3.TGRA (MTU6.TGRA). On reaching 0000h, MTU4.TCNT (MTU7.TCNT) switches to up-counting, and the operation is repeated in this way. TCNTSA (TCNTSB) is a read-only counter. It does not need to be initialized after a reset.
In counting up by MTU3.TCNT and MTU4.TCNT (or MTU6.TCNT and MTU7.TCNT), MTU3.TCNT (or MTU6.TCNT) starts counting up when it matches TCDRA (or TCDRB) and switches to counting down when it matches MTU3.TGRA (or MTU6.TGRA). Furthermore, when MTU4.TCNT (or MTU7.TCNT) matches TDDRA (or TDDRB), TCNTSA (or TCNTSB) is set to the value in MTU3.TGRA (or MTU6.TGRA) and counting is stopped.
When MTU4.TCNT (MTU7.TCNT) matches TDDRA (TDDRB) during down-counting of MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT), TCNTSA (TCNTSB) starts up-counting, and when MTU4.TCNT (MTU7.TCNT) matches 0000h, the operation switches to down-counting. When MTU3.TCNT (MTU6.TCNT) matches TCDRA (TCDRB), TCNTSA (TCNTSB) is cleared to 0000h and stops counting.
TCNTSA (TCNTSB) is compared with the compare register and temporary register, in which the PWM duty is specified, only during the count operation.

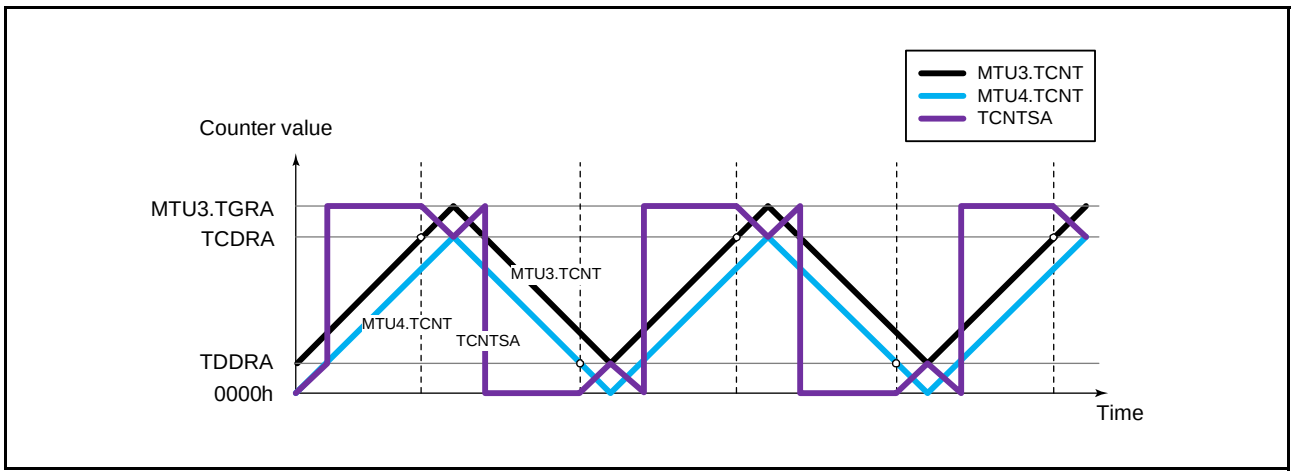


Figure 24.49 Counter Operation in Complementary PWM Mode

(b) Register Operation

In complementary PWM mode, nine registers (compare registers, buffer registers, and temporary registers) are used to control the duty ratio for the PWM output. Figure 24.50 shows an example of operation in complementary PWM mode (MTU3 and MTU4).

MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB (MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB) are constantly compared with the counters to generate PWM waveforms. When these registers match the counter, the value set in the OLSN and OLSP bits in the timer output control registers (TOCR1A and TOCR1B) is output from the PWM output pin.

MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD (MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD) are buffer registers for these compare registers.

When the double buffer function is used, MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF (MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF) are also used as buffer registers B. For details of double buffer operation, refer to section 24.3.8 (2) (s), Double Buffer Function in Complementary PWM Mode.

Data in a compare register can be changed by writing new data to the corresponding buffer register. The buffer registers can be read or written at any time.

When modifying data in a buffer register, write to MTU4.TGRD (MTU7.TGRD) last and enable data transfer from the buffer register to a temporary register. At this time, transfer from the TCBRA (TCBRB) register and MTU3.TGRC (MTU6.TGRC) register, which operate as buffer registers for the timer cycle registers, to temporary registers is also enabled. Data is transferred to all five temporary registers at the same time.

When transfer is enabled in the Ta interval, data written to a buffer register is transferred to the temporary register. The data is not transferred to the temporary register in the Tb1 and Tb2 intervals. Data enabled for transfer in this interval is transferred to the temporary register at the end of this interval.

The value transferred to a temporary register is transferred to the compare register at the end of the Tb1 interval (when matches MTU3.TGRA (MTU6.TGRA) while TCNTSA (TCNTSB) is counting up), or at the end of the Tb2 interval (when matches 0000h while TCNTSA (TCNTSB) is counting down). The timing for transfer from the temporary register to the compare register can be selected with bits MD[3:0] in the timer mode register 1 (TMDR1). Figure 24.50 shows an example in which the trough is selected for the transfer timing.

In the Tb interval in which data is not transferred to the temporary register (Tb1 in Figure 24.50), the temporary register has the same function as the compare register and is compared with the counter. In this interval, therefore, there are two compare match registers for one output phase; the compare register contains the pre-change data and the temporary register contains new data. In this interval, three counters MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) and two registers (compare register and temporary register) are compared, and PWM output is controlled accordingly.

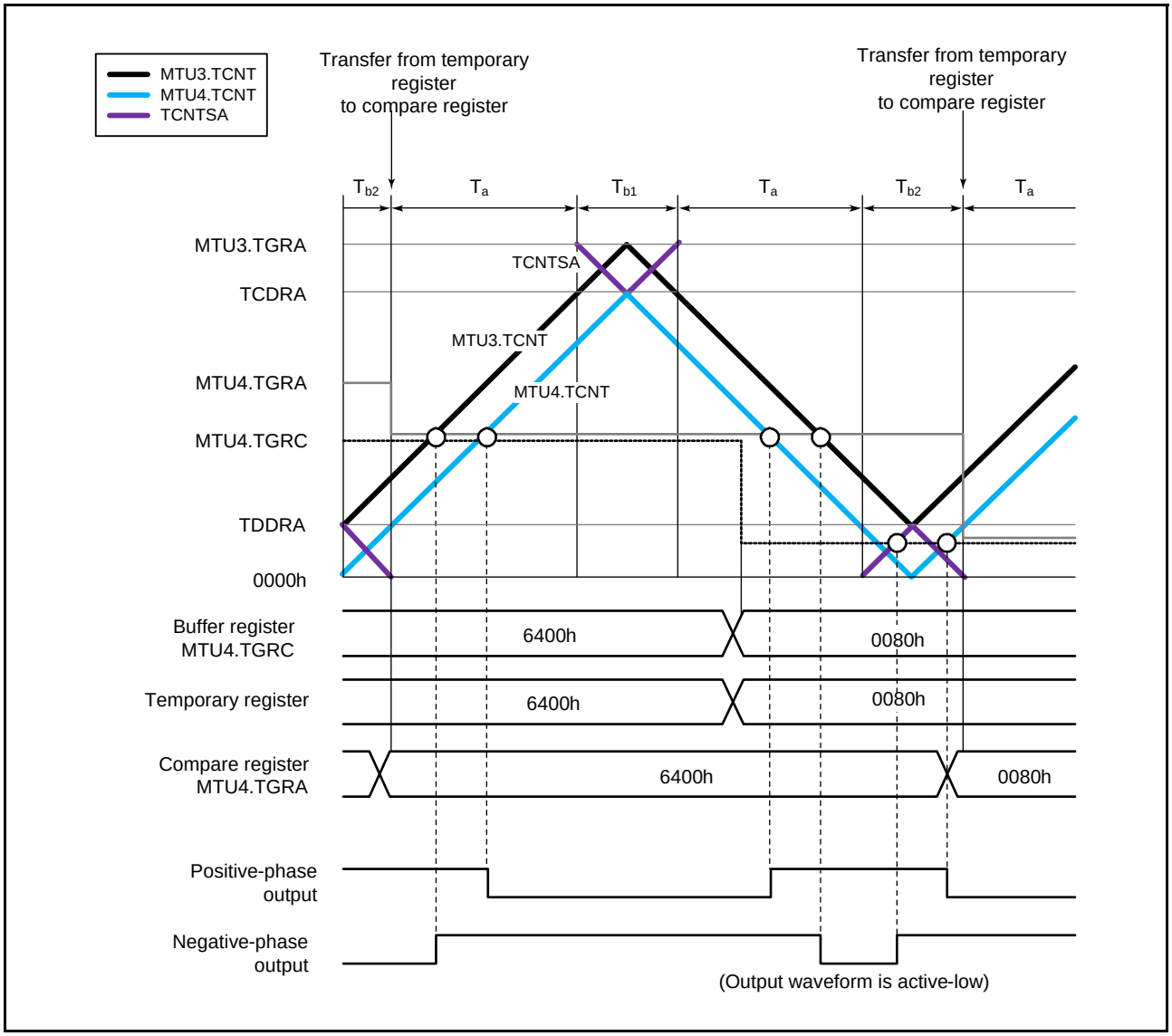


Figure 24.50 Example of Operation in Complementary PWM Mode (MTU3 and MTU4)

(c) Initial Setting

In complementary PWM mode, there are nine registers that require initial setting. In addition, there is a register that specifies whether to generate dead time (it should be used only when dead time generation should be disabled).
Before setting complementary PWM mode with MTU3.TMDR1.MD[3:0] (MTU6.TMDR1.MD[3:0]) bits, initial values should be set in the following registers.
The TOCR1A, TOCR2A, TOCR1B, and TOCR2B registers are used to set the PWM output level. MTU3.TGRC (MTU6.TGRC) operates as the buffer register for MTU3.TGRA (MTU6.TGRA), and should be set with 1/2 the PWM cycle + dead time Td. The timer cycle buffer register (TCBRA or TCBRB) operates as the buffer register for the timer cycle data register (TCDRA or TCDRB), and should be set with 1/2 the PWM cycle. Set dead time Td in the timer dead time data register (TDDRA or TDDRB).
When dead time is not needed, the TDER bit in the timer dead time enable register (TDERA or TDERB) should be cleared to 0, MTU3.TGRC and MTU3.TGRA (MTU6.TGRC and MTU6.TGRA) should be set to 1/2 the PWM carrier cycle + 1, and TDDRA (TDDRB) should be set to 1.
Set the respective initial PWM duty values in three buffer registers A (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD (MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD)).
Set the respective (initial PWM duty – 1) values in three buffer registers B (MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF (MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF)) only when the double buffer function is used.
The values set in the five buffer registers excluding TDDRA (TDDRB) are transferred to the corresponding compare registers as soon as complementary PWM mode is set.
Set MTU4.TCNT (MTU7.TCNT) to 0000h before setting complementary PWM mode.

Table 24.77 Registers and Counters Requiring Initial Setting

Register and Counter	Setting
TOCR1A, TOCR2A, TOCR1B, TOCR2B	PWM output level
MTU3.TGRC MTU6.TGRC	1/2 PWM cycle + dead time Td (1/2 PWM cycle + 1 when dead time generation is disabled by TDERA or TDERB)
TDDRA, TDDRB	Dead time Td (1 when dead time generation is disabled by TDERA or TDERB)
TCBRA, TCBRB	1/2 PWM cycle
MTU3.TGRD, MTU4.TGRC, MTU4.TGRD MTU6.TGRD, MTU7.TGRC, MTU7.TGRD	Initial PWM duty ratio value for each phase
MTU3.TGRE, MTU4.TGRE, MTU4.TGRF MTU6.TGRE, MTU7.TGRE, MTU7.TGRF	Initial PWM duty ratio – 1 value for each phase (only when double buffer function is used)
MTU4.TCNT MTU7.TCNT	0000h

Note: The value set in MTU3.TGRC (MTU6.TGRC) should be the sum of 1/2 the PWM cycle set in TCBRA (TCBRB) and dead time Td set in TDDRA (TDDRB). When dead time generation is disabled by TDERA (TDERB), TGRC should be set to 1/2 the PWM cycle + 1.

(d) PWM Output Level Setting

In complementary PWM mode, the PWM output level is set with bits OLSN and OLSP in timer output control register 1 (TOCR1A or TOCR1B) or bits OLS1P to OLS3P and OLS1N to OLS3N in timer output control register 2 (TOCR2A or TOCR2B).
The output level can be set for each of the three positive phases and three negative phases of 6-phase output.
Complementary PWM mode should be cleared before setting or changing output levels.

(e) Dead Time Setting

In complementary PWM mode, dead time can be set for PWM output.
The dead time is set in the timer dead time data register (TDDRA or TDDRB). The value set in TDDRA (TDDRB) is used as the MTU3.TCNT (MTU6.TCNT) counter start value and creates a non-overlapping interval between MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT). Complementary PWM mode should be cleared before changing the contents of TDDRA (TDDRB).

(f) Dead Time Suppressing

Dead time generation is suppressed by clearing the TDER bit in the timer dead time enable register (TDERA or TDERB) to 0. TDERA (TDERB) can be cleared to 0 only when 0 is written to it after reading TDER = 1.
MTU3.TGRA and MTU3.TGRC (MTU6.TGRA and MTU6.TGRC) should be set to 1/2 PWM cycle + 1 and the timer dead time data register (TDDRA or TDDRB) should be set to 1.
By the above settings, PWM waveforms without dead time can be obtained. Figure 24.51 shows an example of operation without dead time (MTU3 and MTU4).

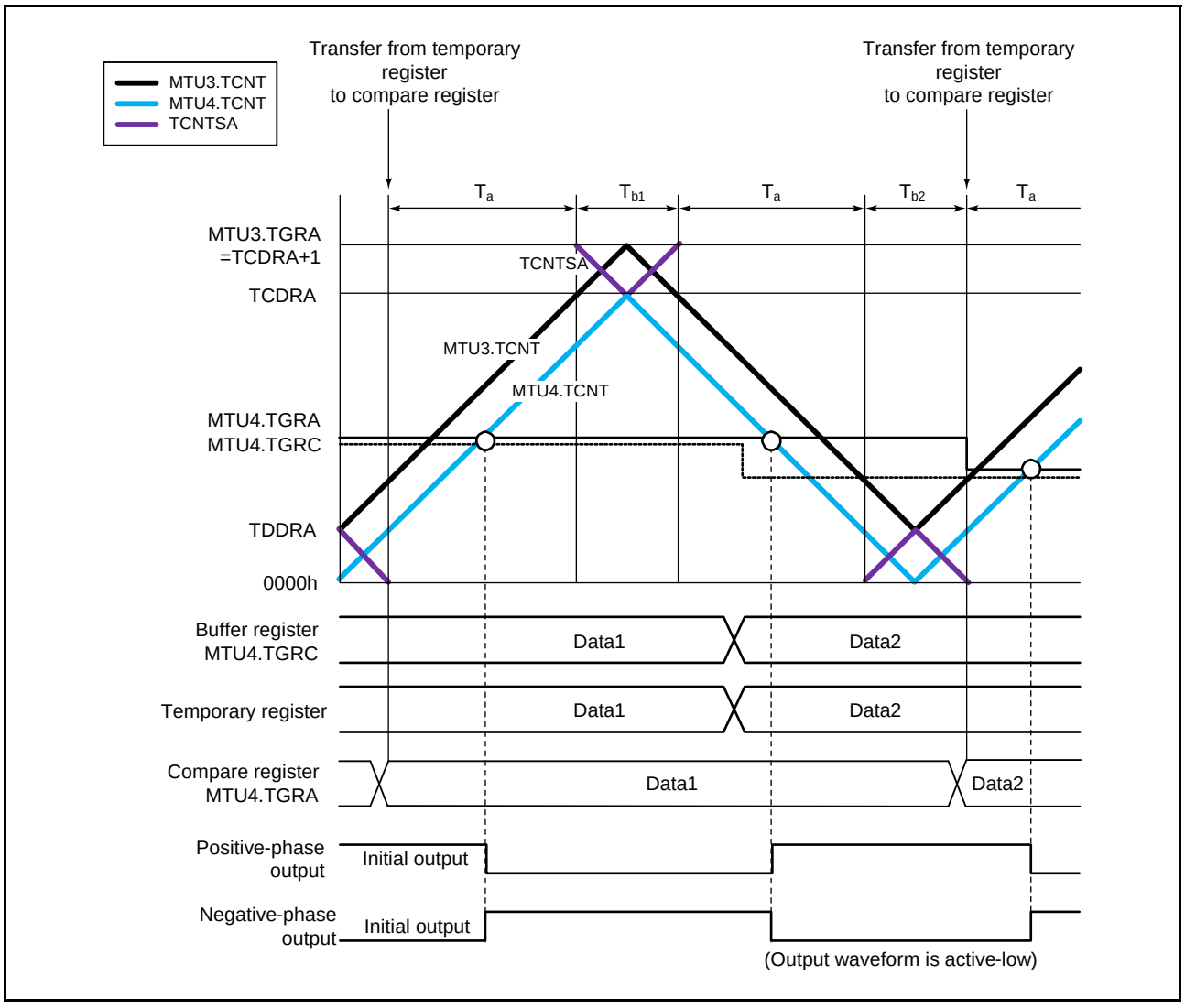


Figure 24.51 Example of Operation without Dead Time (MTU3 and MTU4)

(g) PWM Cycle Setting

In complementary PWM mode, the PWM cycle is set in two registers—MTU3.TGRA (MTU6.TGRA), in which the MTU3.TCNT (MTU6.TCNT) upper limit value is set, and TCDRA (TCDRB), in which the MTU4.TCNT (MTU7.TCNT) upper limit value is set. The settings should be made so as to achieve the following relationship between these two registers:

With dead time: MTU3.TGRA (MTU6.TGRA) setting = TCDRA (TCDRB) setting + TDDRA (TDDRB) setting
Without dead time: MTU3.TGRA (MTU6.TGRA) setting = TCDRA (TCDRB) setting + 1

In addition, the settings should be made so as to achieve the following relationship between the TCDRA (TCDRB) register and the TDDRA (TDDRB) register:

TCDRA (TCDRB) setting > TDDRA (TDDRB) setting × 2 + 2

The MTU3.TGRA and TCDRA (MTU6.TGRA and TCDRB) settings are made by setting values in buffer registers MTU3.TGRC and TCBRA (MTU6.TGRC and TCBRB). When data is written to MTU4.TGRD (MTU7.TGRD) to enable transfers, the values set in MTU3.TGRC and TCBRA (MTU6.TGRC and TCBRB) are transferred simultaneously to the MTU3.TGRA and TCDRA (MTU6.TGRA and TCDRB) with the transfer timing selected with the MTU3.TMDR1.MD[3:0] (MTU6.TMDR1.MD[3:0]) bits.
The new PWM cycle is reflected from the next cycle when data is updated at the crest, or from the current cycle when updated in the trough. Figure 24.52 illustrates the operation when the PWM cycle is updated at the crest.
Refer to the following section, (h), Register Data Updating, for the method of updating the data in each buffer register.

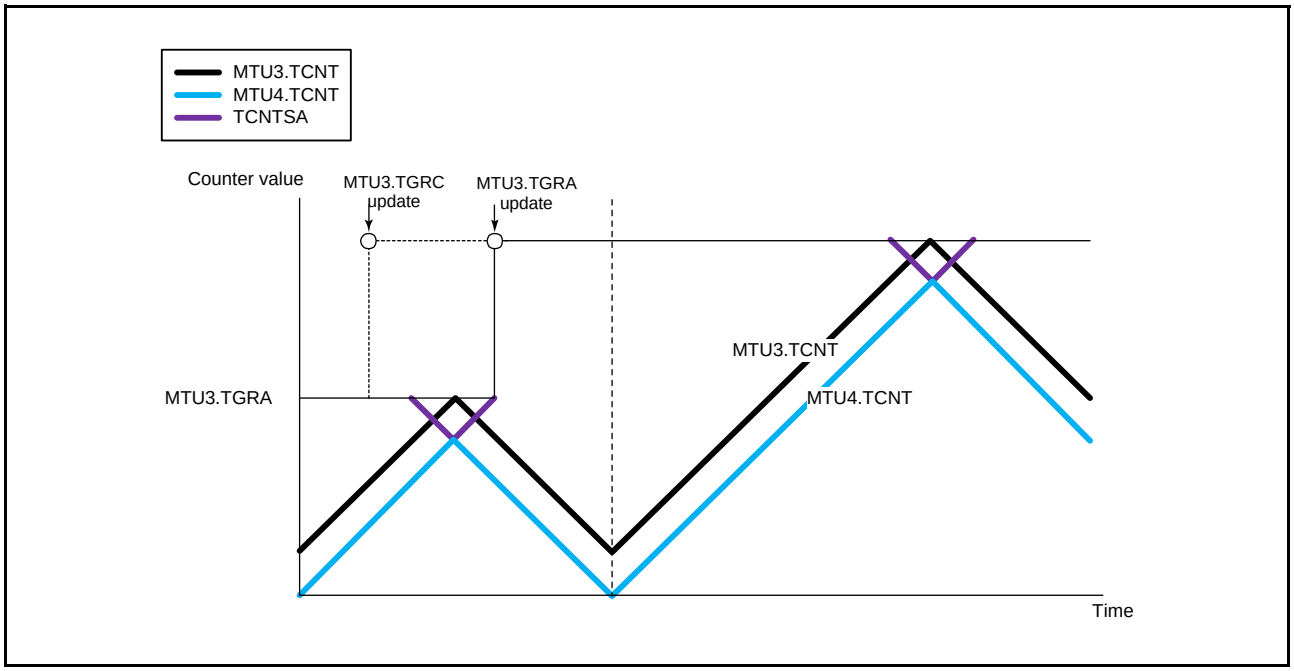


Figure 24.52 Example of PWM Cycle Updating (MTU3 and MTU4)

(h) Register Data Updating

In complementary PWM mode, the buffer register is used to update the data in a compare register. The update data can be written to the buffer register at any time. There are five registers (PWM duty and PWM cycle registers) that have buffer registers and can be updated during operation.

There is a temporary register between each of these registers and its buffer register. While subcounter TCNTSA (TCNTSB) is not counting, if buffer register data is updated, the temporary register value also changes. Data is not transferred from buffer registers to temporary registers while TCNTSA (TCNTSB) is counting; in this case, the value written to a buffer register is transferred after TCNTSA (TCNTSB) halts.

The temporary register value is transferred to the compare register at the data update timing set with MTU3.TMDR1.MD[3:0] (MTU6.TMDR1.MD[3:0]) bits. Figure 24.53 shows an example of data updating in complementary PWM mode (MTU3 and MTU4). This example shows the mode in which data is updated at both the counter crest and trough.

When updating buffer register data, be sure to write to MTU4.TGRD (MTU7.TGRD) at the end of the update. Data is transferred from buffer registers to the temporary registers simultaneously for all five registers after the write to MTU4.TGRD (MTU7.TGRD).

Even when not updating all five registers or when not updating the MTU4.TGRD (MTU7.TGRD) data, be sure to write to MTU4.TGRD (MTU7.TGRD) after writing data to the registers to be updated. In this case, the data written to MTU4.TGRD (MTU7.TGRD) should be the same as the data prior to the write operation.

Refer to section 24.3.8 (2) (s), Double Buffer Function in Complementary PWM Mode, for data updating when the double buffer function is used.

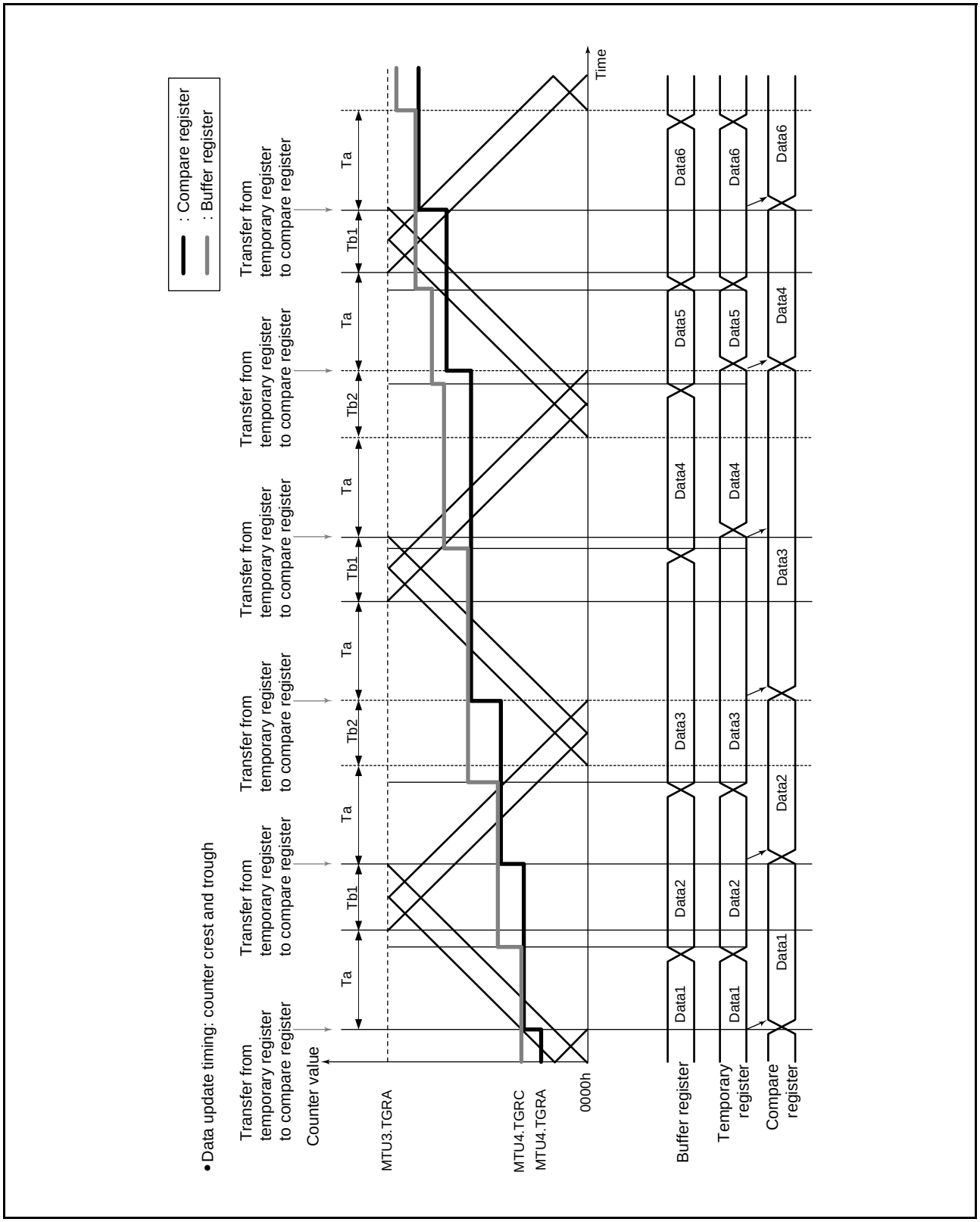


Figure 24.53 Example of Data Updating in Complementary PWM Mode (MTU3 and MTU4)

(i) Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of the OLSN and OLSP bits in the TOCR1A (TOCR1B) register or the OLS1N to OLS3N and OLS1P to OLS3P bits in the TOCR2A register (TOCR2B). This initial output is the non-active level of the PWM output and continues from when complementary PWM mode is set with the MTU3.TMDR1 (MTU6.TMDR1) until MTU4.TCNT (MTU7.TCNT) exceeds the value set in the TDDRA (TDDRB) register. Figure 24.54 shows an example of the initial output in complementary PWM mode.
An example of the waveform when the initial PWM duty ratio value is smaller than the TDDRA (TDDRB) value is shown in Figure 24.55.

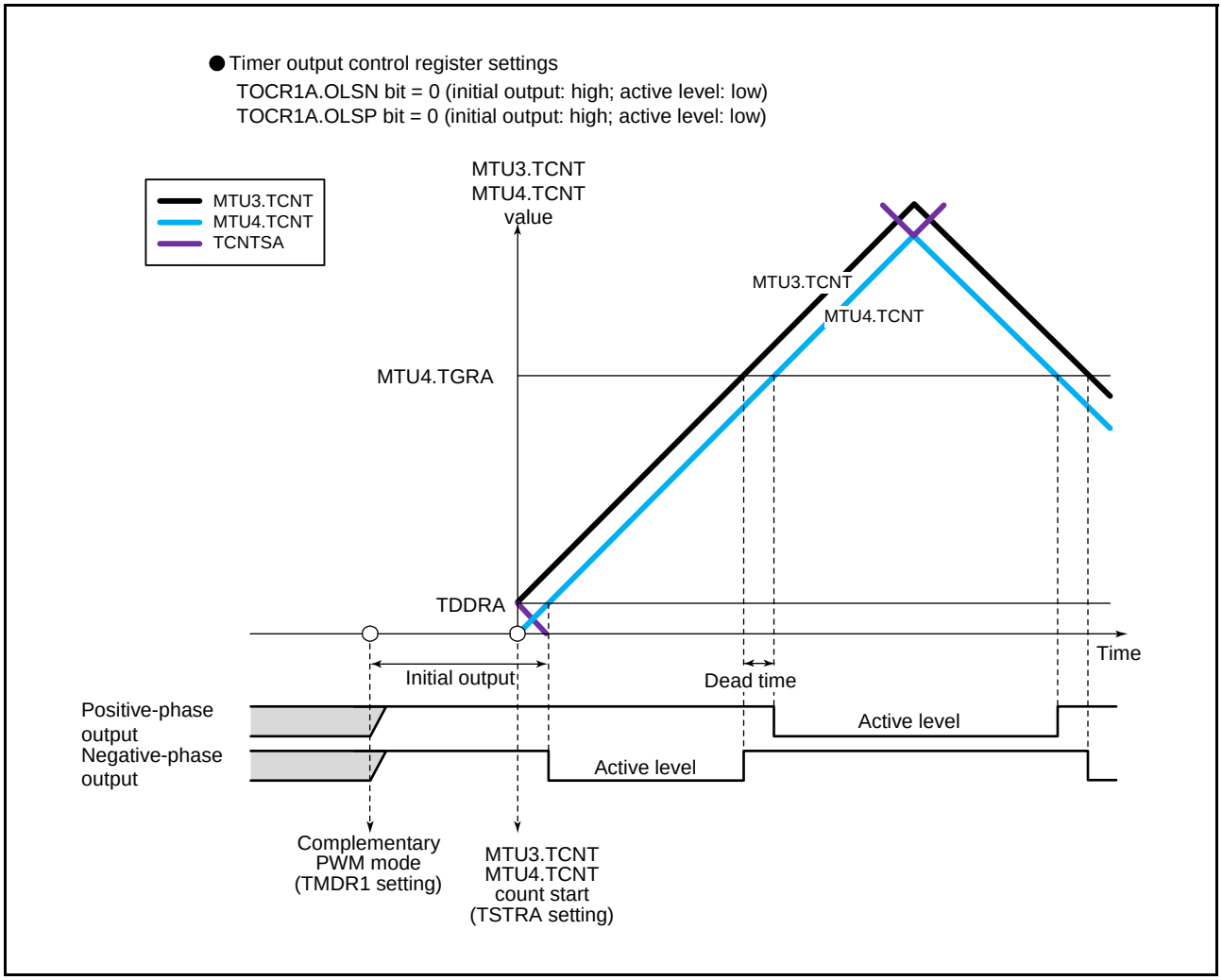


Figure 24.54 Example of Initial Output in Complementary PWM Mode (MTU3 and MTU4) (1))

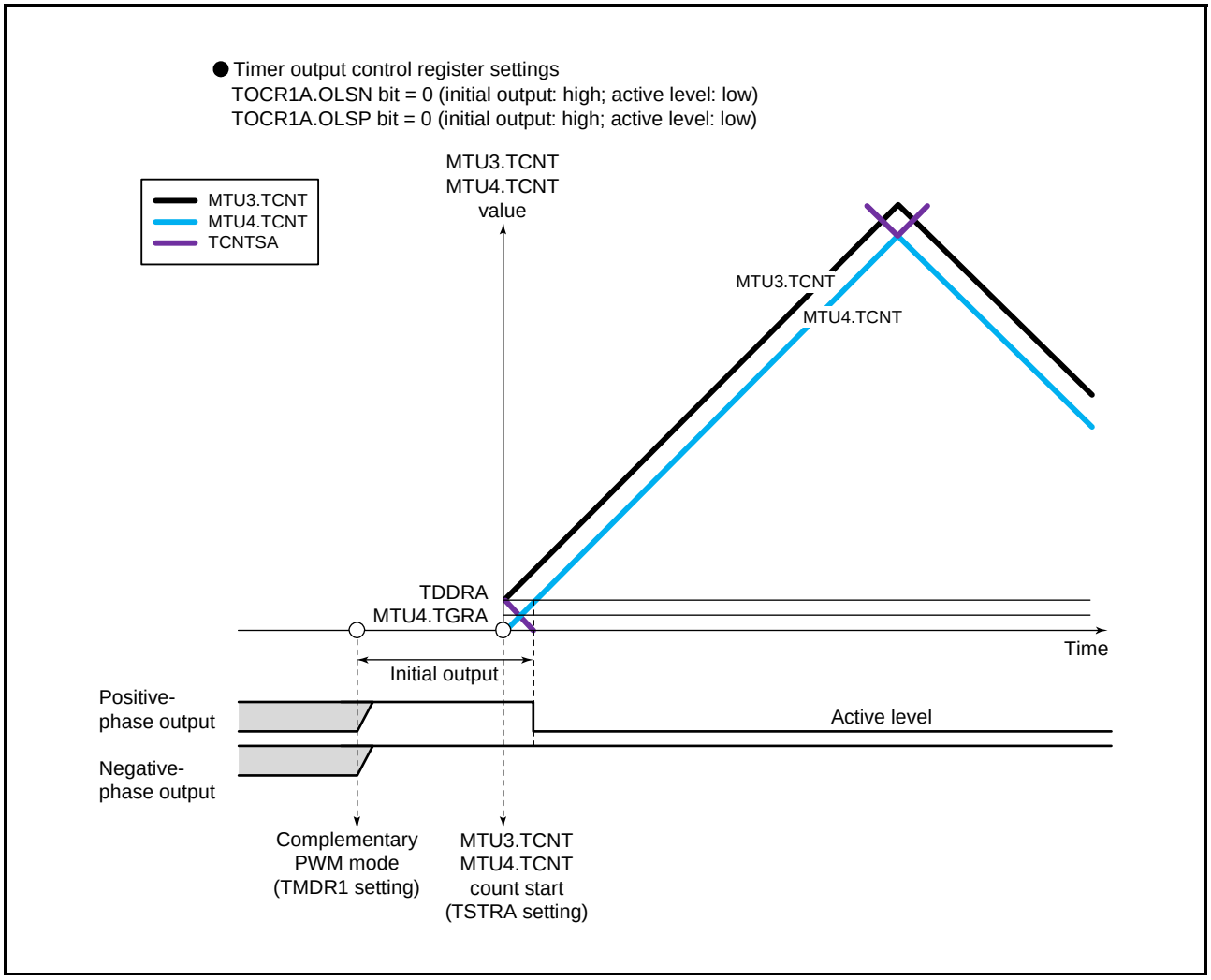


Figure 24.55 Example of Initial Output in Complementary PWM Mode (MTU3 and MTU4) (2)

(j) Method for Generating PWM Output in Complementary PWM Mode

In complementary PWM mode, six phases (three positive and three negative) PWM waveforms can be output. Dead time can be set for PWM waveforms to be output.

A PWM waveform is generated by output of the level selected in the timer output control register in the event of a compare match between a counter and a compare register. While TCNTSA (TCNTSB) is counting, the compare register and temporary register values are simultaneously compared to create consecutive PWM output from 0 to 100% duty ratio. The relative timing of turn-on and turn-off compare match occurrence may vary, but the compare match that turns off each phase takes precedence to secure the dead time and ensure that the positive-phase and negative-phase turn-on times do not overlap. Figure 24.56 to Figure 24.58 show examples of waveform generation in complementary PWM mode.

The positive-phase and negative-phase turn-off timing is generated by a compare match with the counter indicated by a solid line, and the turn-on timing is generated by a compare match with the counter indicated by a dotted line, which operates with a delay equal to the dead time behind the counter indicated by a solid line. In the T1 period, compare match a that turns off the negative phase has the highest priority, and compare matches before a are ignored. In the T2 period, compare match c that turns off the positive phase has the highest priority, and compare matches before c are ignored.

In most cases, compare matches occur in the order a → b → c → d (or c → d → a' → b') as shown in Figure 24.56.

If compare matches deviate from the a → b → c → d order, since the time for which the negative phase is off is shorter than twice the dead time, the positive phase is not turned on. If compare matches deviate from the c → d → a' → b' order, since the time for which the positive phase is off is shorter than twice the dead time, the negative phase is not turned on.

As shown in Figure 24.57, if compare match c follows compare match a before compare match b, compare match b is ignored and the negative phase is turned on by compare match d. This is because turning off the positive phase has priority due to the occurrence of compare match c (positive-phase off timing) before compare match b (positive-phase on timing) (consequently, the waveform does not change because the positive phase goes from off to off).

Similarly, in the example in Figure 24.58, turning off the negative phase has priority due to the occurrence of compare match a' (negative-phase off timing) before compare match d (negative-phase on timing). As a result, the negative phase is not turned on.

Thus, in complementary PWM mode, compare matches at turn-off timings take precedence, and turn-on timing compare matches that occur before a turn-off timing compare match are ignored.

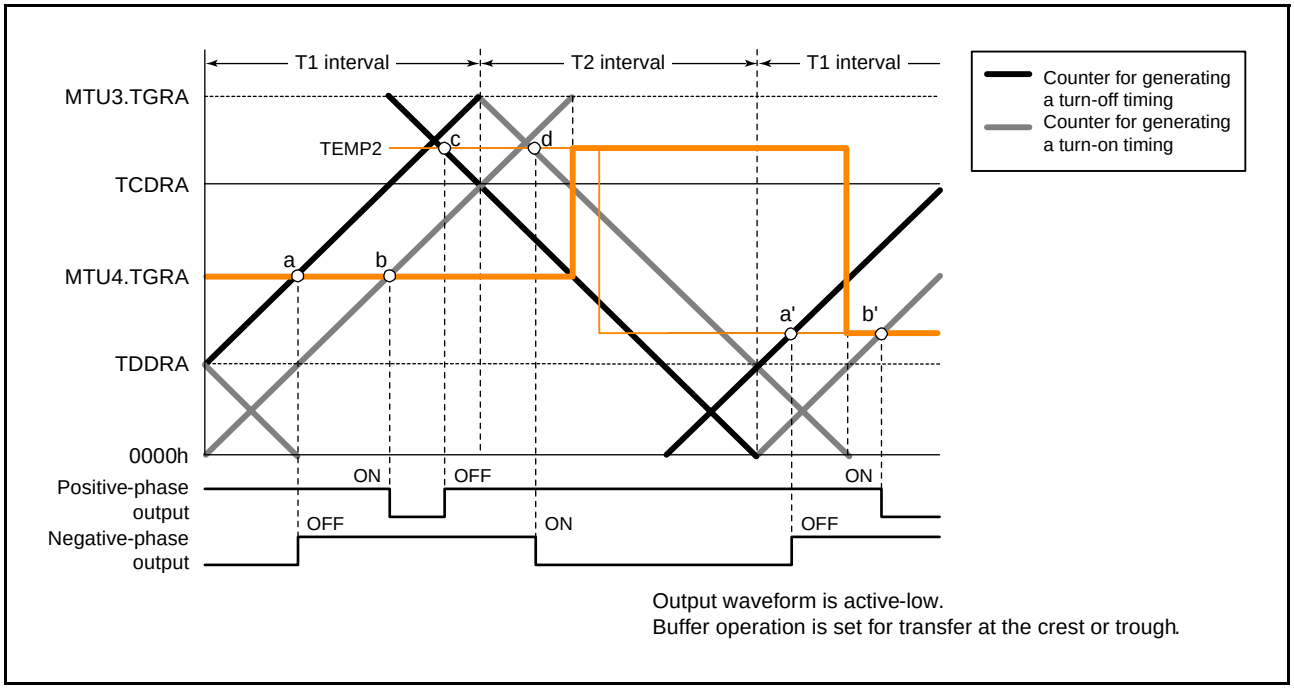


Figure 24.56 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (1)

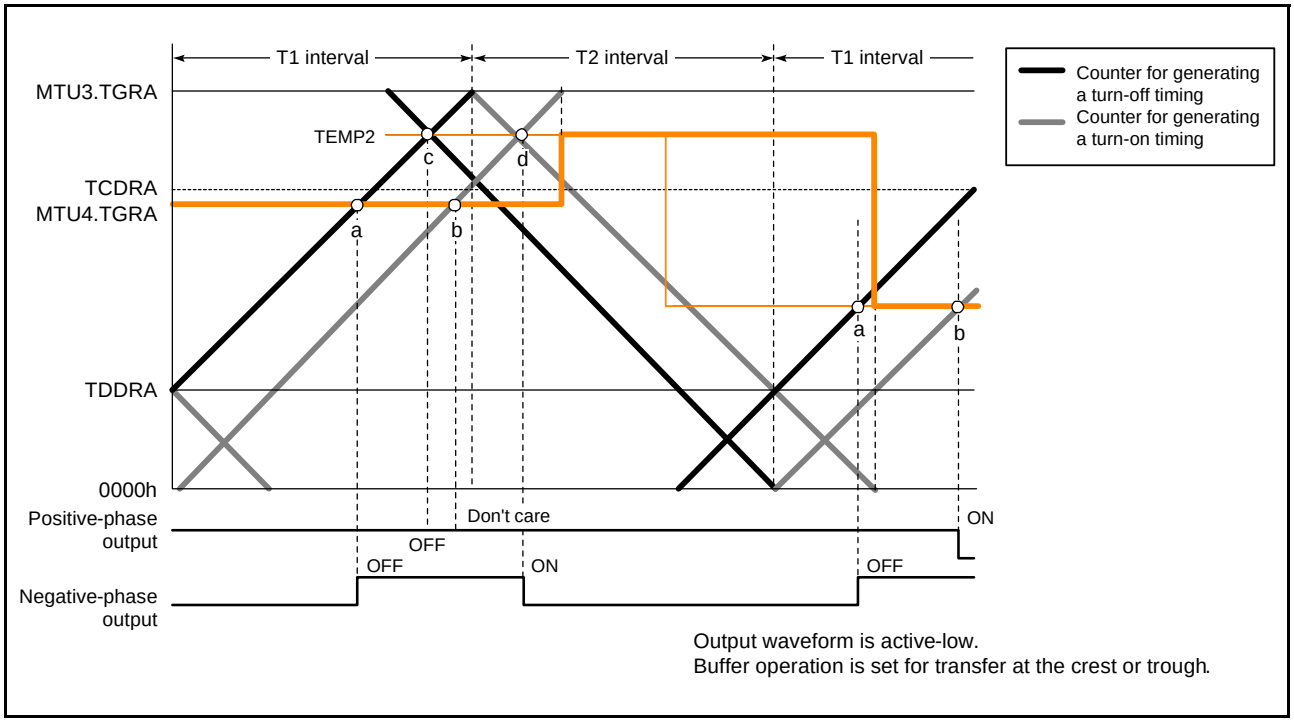


Figure 24.57 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (2)

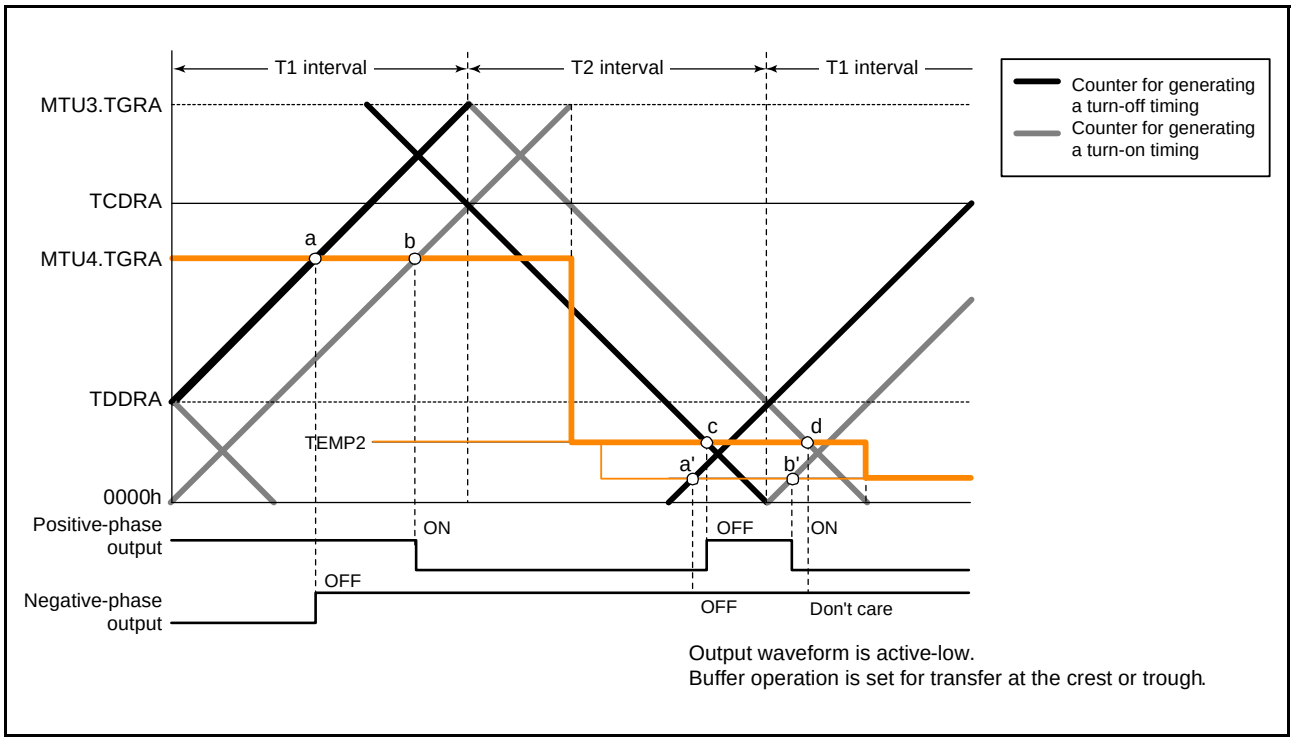


Figure 24.58 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (3)

(k) 0% and 100% Duty Ratio Output in Complementary PWM Mode

In complementary PWM mode, 0% and 100% duty PWM output can be output as required. Figure 24.59 to Figure 24.63 show output examples.

A 100% duty waveform is output when the compare register value is set to 0000h. The waveform in this case has a positive phase with a 100% on-state. A 0% duty waveform is output when the compare register value is set to the same value as MTU3.TGRA (MTU6.TGRA). The waveform in this case has a positive phase with a 100% off-state.

Turn-on and turn-off compare matches occur simultaneously, but if a turn-on compare match and turn-off compare match for the same phase occur simultaneously, both compare matches are ignored and the waveform does not change.

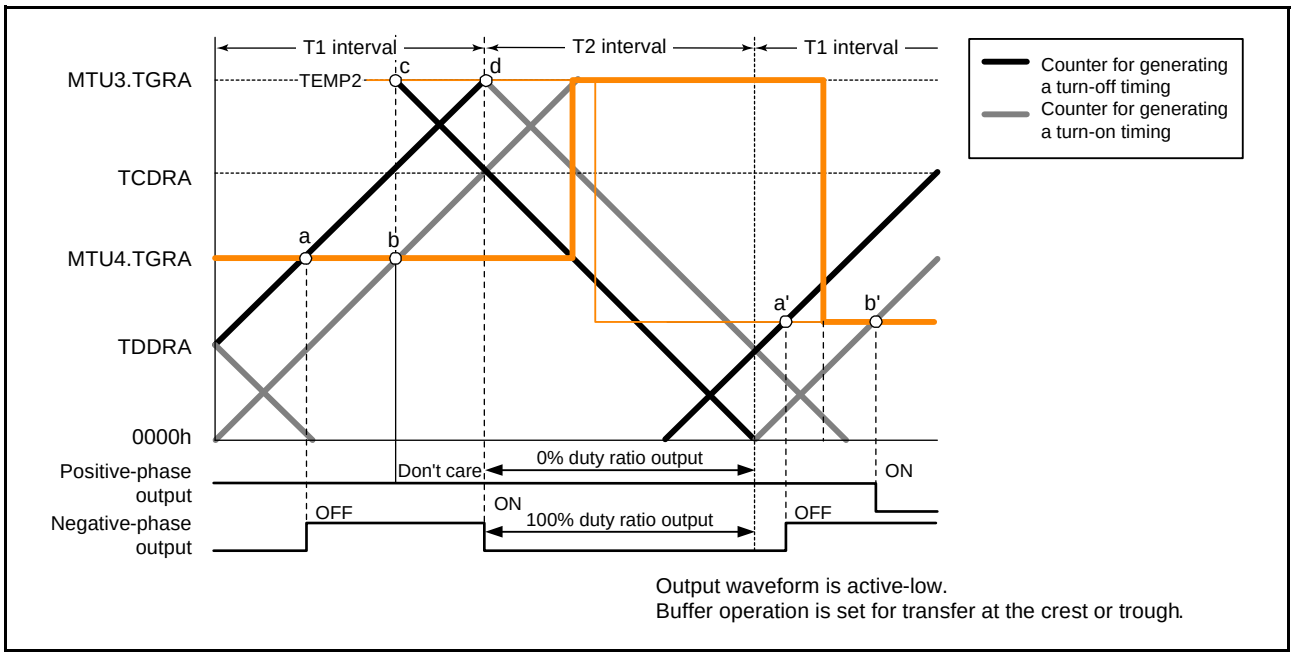


Figure 24.59 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (1)

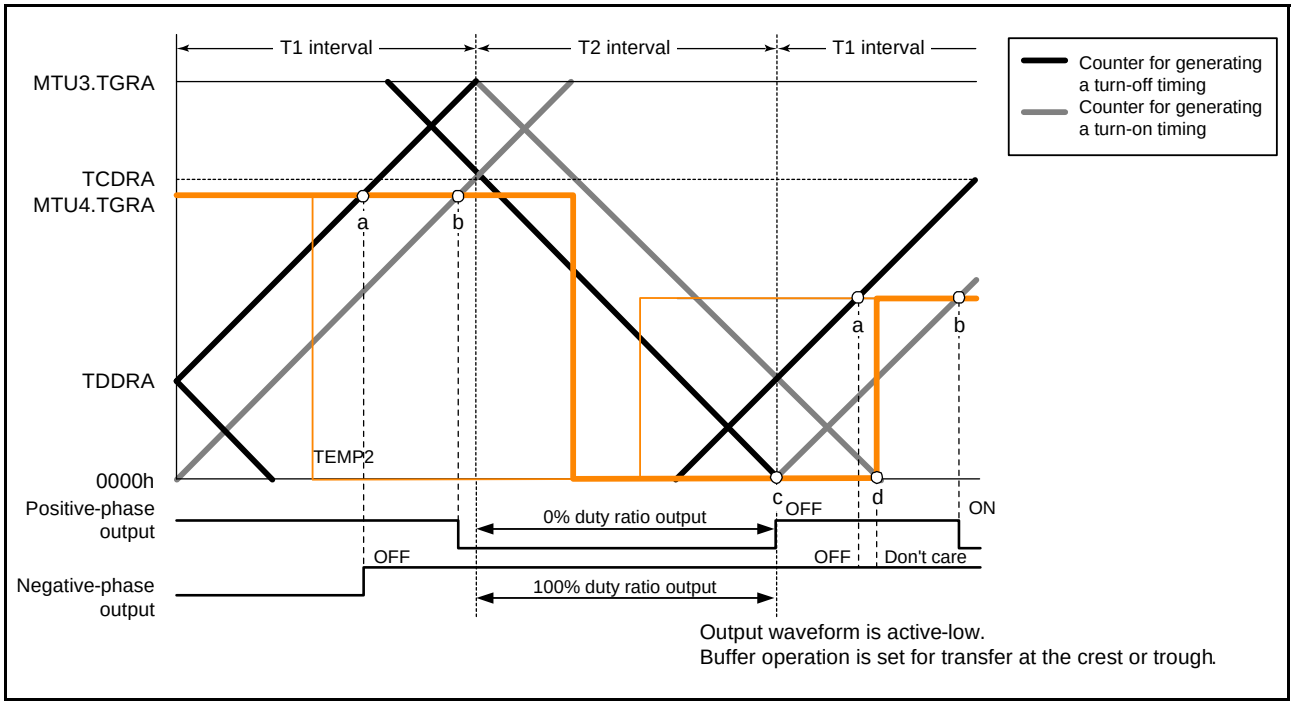


Figure 24.60 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (2)

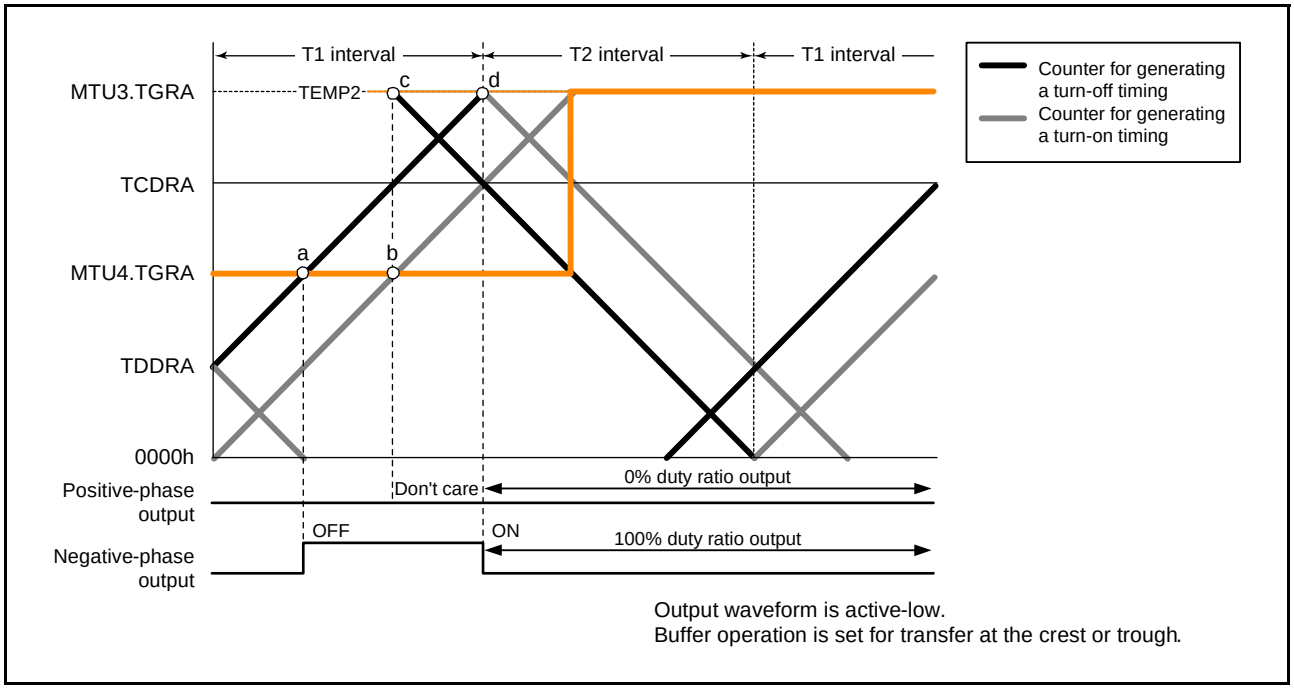


Figure 24.61 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (3)

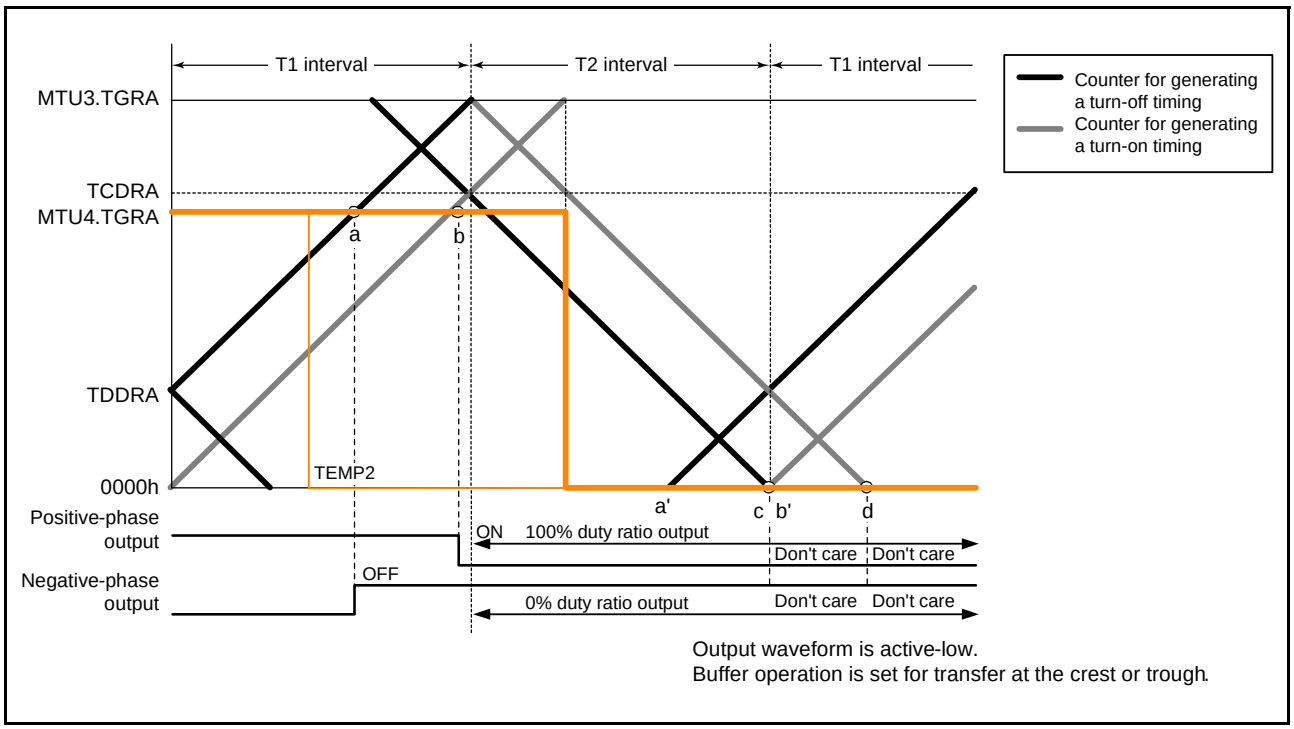


Figure 24.62 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (4)

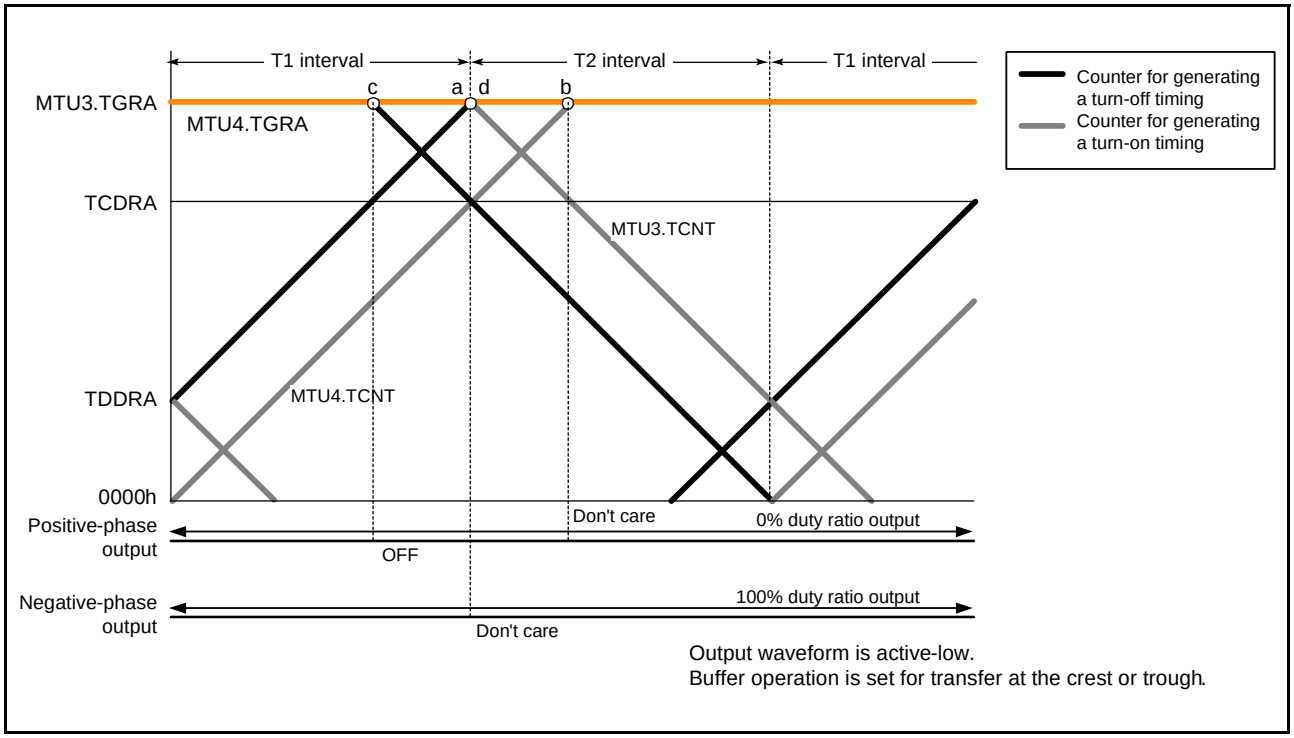


Figure 24.63 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (5)

(l) Toggle Output Synchronized with PWM Cycle

In complementary PWM mode, toggle output from the PWM output pin in synchronization with the PWM cycle can be enabled by setting the PSYE bit in the TOCR1A (TOCR1B) register to 1. An example of a toggle output waveform is shown in Figure 24.64.

This output is toggled by a compare match between MTU3.TCNT and MTU3.TGRA (MTU6.TCNT and MTU6.TGRA) and a compare match between MTU4.TCNT (MTU7.TCNT) and 0000h.

The MTIOC3A (MTIOC6A) pin is assigned for this toggle output. The initial output is high-level output.

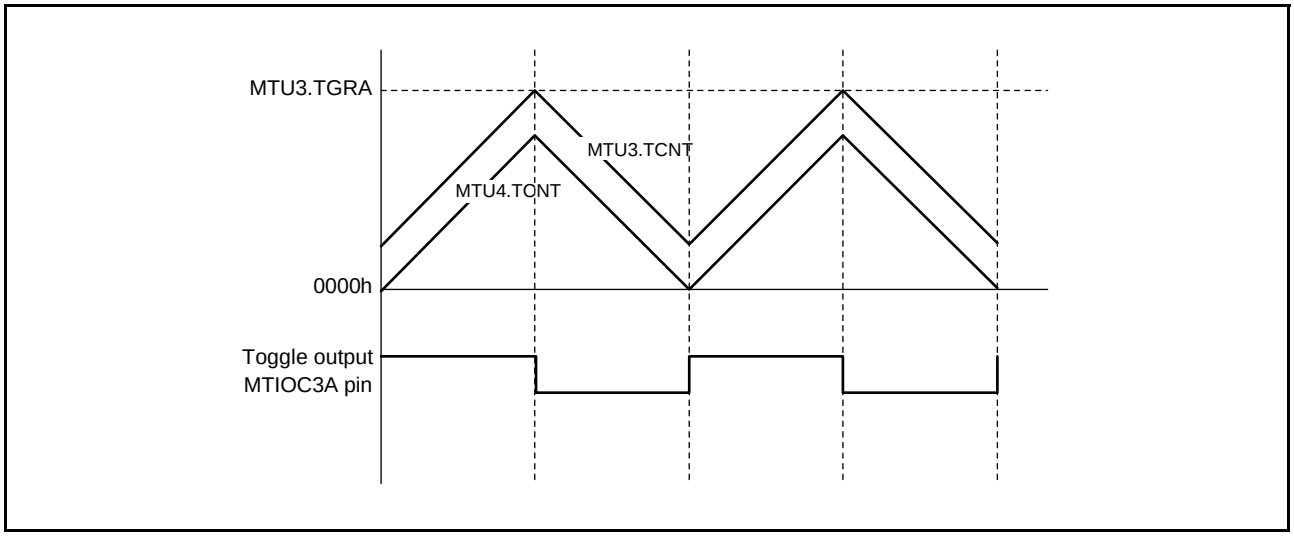


Figure 24.64 Example of Toggle Output Waveform Synchronized with PWM Output (MTU3 and MTU4)

(m) Counter Clearing by Another Channel

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) can be cleared by another channel source when a mode for synchronization with another channel is specified through the TSYRA (TSYRB) register and synchronous clearing is selected with MTU3.TCR.CCLR[2:0] (MTU6.TCR.CCLR[2:0]) bits.

Figure 24.65 illustrates an example of this operation.

Use of this function enables a counter to be cleared and restarted through an external signal.

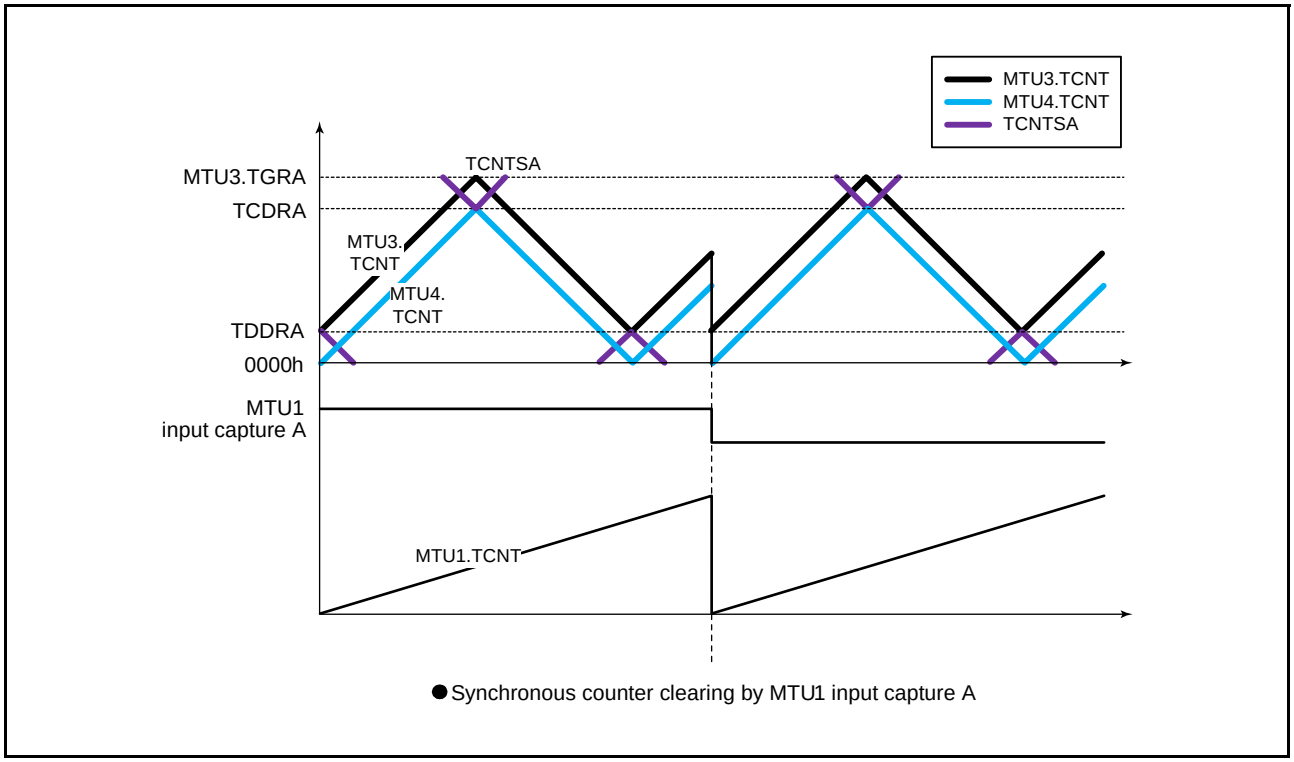


Figure 24.65 Counter Clearing Synchronized with Another Channel (MTU3 and MTU4)

(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the WRE bit in TWCRA (TWCRB) to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval (Tb2 interval) at the trough in complementary PWM mode and controls abrupt change in duty cycle at synchronous counter clearing.

Initial output suppression through the WRE bit = 1 is applicable only when synchronous clearing occurs in the Tb2 interval as indicated by (10) or (11) in Figure 24.66. When synchronous clearing occurs outside that interval, the initial value specified by the OLSN and OLSP bits in TOCR1A (TOCR1B) is output. Even in the Tb2 interval, if synchronous clearing occurs in the initial value output period (indicated by (1) in Figure 24.66) immediately after the counters start operation, initial value output is not suppressed.

This function can be used in both channel combinations of MTU3 and MTU4, and MTU6 and MTU7. In MTU3 and MTU4, synchronous clearing in any of MTU0 to MTU2 can cause counter clearing; in MTU6 and MTU7,compare match or input capture in any of MTU0 to MTU2 can cause counter clearing.

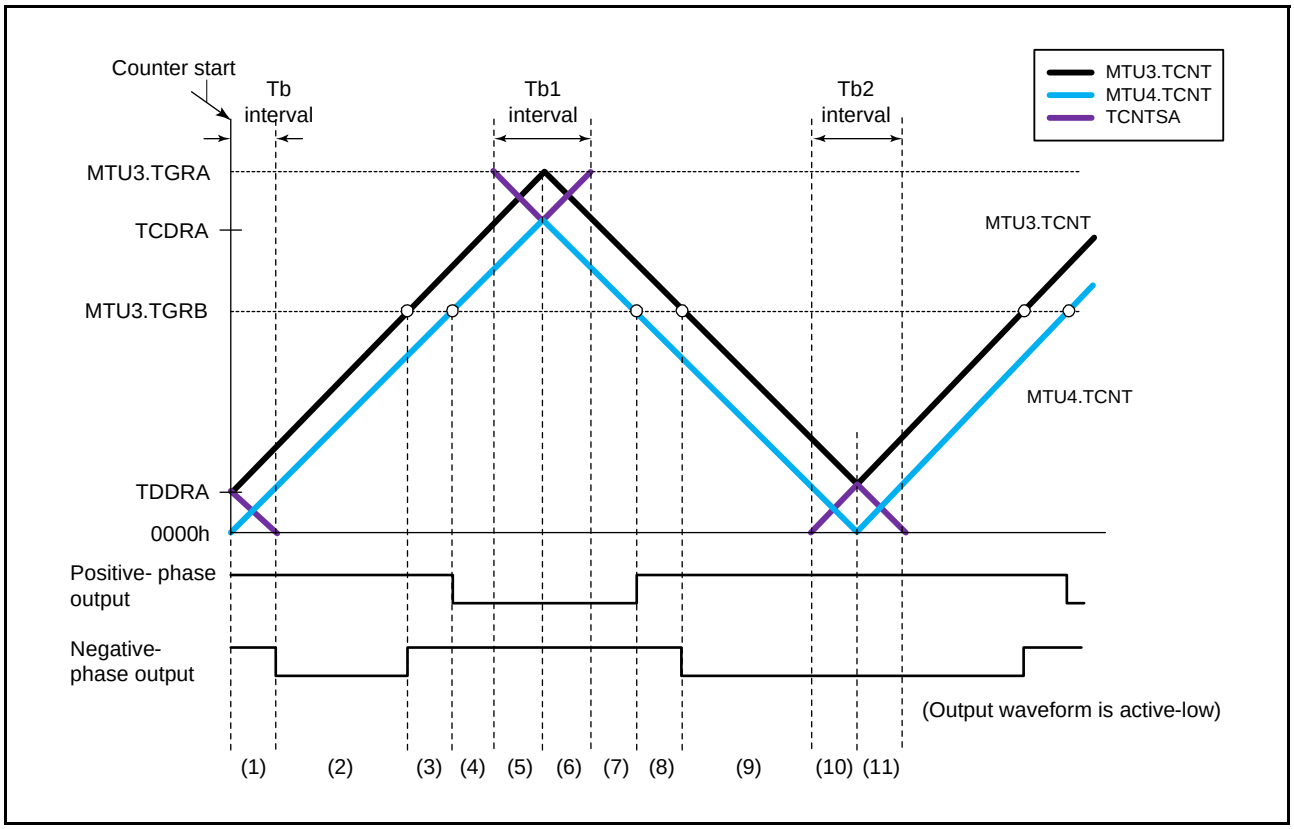


Figure 24.66 Timing for Synchronous Counter Clearing (MTU3 and MTU4)

- Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode.

An example of the procedure for setting output waveform control at synchronous counter clearing in complementary PWM mode is shown in Figure 24.67.

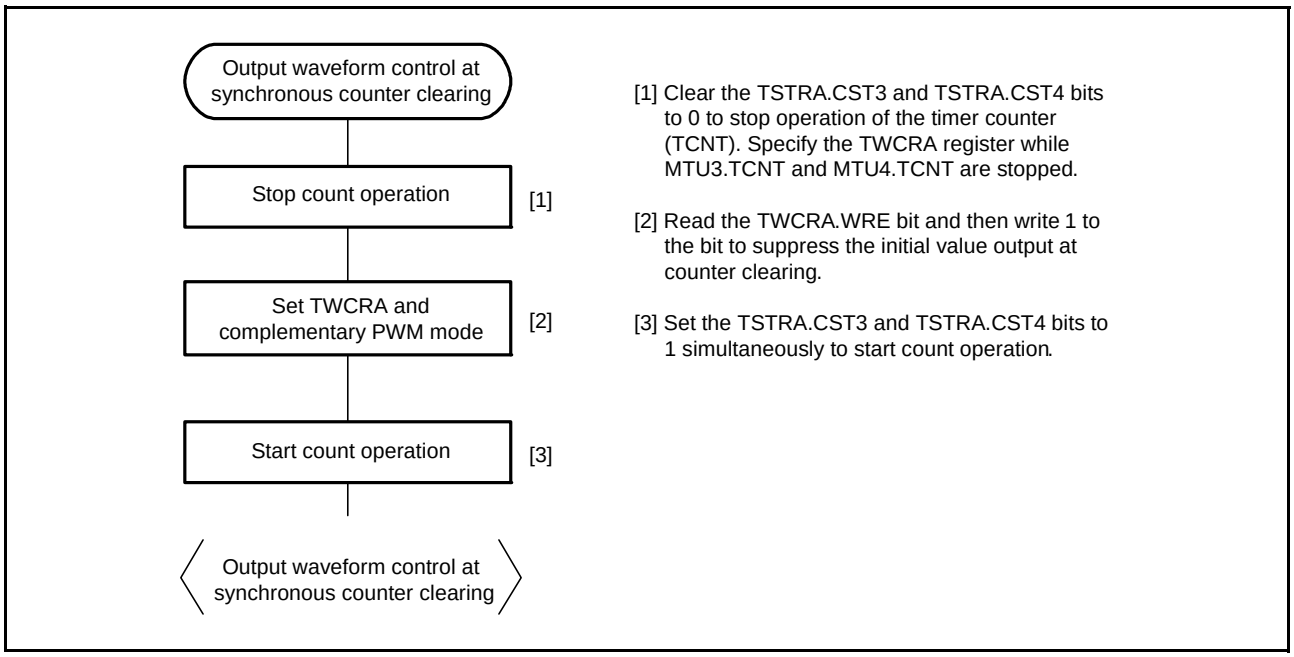


Figure 24.67 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode (MTU3 and MTU4)

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Figure 24.68 to Figure 24.71 show examples of output waveform control in which MTU3 and MTU4 operate in complementary PWM mode and synchronous counter clearing is generated while the WRE bit in TWCRA is set to 1. In the examples shown in Figure 24.68 to Figure 24.71, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 24.66, respectively.

In MTU6 and MTU7, these examples are equivalent to the cases when MTU6 and MTU7 operate in complementary PWM mode and synchronous counter clearing is generated while the SCC bit is cleared to 0 and the WRE bit is set to 1 in TWCRB.

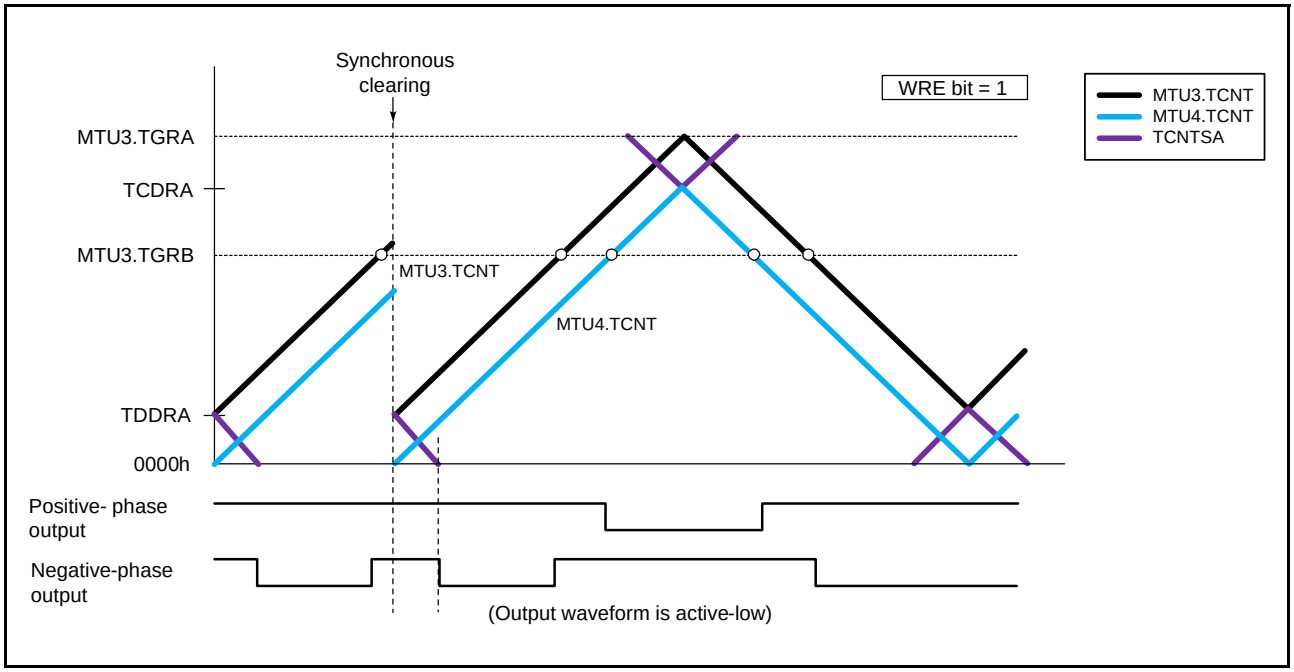


Figure 24.68 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 24.66; TWCRA.WRE Bit is 1)

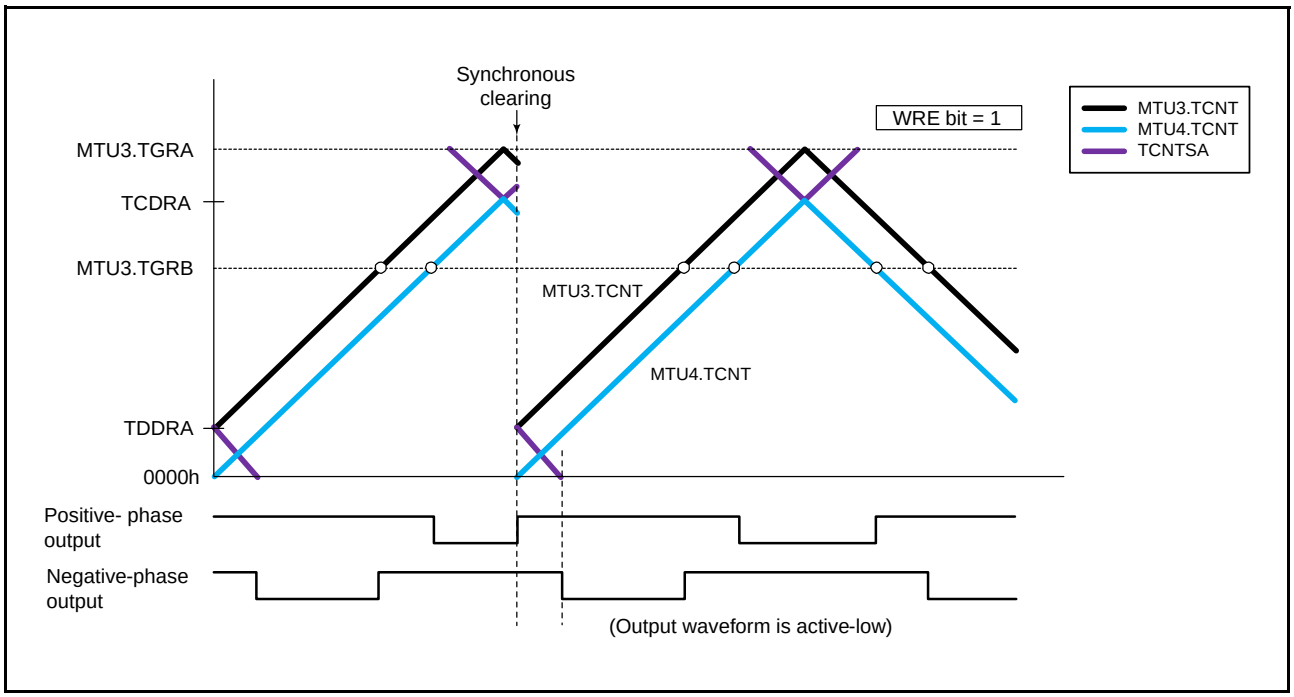


Figure 24.69 Example of Synchronous Clearing in Tb1 interval (Timing (6) in Figure 24.66; TWCRA.WRE Bit is 1)

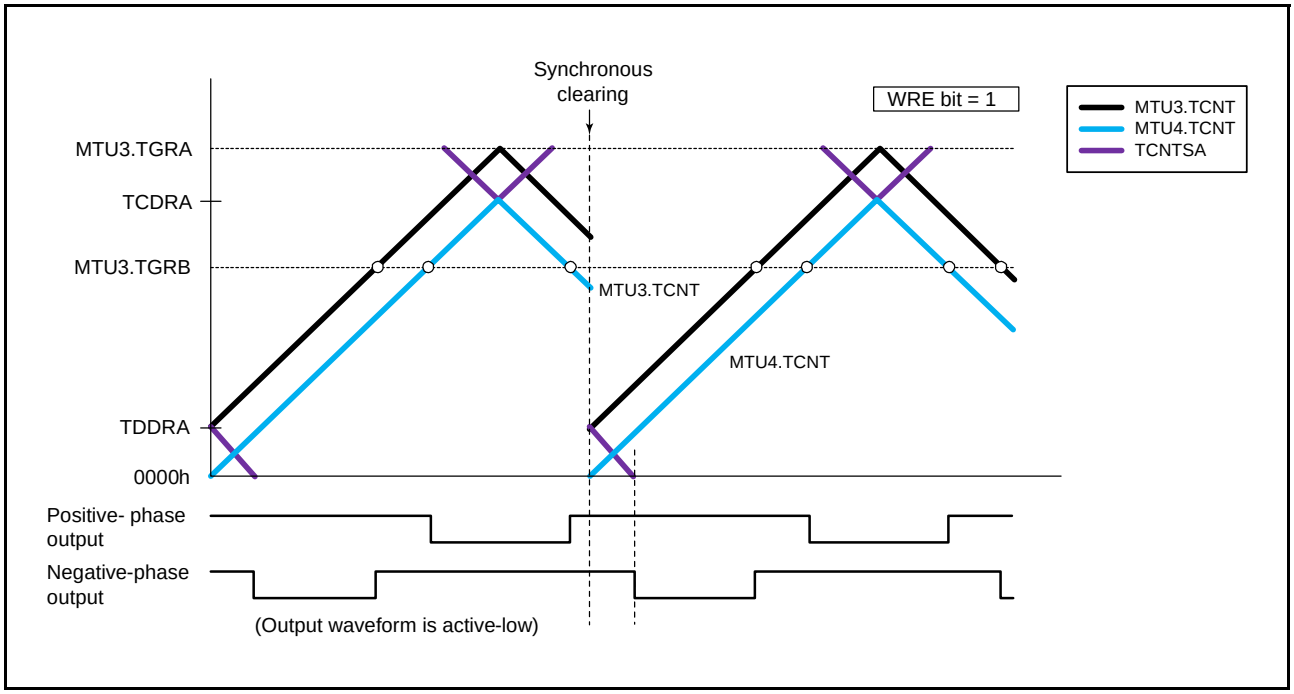


Figure 24.70 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 24.66; TWCRA.WRE Bit is 1)

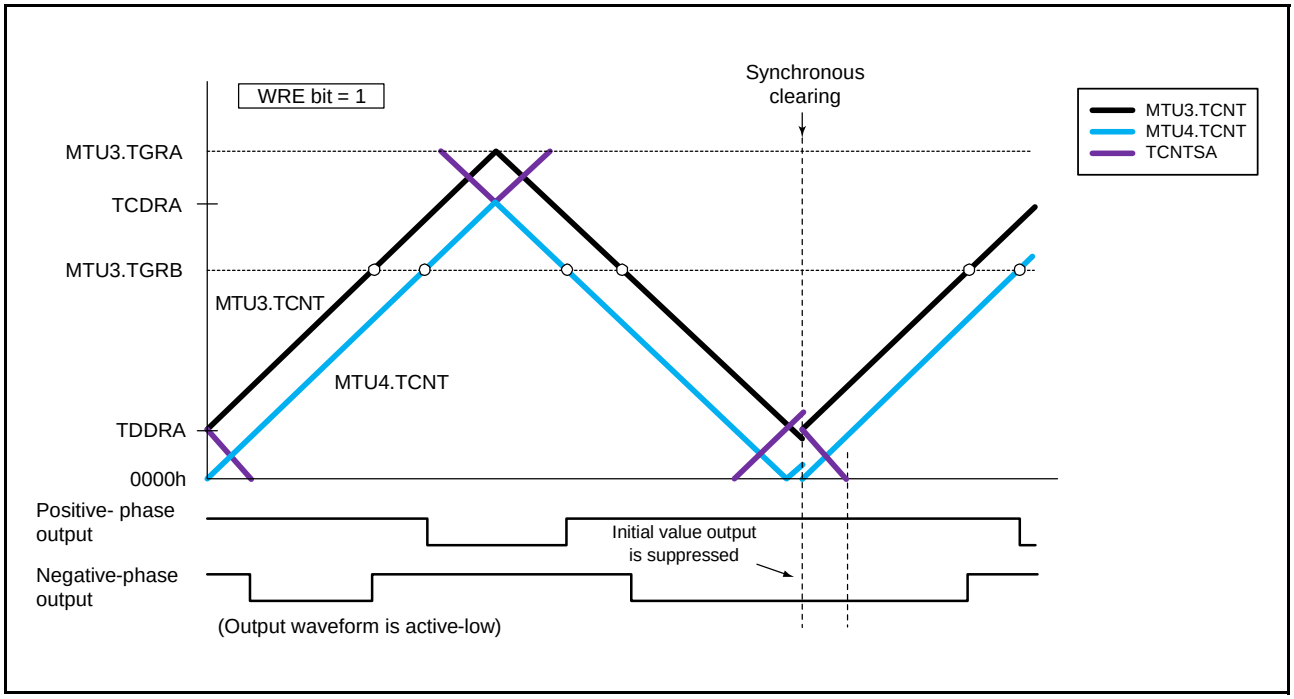


Figure 24.71 Example of Synchronous Clearing in Tb2 interval (Timing (11) in Figure 24.66; TWCRA.WRE Bit is 1)

(o) Suppressing Synchronous Counter Clearing for MTU0 to MTU2, and MTU6 and MTU7

In MTU6 and MTU7, setting the SCC bit in TWCRB to 1 suppresses synchronous counter clearing caused by MTU0 to MTU2.

Synchronous counter clearing is suppressed only within the interval shown in Figure 24.72. When using this function, MTU6 and MTU7 should be set to complementary PWM mode.

For details of synchronous clearing caused by MTU0 to MTU2, refer to section 24.3.10 (2), Synchronous Counter Clearing for MTU6 and MTU7.

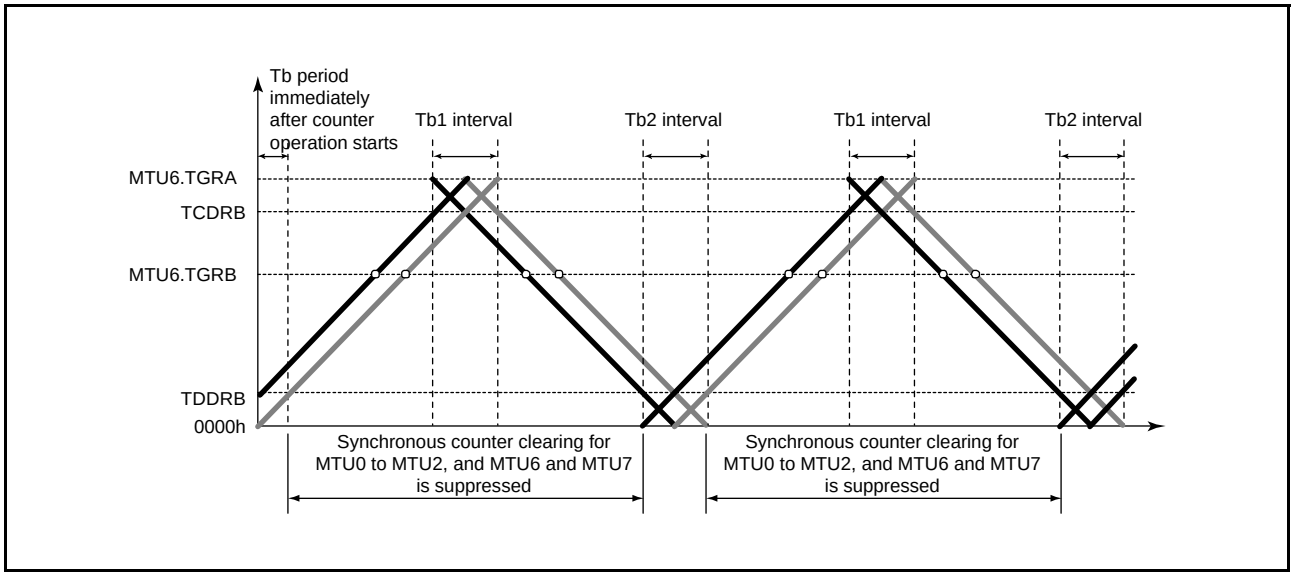


Figure 24.72 Synchronous Clearing-Suppressed Interval Specified by TWCRB.SCC Bit for MTU0 to MTU2, and MTU6 and MTU7

- Example of Procedure for Suppressing Synchronous Counter Clearing for MTU0 to MTU2, and MTU6 and MTU7

An example of the procedure for suppressing synchronous counter clearing for MTU0 to MTU2, and MTU6 and MTU7 is shown in Figure 24.73.

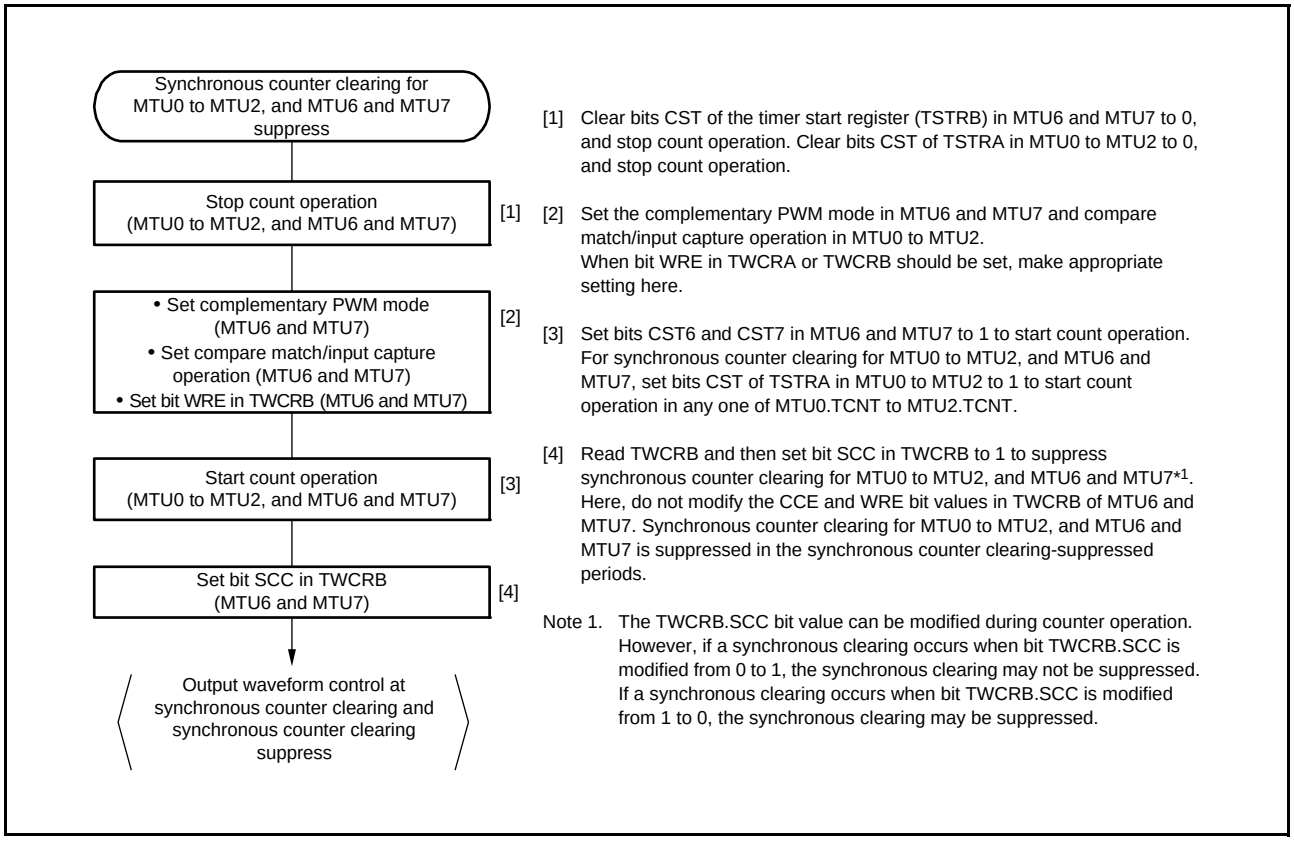


Figure 24.73 Example of Procedure for Suppressing Synchronous Counter Clearing for MTU0 to MTU2, and MTU6 and MTU7

- Examples of Suppression of Synchronous Counter Clearing for MTU 0 to MTU2, and MTU6 and MTU7

Figure 24.74 to Figure 24.77 show examples of operation in which MTU6 and MTU7 operate in complementary PWM mode and synchronous counter clearing for MTU 0 to MTU2, and MTU6 and MTU7 is suppressed by setting the SCC bit in TWCRB in MTU6 and MTU7 to 1. In the examples shown in Figure 24.74 to Figure 24.77, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 24.66, respectively. In these examples, the WRE bit in TWCRB in MTU6 and MTU7 is set to 1.

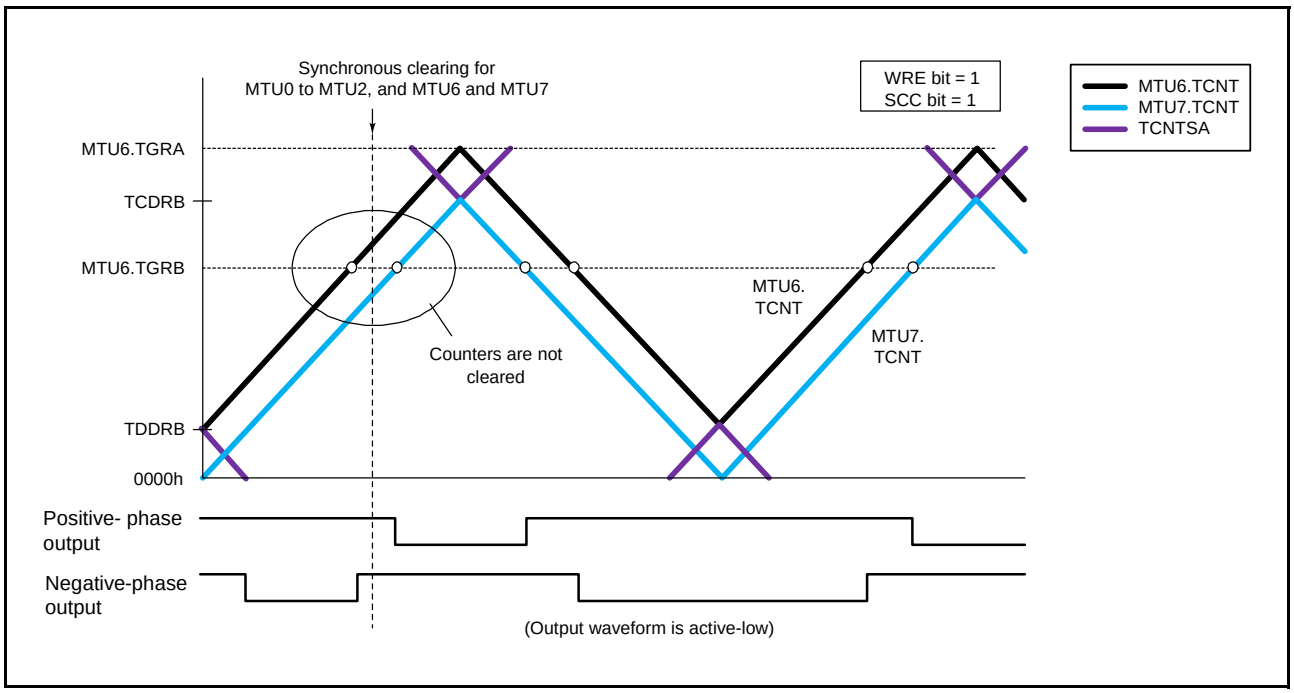


Figure 24.74 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 24.66; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

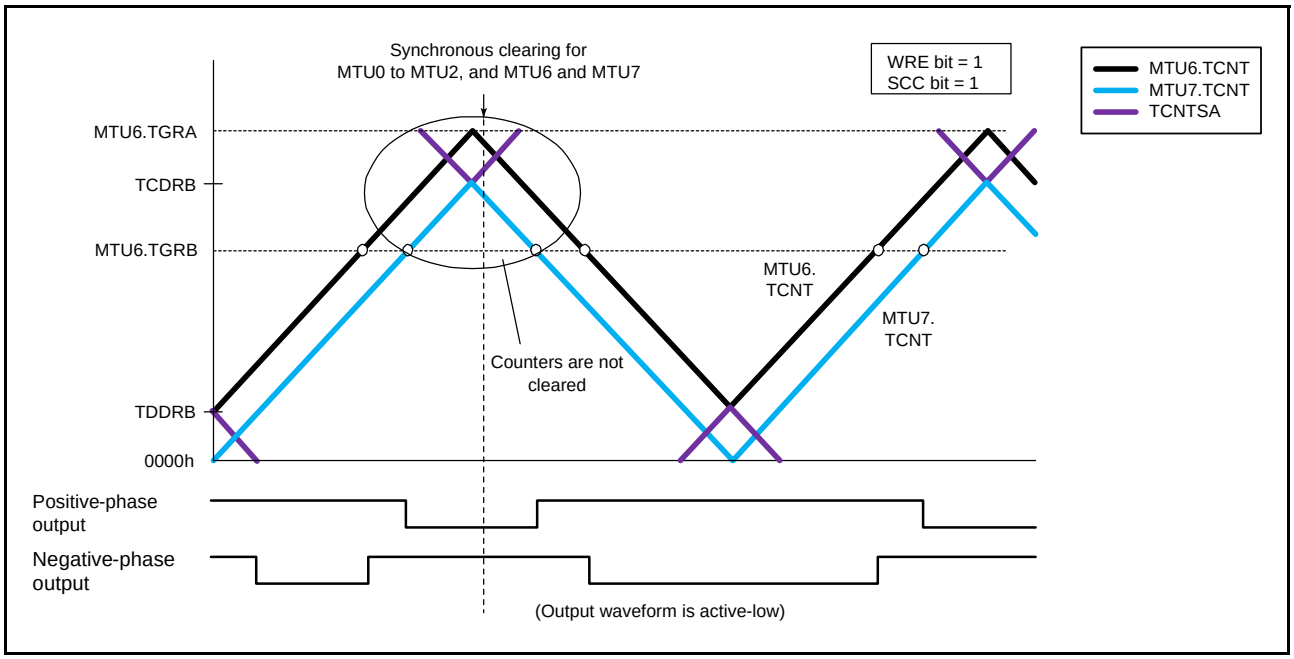


Figure 24.75 Example of Synchronous Clearing in Tb1 interval (Timing (6) in Figure 24.66; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

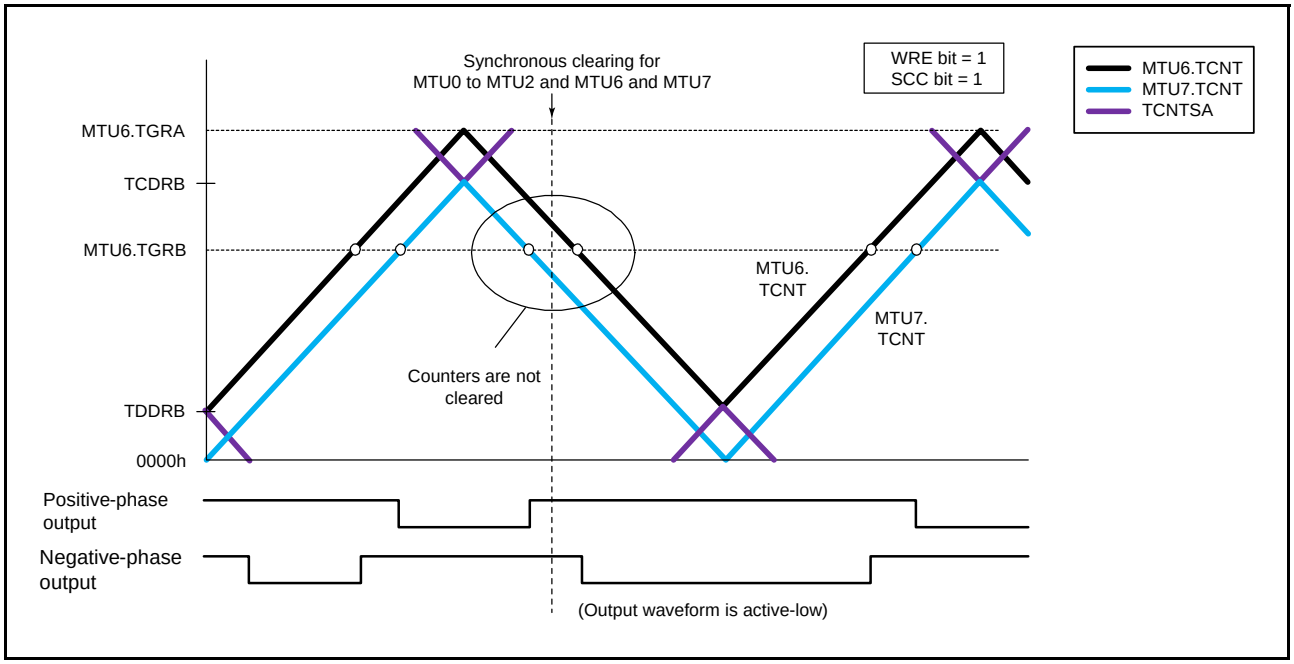


Figure 24.76 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 24.66; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

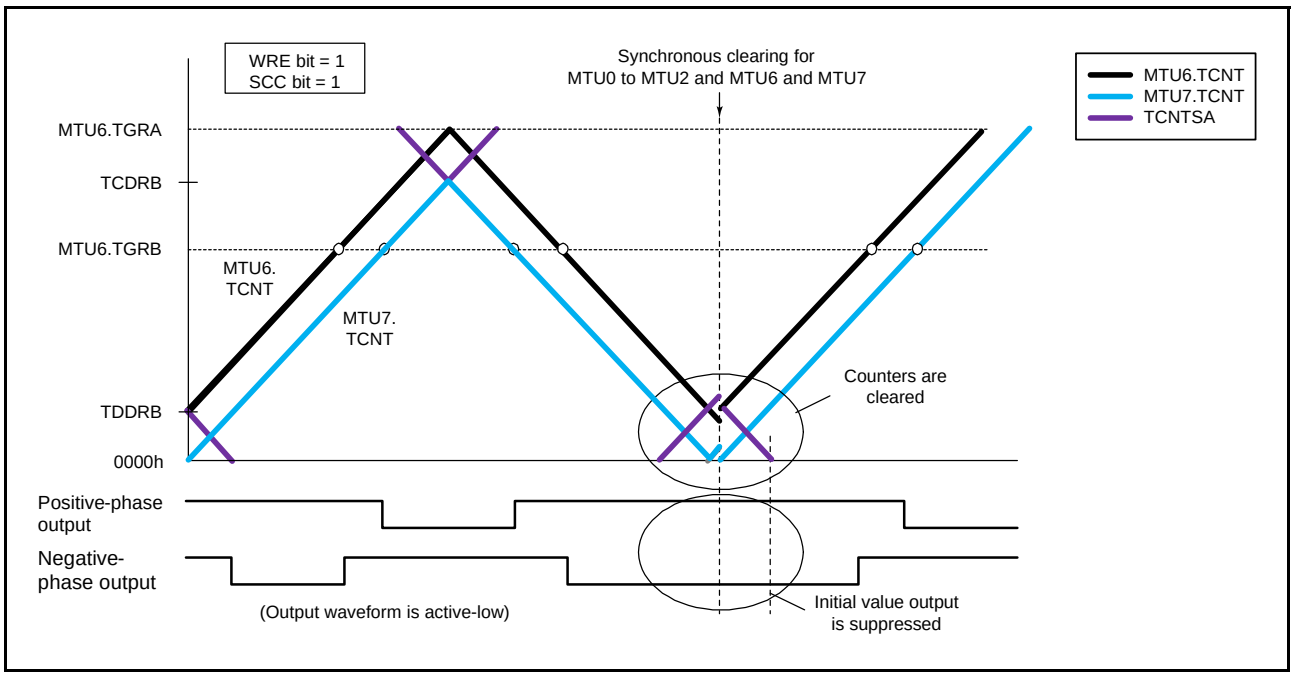


Figure 24.77 Example of Synchronous Clearing in Tb2 interval (Timing (11) in Figure 24.66; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

(p) Counter Clearing by MTU3.TGRA (MTU6.TGRA) Compare Match

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) can be cleared by MTU3.TGRA (MTU6.TGRA) compare match when the TWCRA.CCE (TWCRB.CCE) bit. Figure 24.78 illustrates an operation example.

Note 1. Use this function only in complementary PWM mode 1 (transfer at crest).
Note 2. Do not specify synchronous clearing by another channel (do not set 1 in the SYNC0 to SYNC4, or SYNC6 and SYNC7 bits in the timer synchronous register (TSYRA or TSYRB) and the CE0A to CE0D, CE1A, CE1B, CE2A, or CE2B bits in TSYCR).
Note 3. Do not set the PWM duty value to 0000h.
Note 4. Do not set the PSYE bit in timer output control register 1 (TOCR1A or TOCR1B) to 1.

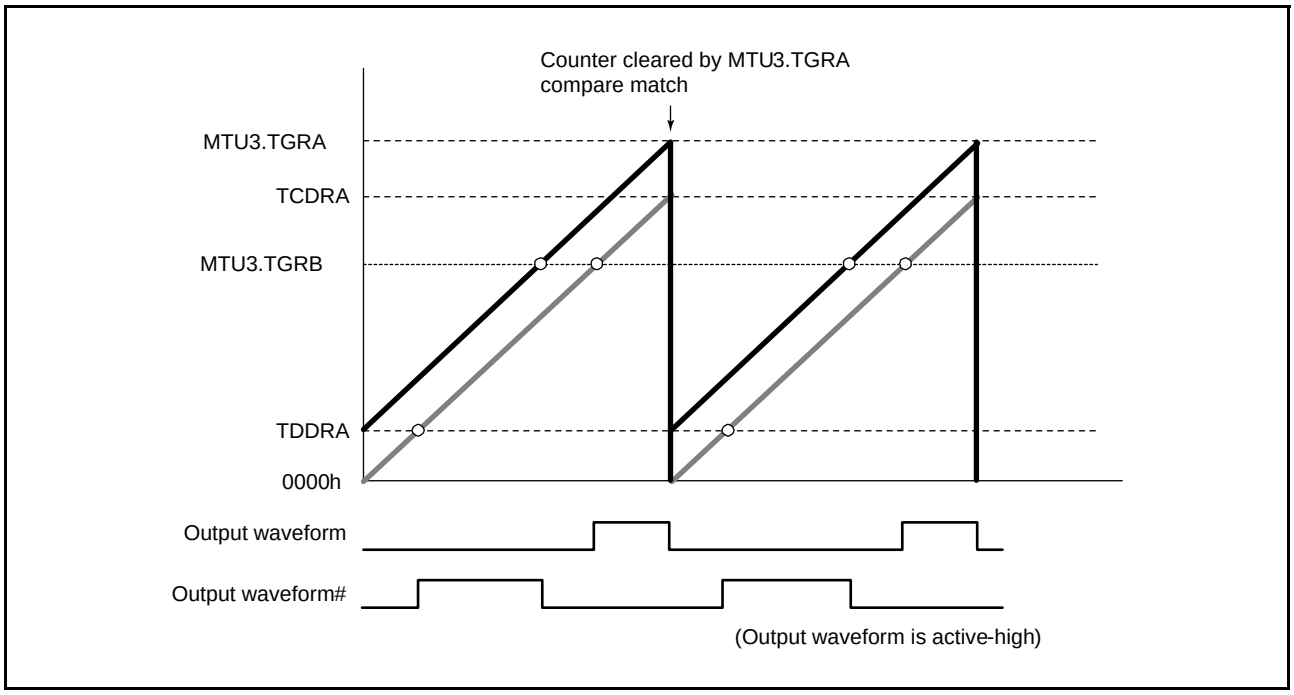


Figure 24.78 Example of Counter Clearing Operation by MTU3.TGRA Compare Match

(q) Example of Waveform Output for Driving AC Synchronous Motor (Brushless DC Motor)

In complementary PWM mode when MTU3 and MTU4 are used, a brushless DC motor can easily be controlled using the TGCRA register. Figure 24.79 to Figure 24.82 show examples of brushless DC motor driving waveforms created using TGCRA.

To switch the output phases for a 3-phase brushless DC motor by means of external signals detected with a Hall element, etc., clear the TGCRA.FB bit to 0. In this case, the external signals indicating the magnetic pole position should be input to timer input pins MTIOC0A, MTIOC0B, and MTIOC0C in MTU0 (make appropriate settings with the MPC and port mode registers (PMR) of the I/O ports). When an edge is detected at pin MTIOC0A, MTIOC0B, or MTIOC0C, the output on/off state is switched automatically.

When the TGCRA.FB bit is 1, the output on/off state is switched when the UF, VF, or WF bit in TGCRA is cleared to 0 or set to 1.

The driving waveforms are output from the 6-phase PWM output pins for complementary PWM mode.

With this 6-phase output, while the output is turned on, chopping output is available through complementary PWM mode output function by setting the N bit or P bit in TGCRA to 1. When the N bit or P bit is 0, the level output is selected.

The active level of the 6-phase output (on output level) can be set with the TOCR1A.OLSN and TOCR1A.OLSP bits regardless of the setting of the N and P bits.

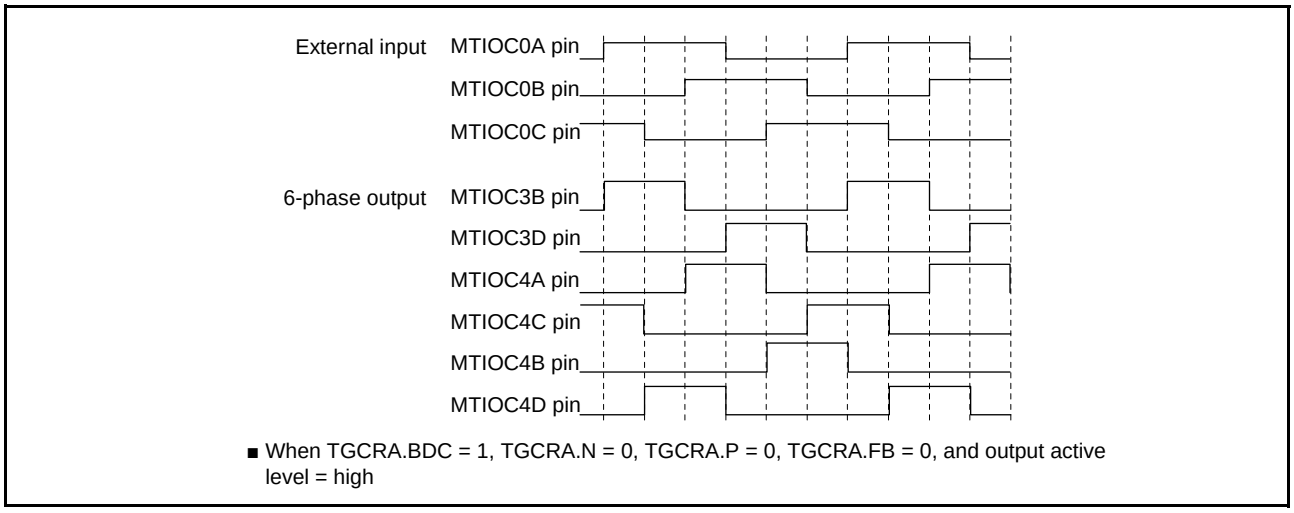


Figure 24.79 Example of Output Phase Switching by External Input (1)

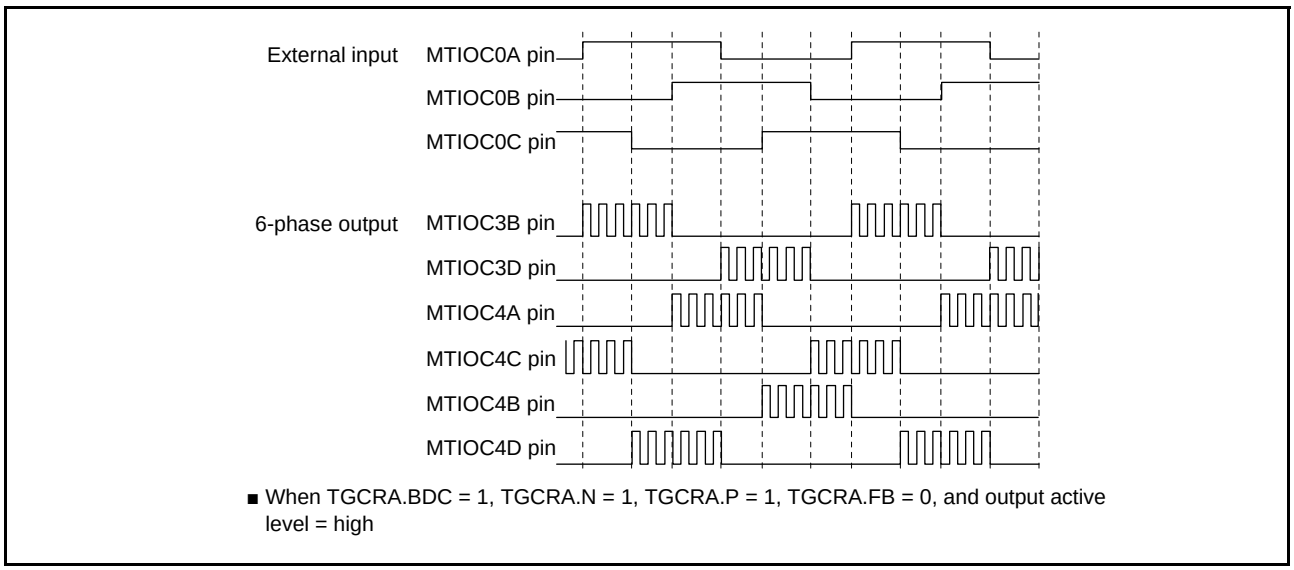
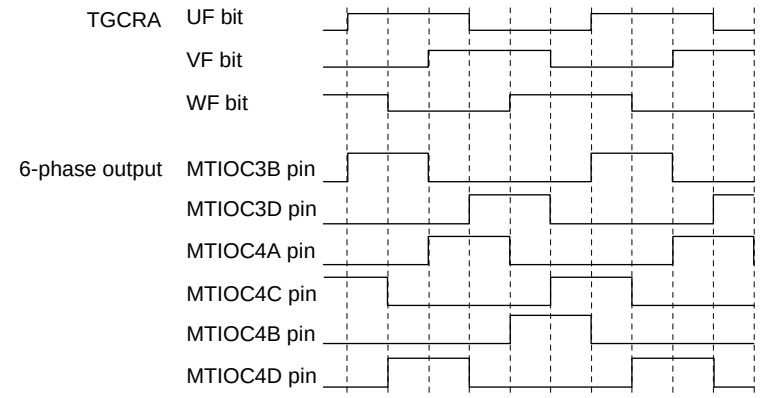
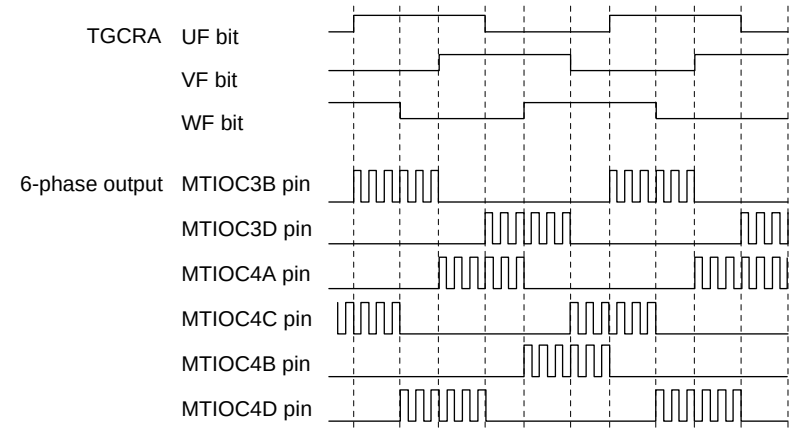


Figure 24.80 Example of Output Phase Switching by External Input (2)



■ When TGCRA.BDC = 1, TGCRA.N = 0, TGCRA.P = 0, TGCRA.FB = 1, and output active level = high

Figure 24.81 Example of Output Phase Switching through UF, VF, and WF Bit Settings (1)



■ When TGCRA.BDC = 1, TGCRA.N = 1, TGCRA.P = 1, TGCRA.FB = 1, and output active level = high

Figure 24.82 Example of Output Phase Switching through UF, VF, and WF Bit Settings (2)

(r) A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter start request can be issued using MTU3.TGRA (MTU6.TGRA) compare match, MTU4.TCNT (MTU7.TCNT) underflow (trough), or compare match on a channel other than MTU3 and MTU4 (MTU6 and MTU7).
When start requests using MTU3.TGRA (MTU6.TGRA) compare match are specified, A/D conversion can be started at the crest of the MTU3.TCNT (MTU6.TCNT) count.
A/D converter start requests can be specified by setting the TIER.TTGE bit. To issue an A/D converter start request at an MTU4.TCNT (MTU7.TCNT) underflow (trough), set the MTU4.TIER.TTGE2 (MTU7.TIER.TTGE2) bit to 1.

(s) Double Buffer Function in Complementary PWM Mode

In complementary PWM mode 3 (transfer at the crest and trough), the PWM output setting resolution can be improved from ±2 to ±1 by setting the TMDR2A.DRS (TMDR2B.DRS) bit to 1.
When setting buffer registers A (MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD), set also buffer registers B (MTU3.TGRE, MTU4.TGRE, MTU4.TGRF, MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF) at the same time. Each buffer register B should be set to the buffer register A value or (buffer register A value – 1). For details of the setting procedure, refer to section 24.3.8 (1), Example of Complementary PWM Mode Setting Procedure

Note: When a buffer register B is set to the buffer register A value, symmetric PWM waveforms are output. When a buffer register B is set to (buffer register A value – 1), asymmetric PWM waveforms are output.

Figure 24.83 shows an example of double buffer operation.
Each register data is transferred as follows.

- After MTU4.TGRD or MTU7.TGRD (buffer A) is written to, data is transferred from MTU4.TGRD or MTU7.TGRD (buffer A) to Temp3A or Temp6A (temporary A) and from MTU4.TGRF or MTU7.TGRF (buffer B) to Temp3B or Temp6B (temporary B).
- With timing (1) in the figure, data is transferred from Temp3A or Temp6A (temporary A) to MTU4.TGRB or MTU7.TGRB (compare).
- With timing (2) in the figure, data is transferred from Temp3B or Temp6B (temporary B) to MTU4.TGRB or MTU7.TGRB (compare).

In the crest interval (Tb1 interval), the compare register and temporary register A are valid; in the trough interval (Tb2 interval), the compare register and temporary register B are valid.

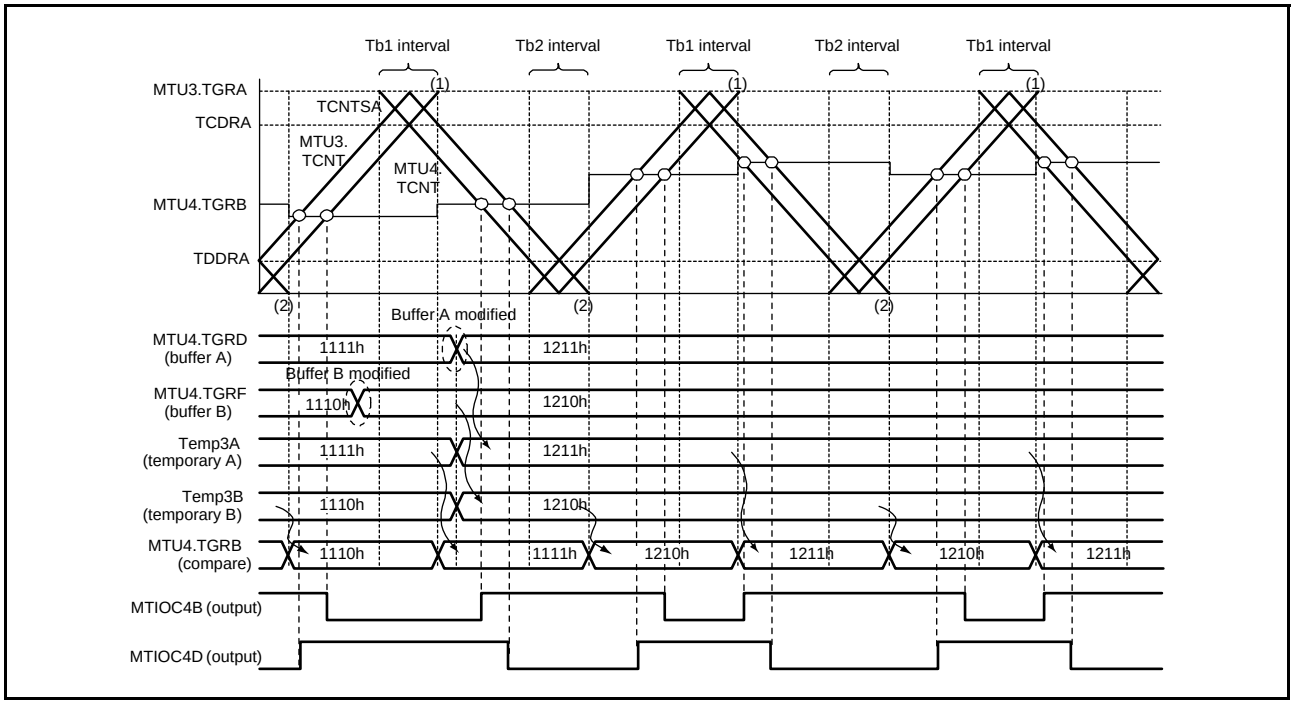


Figure 24.83 Example of Double Buffer Operation

Figure 24.84 shows an example when the buffer write value is smaller than the TDDRA (TDDRB) value, and Figure 24.85 shows an example when the write value is greater than TCDRA (TCDRB).
In the crest interval, the output is controlled according to the compare match with the compare register or temporary register A; in the trough interval, the output is controlled according to the compare match with the compare register or temporary register B.

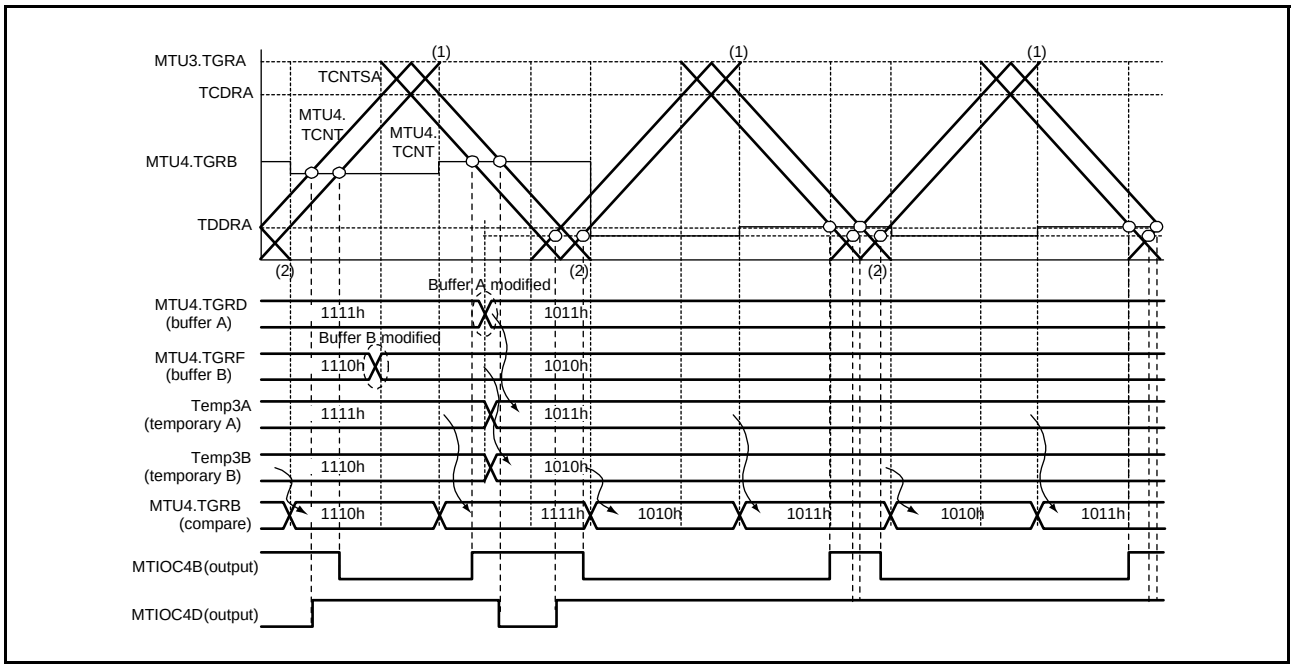


Figure 24.84 Example of Double Buffer Operation (Buffer Write Value is Smaller than TDDRA)

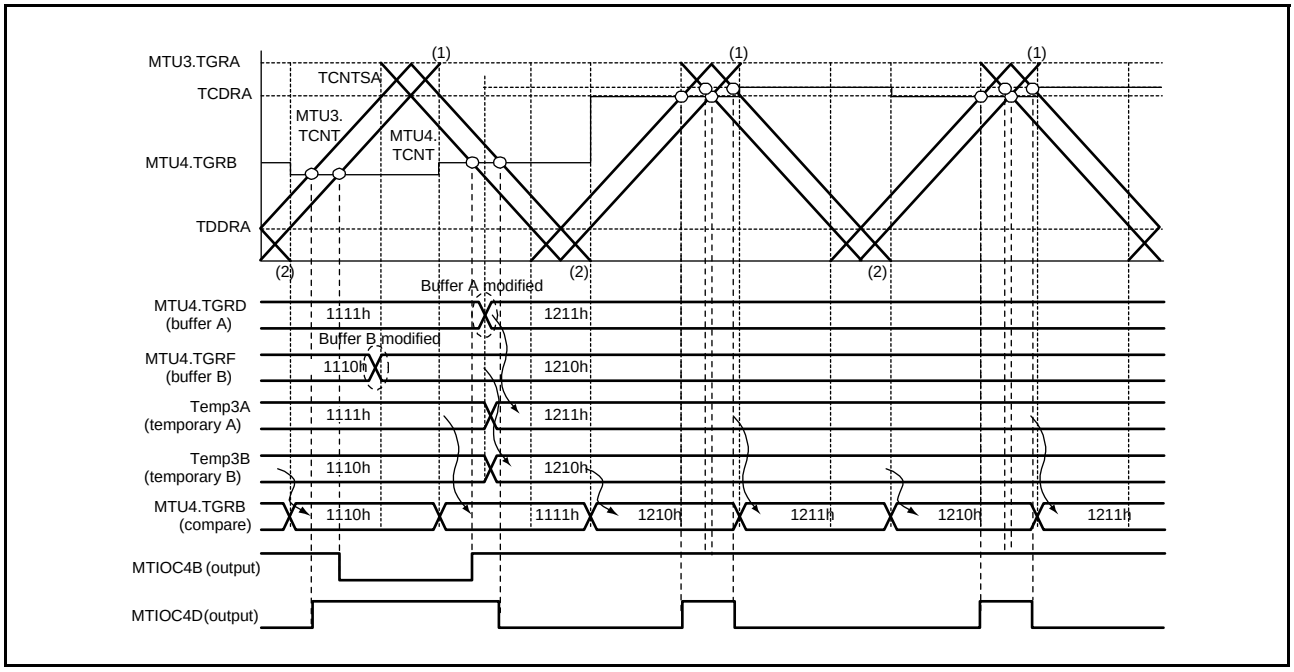


Figure 24.85 Example of Double Buffer Operation (Buffer Write Value is Greater than TCDRA)

(3) Interrupt Skipping Function 1 in Complementary PWM Mode

Interrupts TGIA3 (TGIA6) (at the crest) and TCIV4 (TCIV7) (at the trough) in MTU3 and MTU4 (MTU6 and MTU7) can be skipped up to seven times by making settings in the TITCR1A (TITCR1B) register.
Transfers from a buffer register to a temporary register or a compare register can be skipped in coordination with interrupt skipping by making settings in the TBTERA (TBTERB) register. For the linkage with buffer registers, refer to description (c), Buffer Transfer Control Linked with Interrupt Skipping, below.
A/D converter start requests generated by the A/D converter start request delaying function can also be skipped in coordination with interrupt skipping by making settings in the MTU4.TADCR (MTU7.TADCR) register. For the linkage with the A/D converter start request delaying function, refer to section 24.3.9, A/D Converter Start Request Delaying Function.
The TITCR1A (TITCR1B) register should be set while interrupt skipping function 1 is selected by setting the TITM bit in the timer interrupt skipping mode register (TITMRA or TITMRB) to 0, TGIA3 (TGIA6) interrupt requests are disabled by setting the MTU3.TIER (MTU6.TIER) register, TCIV4 (TCIV7) interrupt requests are disabled by setting the MTU4.TIER (MTU7.TIER) register, and a compare match is not generated. Before changing the skipping count, be sure to clear the T3AEN (T6AEN) and T4VEN (T7VEN) bits to 0 to clear the skipping counter.

(a) Example of Interrupt Skipping Function 1 Setting Procedure

Figure 24.86 shows an example of the interrupt skipping function 1 setting procedure. Figure 24.87 shows the periods during which interrupt skipping count can be changed.

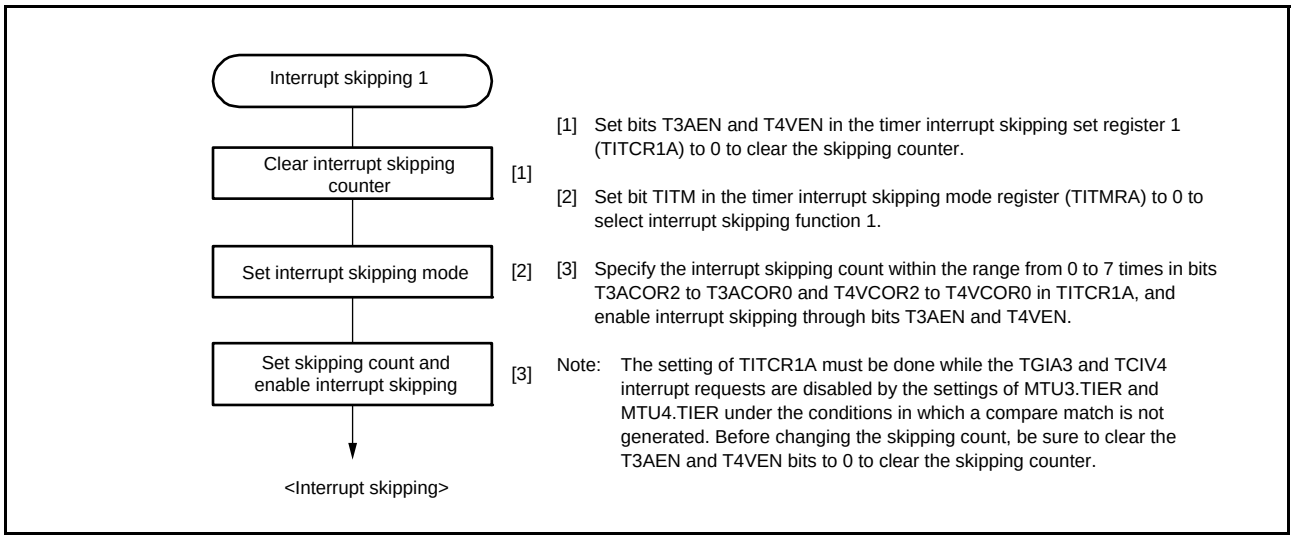


Figure 24.86 Example of Interrupt Skipping Function 1 Setting Procedure

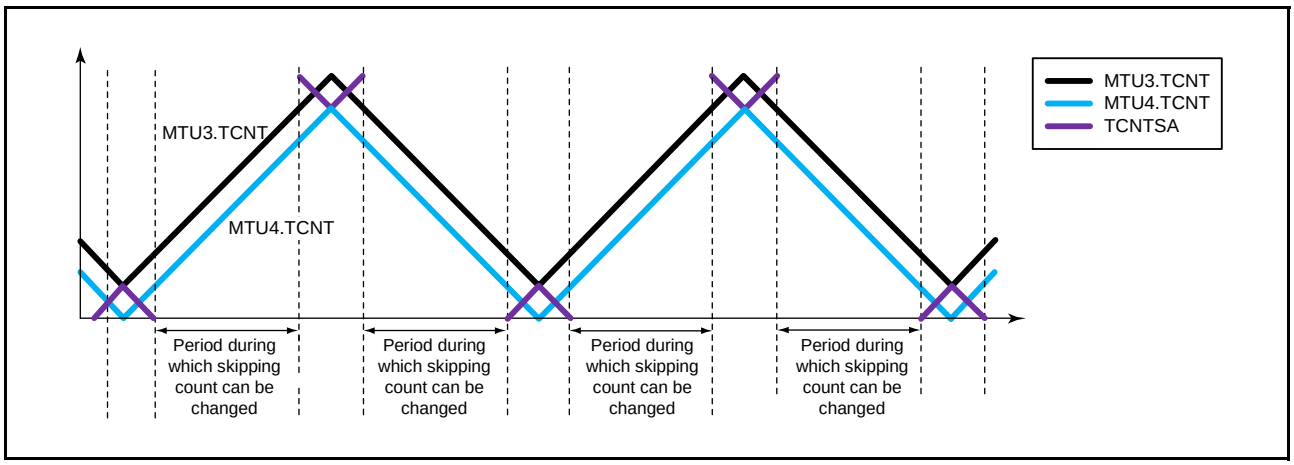


Figure 24.87 Periods during which Interrupt Skipping Count can be Changed

(b) Example of Interrupt Skipping Function 1

Figure 24.88 shows an example of TGIA3 (TGIA6) interrupt skipping in which the interrupt skipping count is set to three by the T3ACOR (T6ACOR) bits and the T3AEN (T6AEN) bit is set to 1 in the TITCR1A (TITCR1B) register.

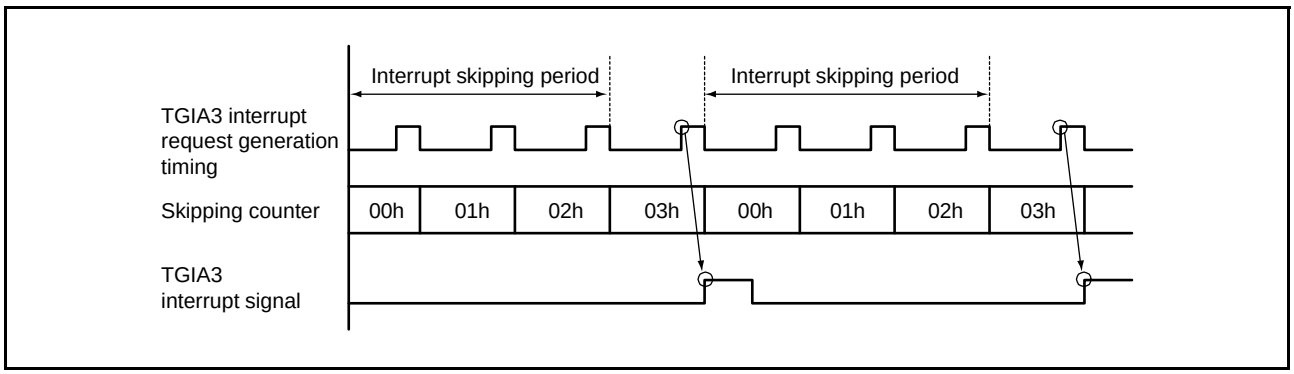
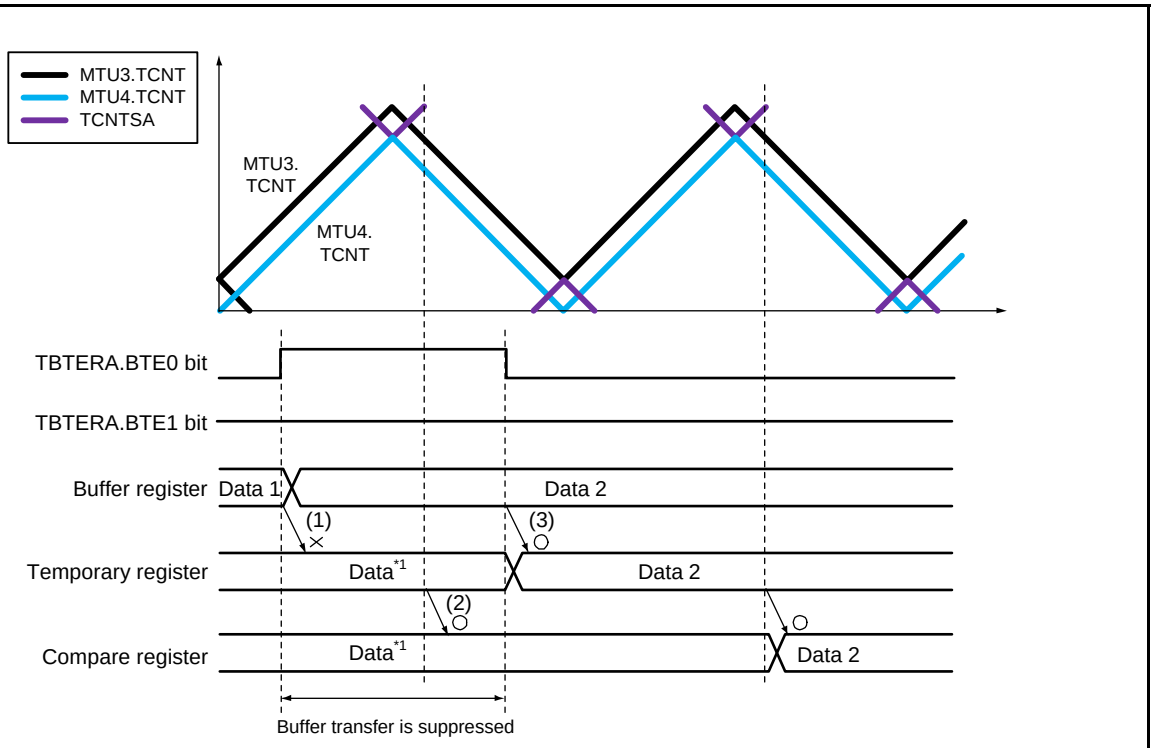


Figure 24.88 Example of Interrupt Skipping Function 1

(c) Buffer Transfer Control Linked with Interrupt Skipping

In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE[1:0] bits in the TBTERA (TBTERB) register.
Figure 24.89 shows an example of operation when buffer transfer is disabled (BTE[1:0] = 01b). While this setting is valid, data is not transferred from the buffer register to the temporary register.
Figure 24.90 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE[1:0] = 10b). While this setting is valid, data is not transferred from the buffer register outside the buffer transfer-enabled period. Note that the buffer transfer-enabled period differs depending on whether only the T3AEN (T6AEN) bit in the TITCR1A (TITCR1B) register is set to 1, only the T4VEN (T7VEN) bit in the TITCR1A (TITCR1B) register is set to 1, or both the T3AEN and T4VEN (T6AEN and T7VEN) bits are set to 1. Figure 24.91 shows the relationship between the T3AEN (T6AEN) and T4VEN (T7VEN) bit settings in TITCR1A (TITCR1B) and buffer transfer-enabled period.

Note: This function must be used in combination with interrupt skipping function 1.
When interrupt skipping is disabled (the T3AEN and T4VEN (T6AEN and T7VEN) bits in the timer interrupt skipping set register 1 (TITCR1A or TITCR1B) are cleared to 0 or the skipping count set bits (T3ACOR and T4VCOR (T6ACOR and T7VCOR)) in TITCR1A (TITCR1B) are cleared to 0), make sure that buffer transfer is not linked with interrupt skipping (clear the BTE1 bit in TBTERA or TBTERB to 0).
If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.



(1) No data is transferred from the buffer register to the temporary register in the buffer transfer-disabled period (bits BTE[1:0] in TBTERA are set to 01b).
(2) Data is transferred from the temporary register to the compare register even in the buffer transfer-disabled period.
(3) After buffer transfer is enabled, data is transferred from the buffer register to the temporary register.

Note 1. When buffer transfer at the crest is selected.

Figure 24.89 Example of Operation When Buffer Transfer is Disabled (BTE[1:0] = 01b)

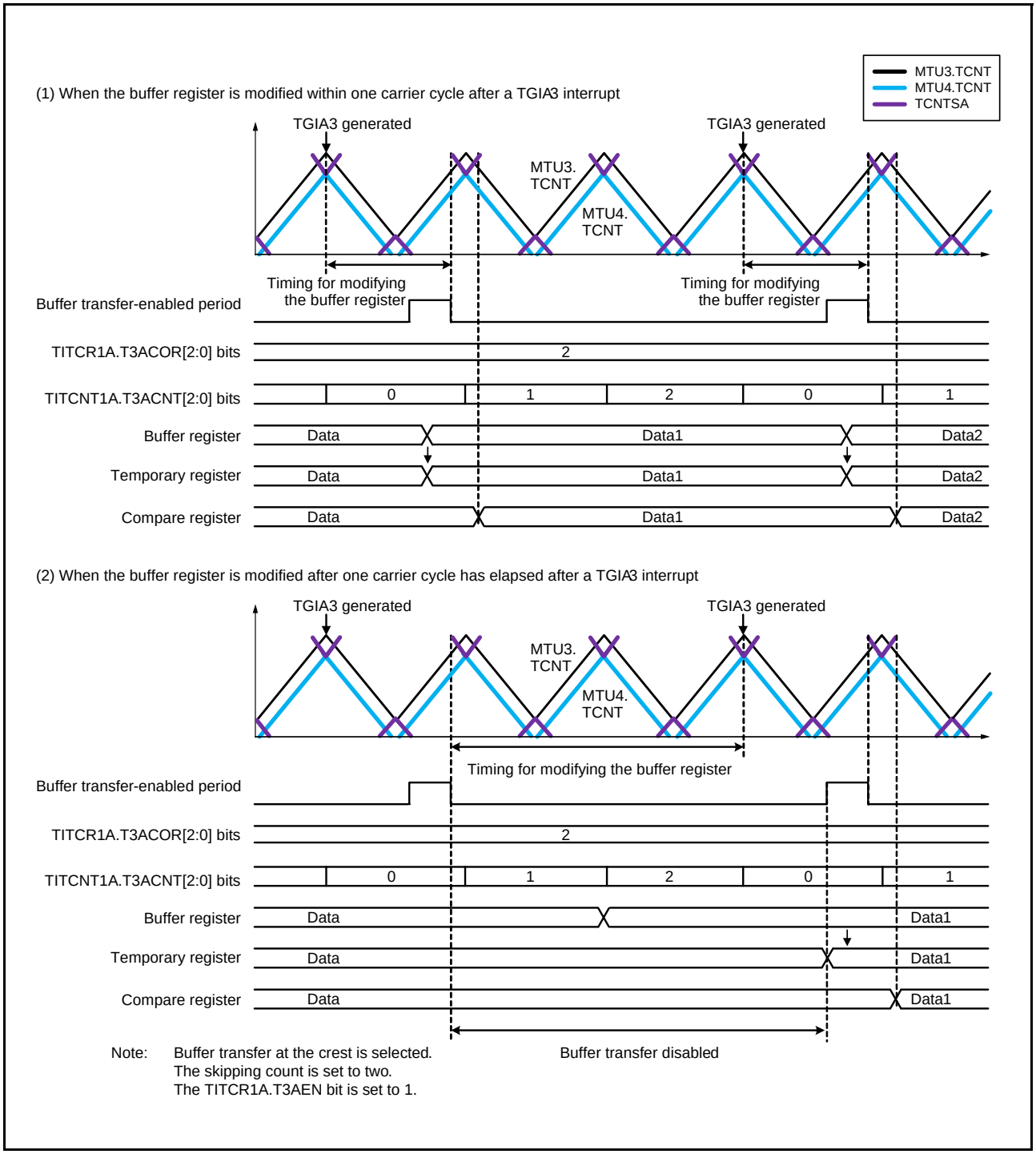


Figure 24.90 Example of Operation When Buffer Transfer is Linked with Interrupt Skipping (BTE[1:0] = 10b)

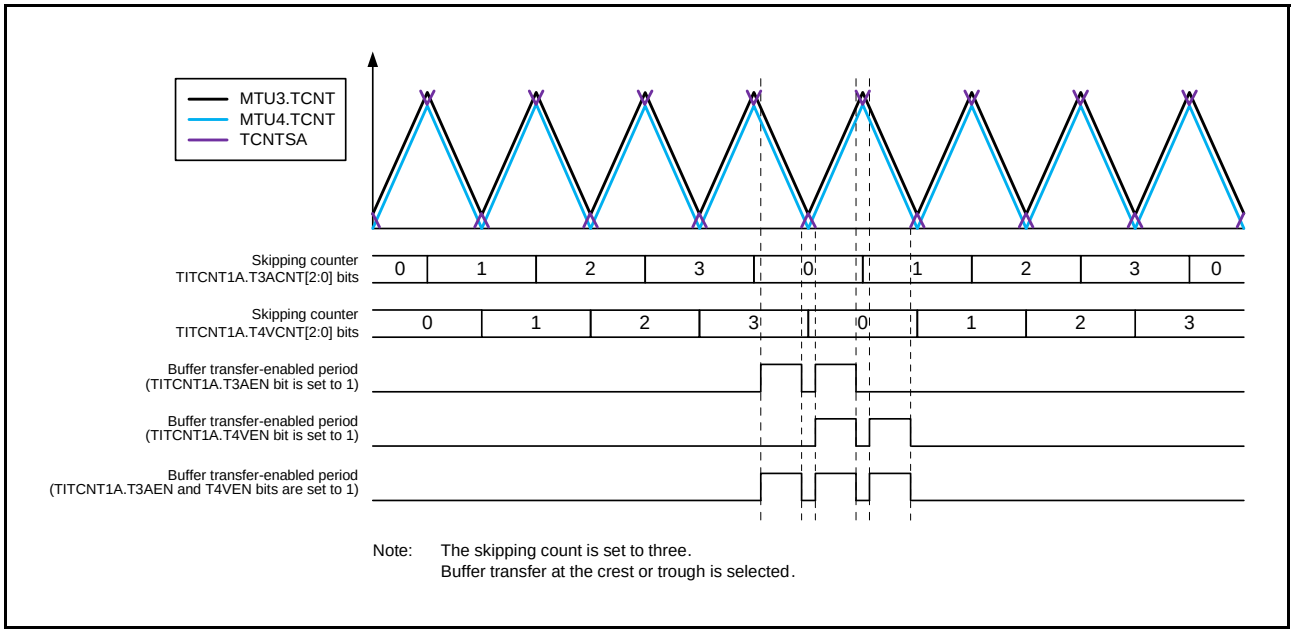


Figure 24.91 Relationship between Bits T3AEN and T4VEN in TITCR1A and Buffer Transfer-Enabled Period

(4) Complementary PWM Mode Output Protection Functions

The following output protection functions are provided for complementary PWM mode.

(a) Register and Counter Miswrite Prevention Function

Access from the CPU to the mode registers, control registers, compare registers, and counters can be enabled or disabled by setting the RWE bit in the TRWERA (TRWERB) register. The applicable registers are some of the registers in MTU3, MTU4, MTU6, and MTU7 shown below:
47 registers in total
MTU3.TCR, MTU4.TCR, MTU3.TCR2, MTU4.TCR2, MTU3.TMDR1, MTU4.TMDR1, MTU3.TIORH, MTU4.TIORH, MTU3.TIORL, MTU4.TIORL, MTU3.TIER, MTU4.TIER, MTU3.TCNT, MTU4.TCNT, MTU3.TGRA, MTU4.TGRA, MTU3.TGRB, MTU4.TGRB, MTU.TOERA, MTU.TOCR1A, MTU.TOCR2A, MTU.TGCRA, MTU.TCDRA, MTU.TDDRA,
MTU6.TCR, MTU7.TCR, MTU6.TCR2, MTU7.TCR2, MTU6.TMDR1, MTU7.TMDR1, MTU6.TIORH, MTU7.TIORH, MTU6.TIORL, MTU7.TIORL, MTU6.TIER, MTU7.TIER, MTU6.TCNT, MTU7.TCNT, MTU6.TGRA, MTU7.TGRA, MTU6.TGRB, MTU7.TGRB, MTU.TOERB, MTU.TOCR1B, MTU.TOCR2B, MTU.TCDRB, and MTU.TDDRB
This function can disable CPU access to the mode registers, control registers, and counters to prevent miswriting due to CPU runaway. In the access-disabled state, the applicable registers are read as undefined and writing to these registers is ignored.

(b) Halting of PWM Output by External Signal

The PWM output pins of MTU0, MTU3, MTU4, MTU6, and MTU7 can be set to the high-impedance state automatically.
Refer to section 25, Port Output Enable 3 (POE3a), for details.

24.3.9 A/D Converter Start Request Delaying Function

A/D converter start requests can be issued in MTU4 or MTU7 by making settings in the timer A/D converter start request control register (MTU4.TADCR or MTU7.TADCR), timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB, or MTU7.TADCORA and MTU7.TADCORB), and timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB, or MTU7.TADCOBRA and MTU7.TADCOBRB).

The A/D converter start request delaying function compares MTU4.TCNT with MTU4.TADCORA or MTU4.TADCORB (MTU7.TCNT with MTU7.TADCORA or MTU7.TADCORB), and when their values match, the function issues a respective A/D converter start request (TRG4AN or TRG4BN (TRG7AN or TRG7BN)).

A/D converter start requests (TRG4AN and TRG4BN (TRG7AN and TRG7BN)) can be skipped in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in MTU4.TADCR (the ITA6AE, ITA7VE, ITB6AE, and ITB7VE bits in MTU7.TADCR).

(1) Example of Procedure for Specifying A/D Converter Start Request Delaying Function

Figure 24.92 shows an example of procedure for specifying the A/D converter start request delaying function.

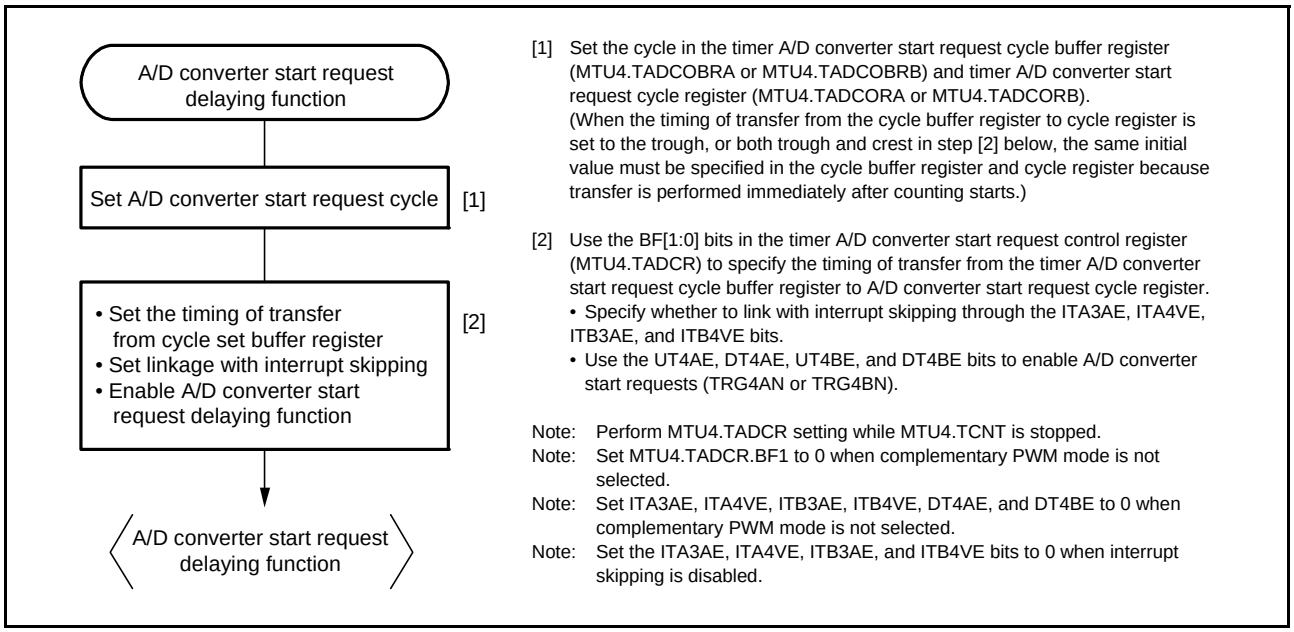
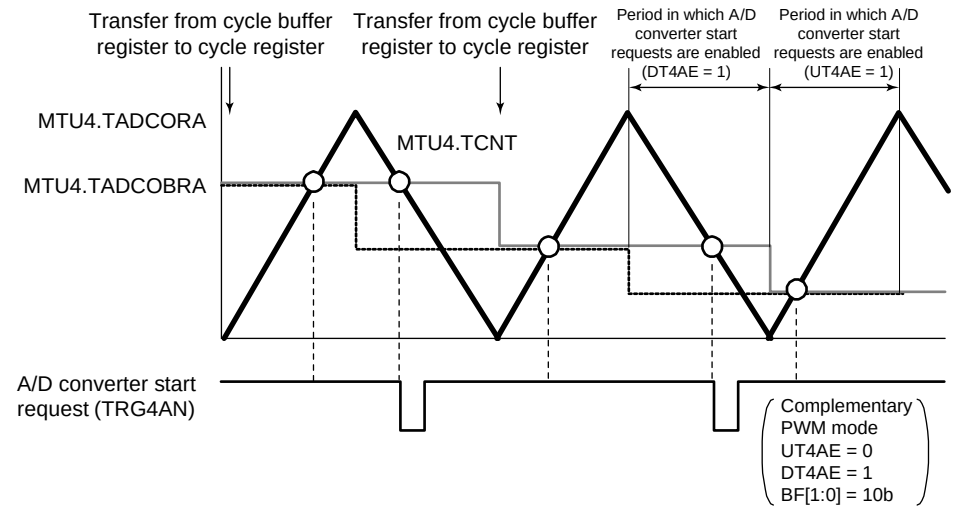


Figure 24.92 Example of Procedure for Specifying A/D Converter Start Request Delaying Function (MTU3 and MTU4)

(2) Basic Example of A/D Converter Start Request Delaying Function Operation

Figure 24.93 shows a basic example of A/D converter start request signal (TRG4AN (TRG7AN)) operation when the trough of MTU4.TCNT (MTU7.TCNT) is specified for the buffer transfer timing and an A/D converter start request signal is output during MTU4.TCNT (MTU7.TCNT) down-counting.



Period in which A/D converter start requests are enabled (UT4AE = 1): 0 ≤ MTU4.TCNT ≤ TCDR - 1
Period in which A/D converter start requests are enabled (DT4AE = 1): TCDR ≥ MTU4.TCNT ≥ 1

UT4AE, DT4AE, BF[1:0]: Bits in TADCR

Figure 24.93 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation

(3) Period in Which A/D Converter Start Requests are Enabled

When the MTU4.TCNT (MTU7.TCNT) counter and the MTU4.TADCORA or MTU4.TADCORB (MTU7.TADCORA or MTU7.TADCORB) register matches within the period enabled by the UT4AE and UT4BE (UT7AE and UT7BE) bits, the corresponding A/D converter start request (TRG4AN or TRG4BN) is issued.

When the UT4AE and UT4BE (UT7AE and UT7BE) bits in the MTU4.TADCR (MTU7.TADCR) register are set to 1 in complementary PWM mode, A/D converter start requests are enabled during the MTU4.TCNT (MTU7.TCNT) up-counting (0 ≤ MTU4.TCNT (MTU7.TCNT) ≤ TCDR – 1). When the DT4AE and DT4BE (DT7AE and DT7BE) bits in the MTU4.TADCR (MTU7.TADCR) register are set to 1, A/D converter start requests are enabled during MTU4.TCNT (MTU7.TCNT) down-counting (TCDR ≥ MTU4.TCNT (MTU7.TCNT) ≥ 1). Refer to Figure 24.93.

(4) Buffer Transfer

The data in the timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB, or MTU7.TADCORA and MTU7.TADCORB) is updated by writing data to the timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB, or MTU7.TADCOBRA and MTU7.TADCOBRB). Data is transferred from the buffer registers to the respective cycle set registers at the timing selected with the BF[1:0] bits in the MTU4.TADCR (MTU7.TADCR) register.

In complementary PWM mode, data is also transferred from the timer A/D converter start request cycle set buffer registers to the timer A/D converter start request cycle set registers when MTU4.TGRD (MTU7.TGRD) register is updated.

There are notes on the timing for transferring data when using buffer transfer in complementary PWM mode.

For details, section 24.6.28, Usage Notes on A/D Converter Delaying Function in Complementary PWM Mode.

In modes other than complementary PWM mode, set the BF1 bit in the MTU4.TADCR (MTU7.TADCR) register to 0.

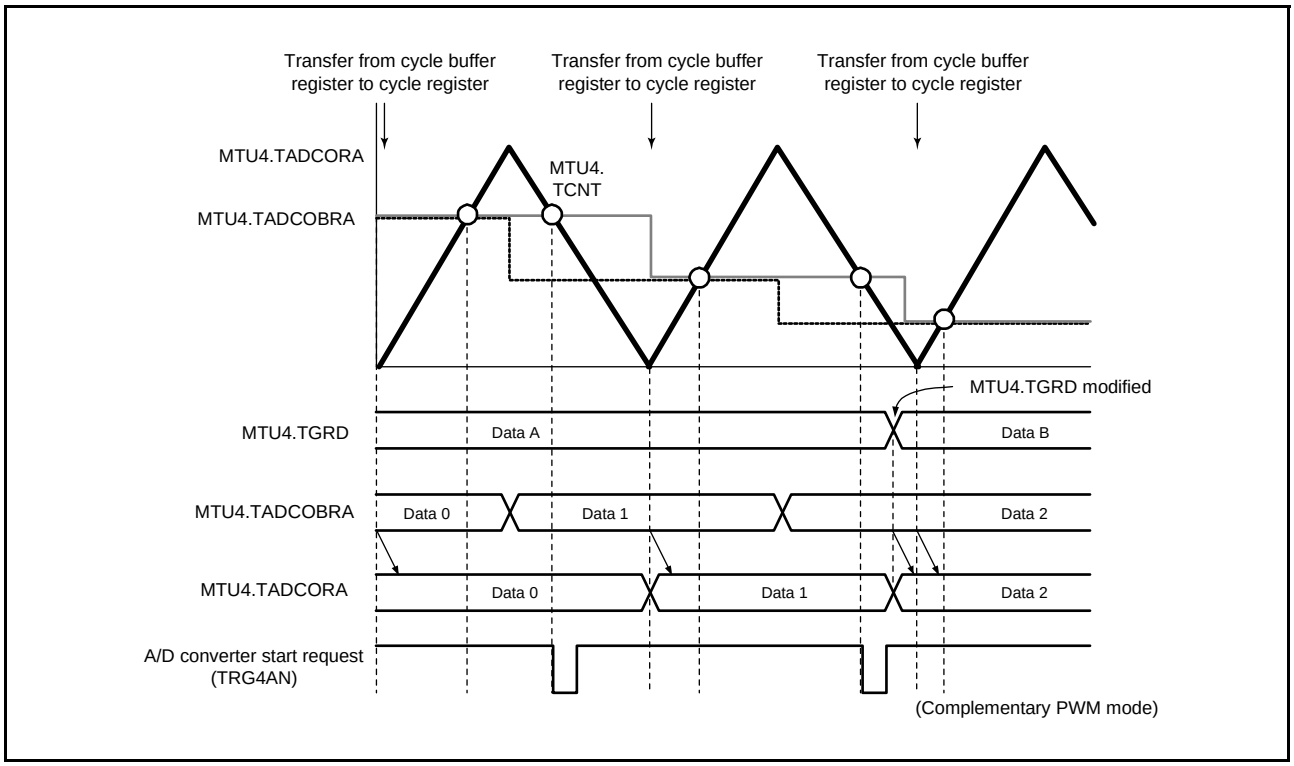


Figure 24.94 Example of A/D Converter Start Request Signal (TRG4AN) and Buffer Transfer Operation

(5) A/D Converter Start Request Delaying Function Linked with Interrupt Skipping Function 1

In complementary PWM mode, A/D converter start requests (TRG4AN and TRG4BN (TRG7AN and TRG7BN)) can be issued in coordination with interrupt skipping 1 by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE (ITA6AE, ITA7VE, ITB6AE, and ITB7VE) bits in the MTU4.TADCR (MTU7.TADCR) register.

Figure 24.95 shows an example of A/D converter start request signal (TRG4AN (TRG7AN)) operation when TRG4AN (TRG7AN) output is enabled during MTU4.TCNT (MTU7.TCNT) up-counting and down-counting and A/D converter start requests are linked with interrupt skipping 1.

Figure 24.96 shows another example of A/D converter start request signal (TRG4AN (TRG7AN)) operation when TRG4AN (TRG7AN) output is enabled during MTU4.TCNT (MTU7.TCNT) up-counting and A/D converter start requests are linked with interrupt skipping 1.

In modes other than complementary PWM mode, do not use the A/D converter start request delaying function linked with the interrupt skipping function 1.

Set the ITA3AE, ITA4VE, ITB3AE, and ITB4VE (ITA6AE, ITA7VE, ITB6AE, and ITB7VE) bits in the MTU4.TADCR (MTU7.TADCR) register to 0.

Note: This function should be used in combination with interrupt skipping 1.
When interrupt skipping is disabled (the T3AEN and T4VEN (T6AEN and T7VEN) bits in the timer interrupt skipping set register (TITCR1A (TITCR1B)) are cleared to 0 or the skipping count set bits (T3ACOR and T4VCOR (T6ACOR and T7VCOR)) in TITCR1A (TITCR1B) are cleared to 0), make sure that A/D converter start requests are not linked with interrupt skipping 1 (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE (ITA6AE, ITA7VE, ITB6AE, and ITB7VE) bits in the timer A/D converter start request control register (MTU4.TADCR (MTU7.TADCR)) to 0).
When this function is used, MTU4.TADCORA and MTU4.TADCORB (MTU7.TADCORA and MTU7.TADCORB) should be set with the value ranging 0002h to the value set in TCDRA minus 2 (value set in TCDRB minus 2).

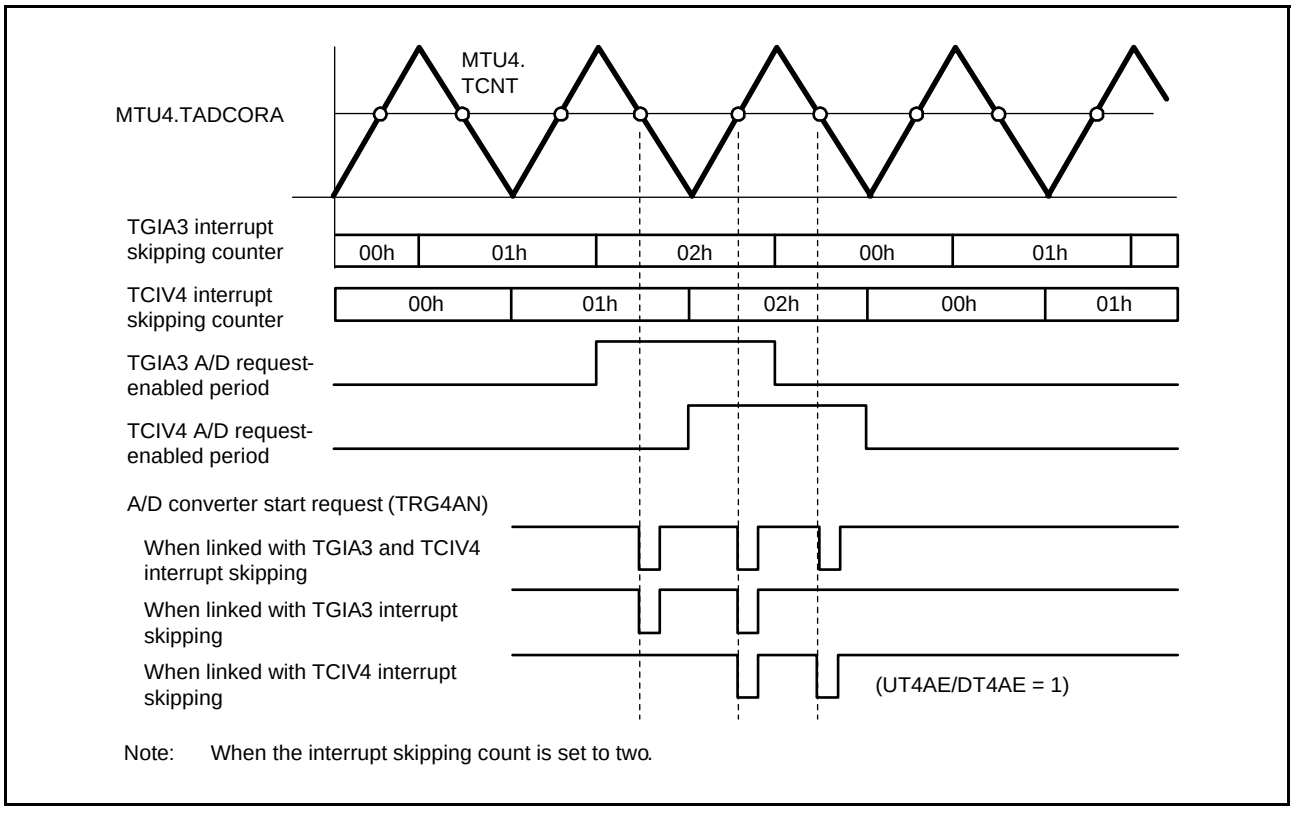


Figure 24.95 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping Function 1 (UT4AE and DT4AE = 1)

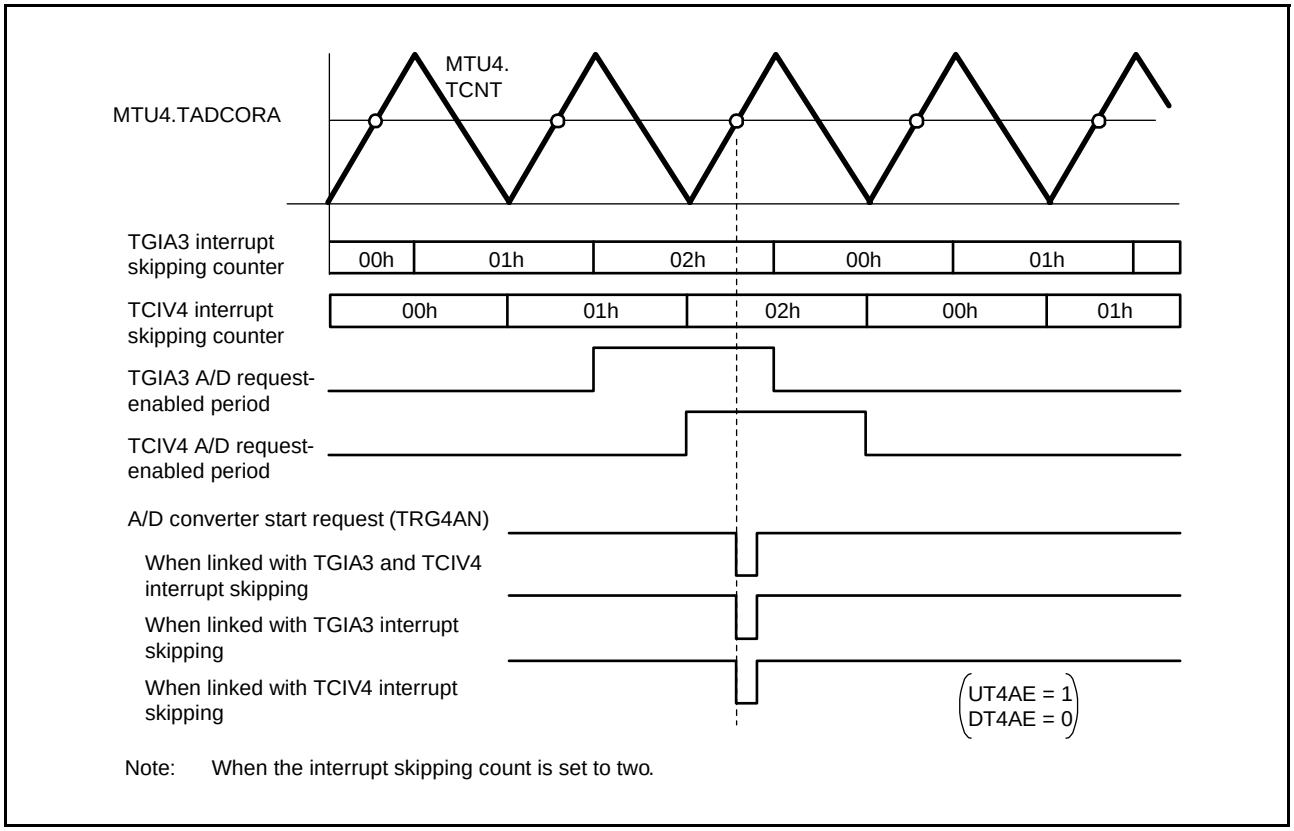


Figure 24.96 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping Function 1 (UT4AE = 1, DT4AE = 0)

(6) A/D Converter Start Request Delaying Function Linked with Interrupt Skipping Function 2

By setting the TITM bit to 1 in the TITMRA (TITMRB) register, the counter starts down-counting from the value (0 to 7) set in the TRG4COR[2:0] (TRG7COR[2:0]) bits in TITCR2A (TITCR2B) register every time an A/D converter start trigger (TRG4AN or TRG4BN (TRG7AN or TRG7BN)) is generated. When the counter value reaches 0 and is reloaded, the TRG4AN and TRG4BN (TRG7AN and TRG7BN) interrupts become valid and an A/D converter start request signal (TRG4ABN (TRG7ABN)) is output.
This function is valid only when the A/D converter request delaying function is enabled.

(a) Example of Procedure for Setting Interrupt Skipping Function 2

Figure 24.97 shows an example of procedure for setting interrupt skipping function 2.

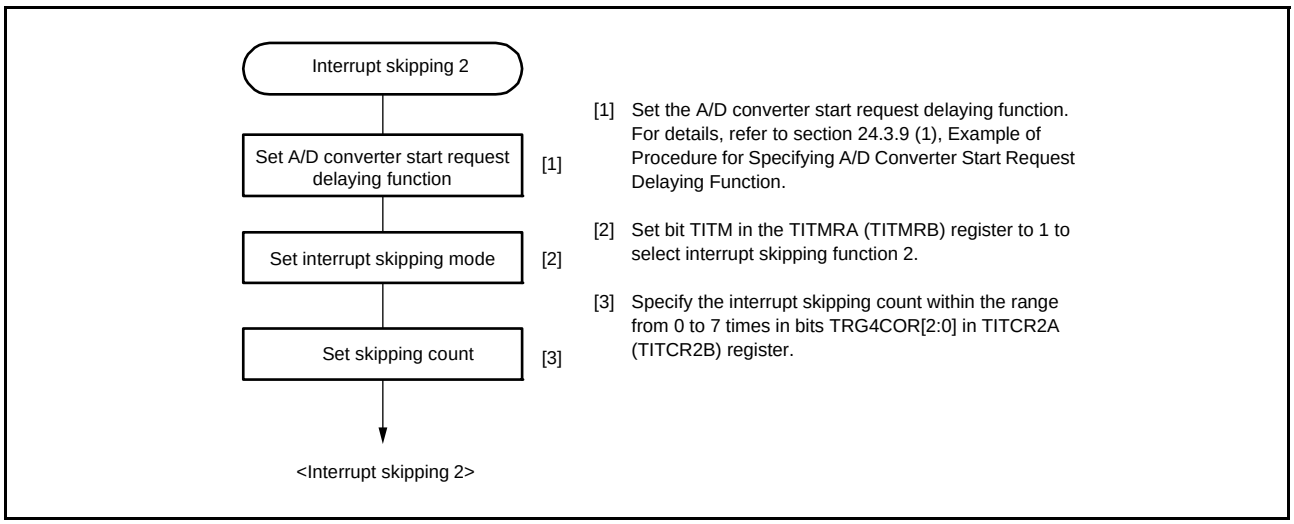


Figure 24.97 Example of Procedure for Setting Interrupt Skipping Function 2

(b) Example of Interrupt Skipping Function 2 Operation

Figure 24.98 shows an example of interrupt skipping function 2 operation.

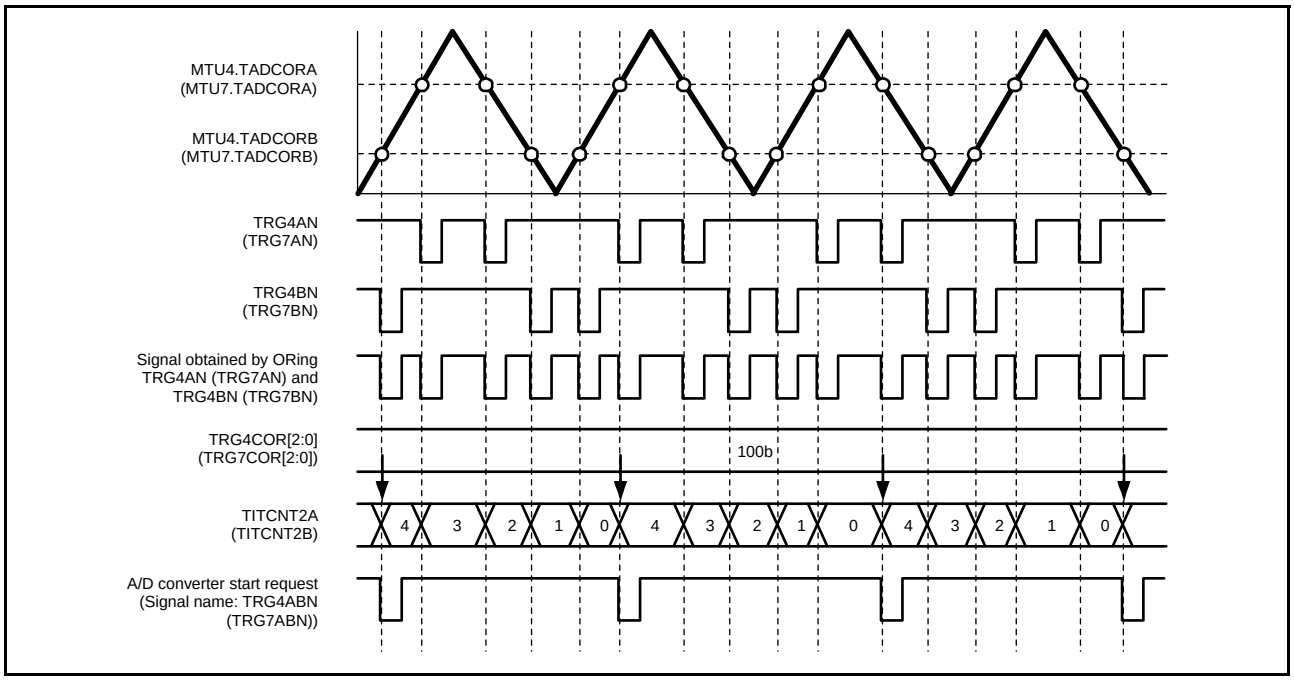


Figure 24.98 Example of Interrupt Skipping Function 2 Operation (Skipping Count is Set to Four)

24.3.10 Synchronous Operation of MTU0 to MTU4, MTU6, and MTU7

(1) Synchronous Counter Start for MTU0 to MTU4, MTU6, and MTU7

The counters in MTU0 to MTU4, MTU6, and MTU7 can be started synchronously by making the TCSYSTR settings.

(a) Example of Procedure for Setting Synchronous Counter Start for MTU0 to MTU4, MTU6, and MTU7

Figure 24.99 shows an example of procedure for setting synchronous counter start for MTU0 to MTU4, MTU6, and MTU7.

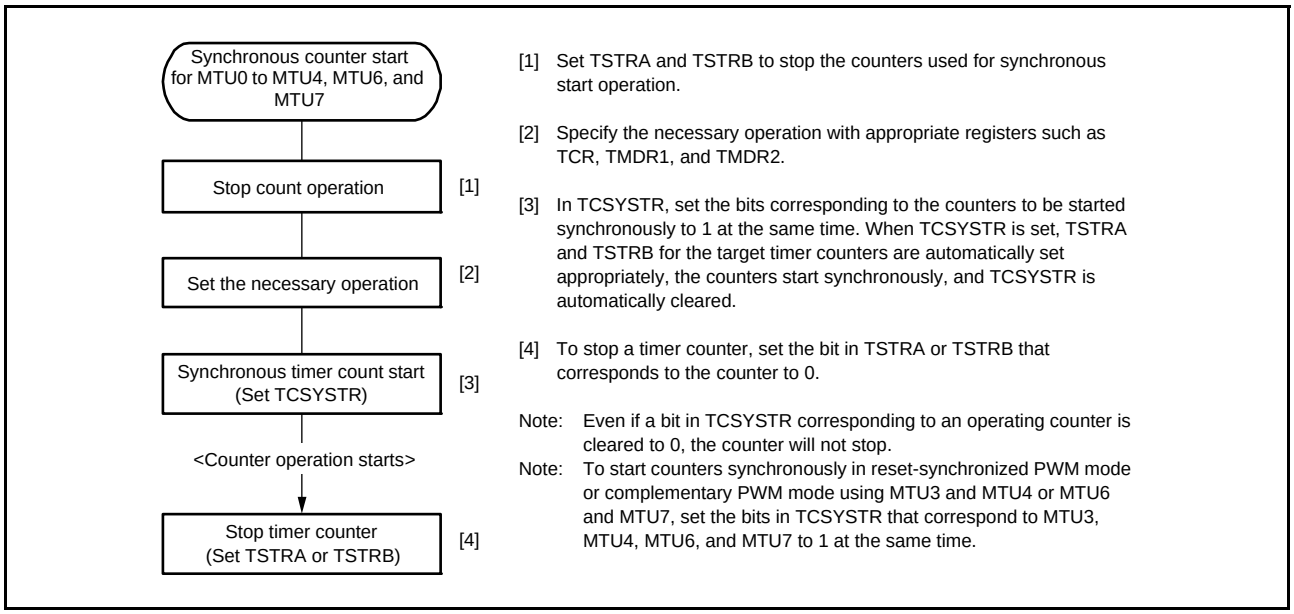


Figure 24.99 Example of Procedure for Setting Synchronous Counter Start for MTU0 to MTU4, MTU6, and MTU7

(b) Examples of Synchronous Counter Start Operation

Figure 24.100 shows an examples of synchronous counter start operation for MTU0 to MTU4, MTU6, and MTU7.

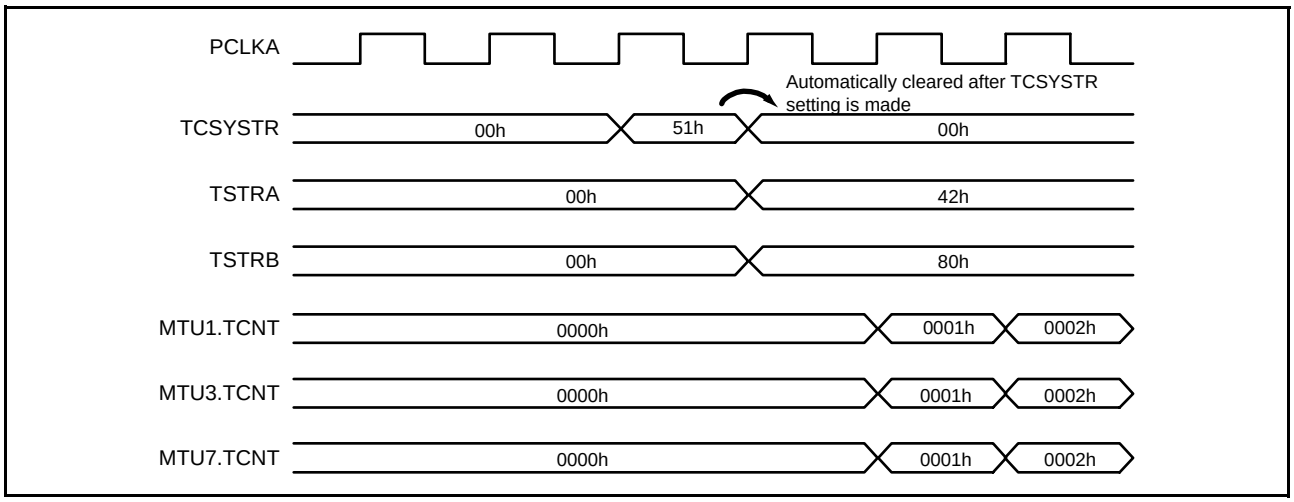


Figure 24.100 Examples of Synchronous Counter Start Operation for MTU0 to MTU4, MTU6, and MTU7

(2) Synchronous Counter Clearing for MTU6 and MTU7

The counters in MTU6 and MTU7 can be cleared by the TGImn interrupt generation timing (m = A to D; n = 0 to 2) through the TSYCR setting.

(a) Example of Procedure for Specifying Synchronous Counter Clearing for MTU6 and MTU7

Figure 24.101 shows an example of procedure for specifying synchronous counter clearing for MTU6 and MTU7 by interrupt generation timing.

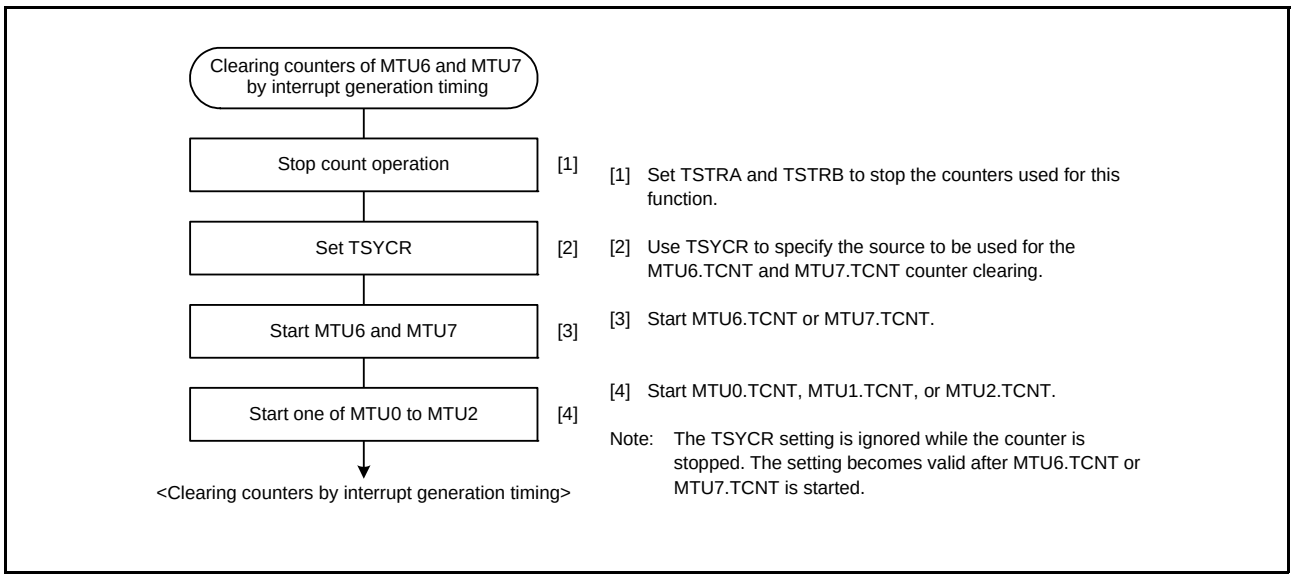


Figure 24.101 Example of Procedure for Specifying Synchronous Counter Clearing for MTU6 and MTU7

(b) Examples of Synchronous Counter Clearing for MTU6 and MTU7

Figure 24.102 and Figure 24.103 show examples of synchronous counter clearing for MTU6 and MTU7 by interrupt generation timing.

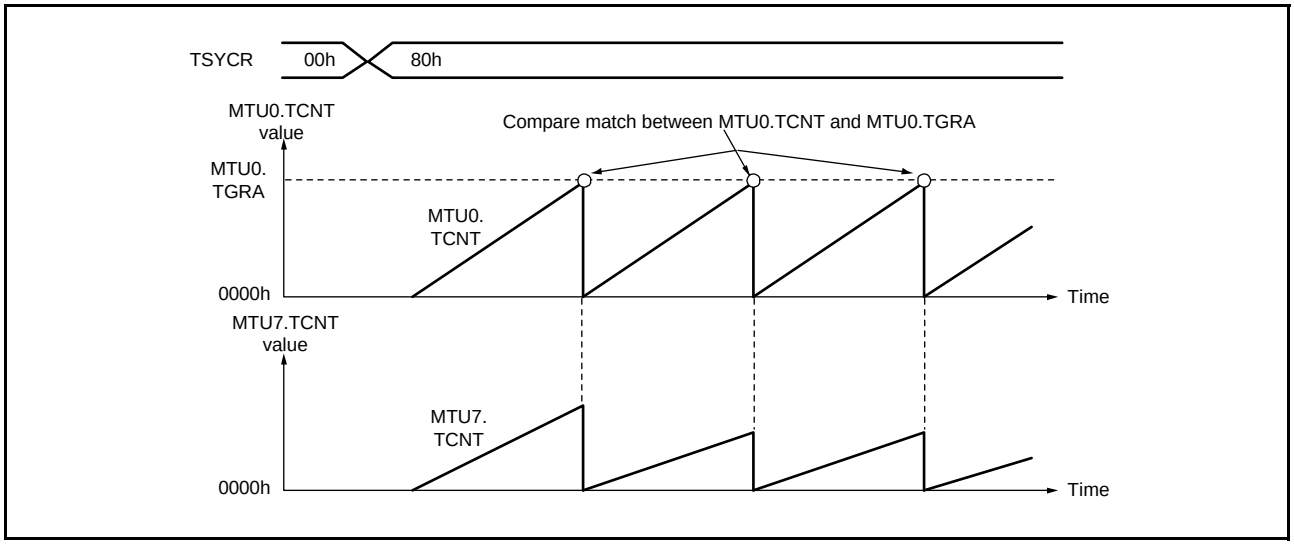


Figure 24.102 Example of Synchronous Counter Clearing for MTU6 and MTU7 (1)

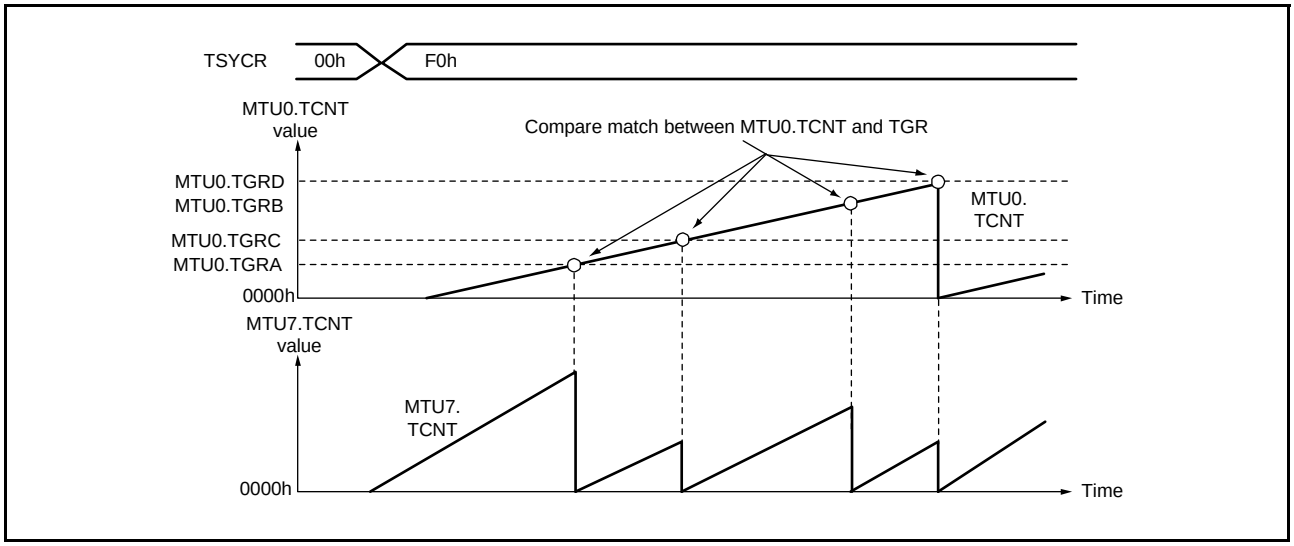


Figure 24.103 Example of Synchronous Counter Clearing for MTU6 and MTU7 (2)

24.3.11 External Pulse Width Measurement

The pulse widths of up to three external input lines can be measured in MTU5.
When the IOC[4:0] bits in MTU5.TIORU, MTU5.TIORV, MTU5.TIORW are set for pulse width measurement, the pulse width of the signal input to the MTIC5U, MTIC5V, and MTIC5W pins are measured. TCNTU, TCNTV, and TCNTW count up while the level specified by the IOC[4:0] bits is input.
Figure 24.104 shows an example of setting external pulse width measurement, and Figure 24.105 an example of external pulse width measurement.

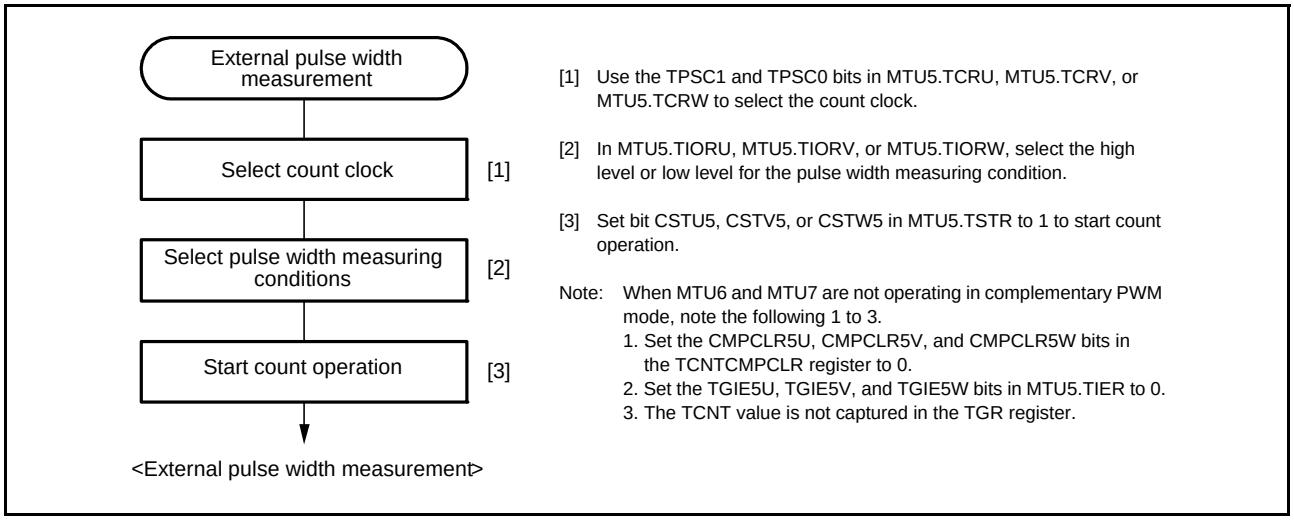


Figure 24.104 Example of External Pulse Width Measurement Setting Procedure

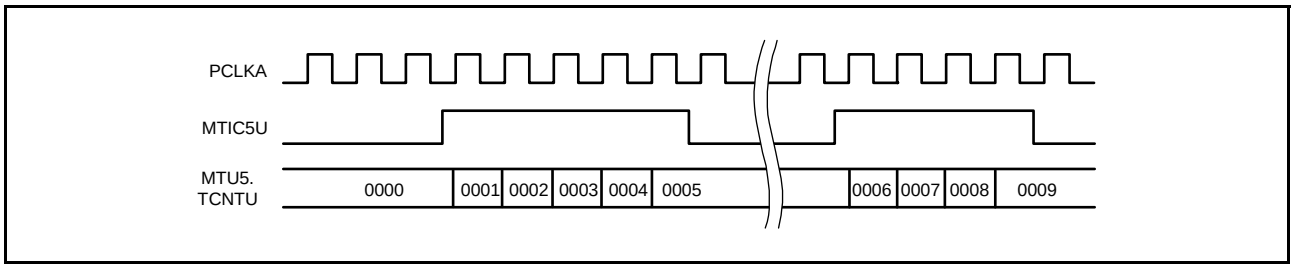


Figure 24.105 Example of External Pulse Width Measurement (Measuring High Pulse Width)

24.3.12 Dead Time Compensation

Figure 24.106 shows an example of the motor control circuit used to feed back a delay in the dead time (delay between complementary PWM output and inverter output) to MTU5. The MTU5 external pulse measurement function allows the delay between the complementary PWM output and inverter output to be measured and reflected in the duty ratio, which can be used as dead time compensation for the PWM output waveform in complementary PWM operation when MTU6 and MTU7 are used (Figure 24.107). Figure 24.108 shows the procedure for setting dead time compensation using MTU5. For details on MTU5 operation at this time, refer to section 24.3.13, TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode.

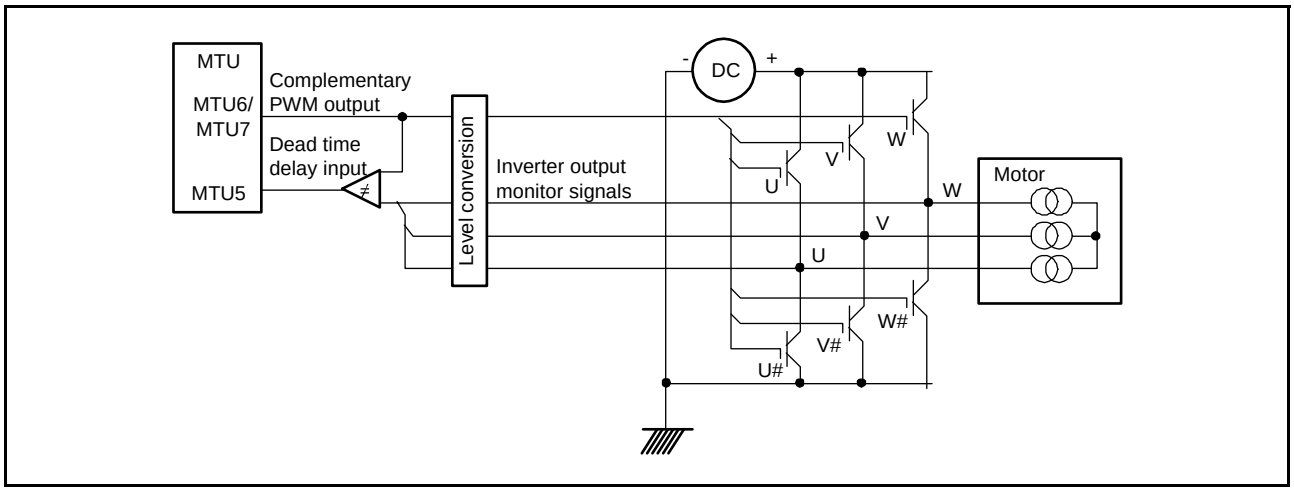


Figure 24.106 Motor Control Circuit Example

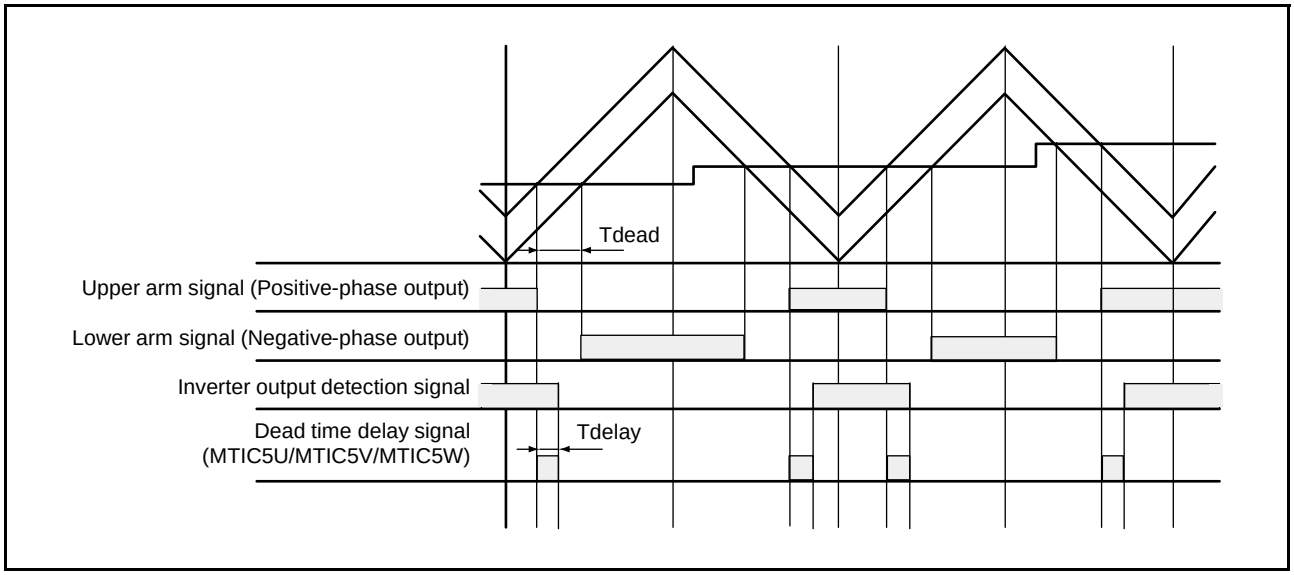
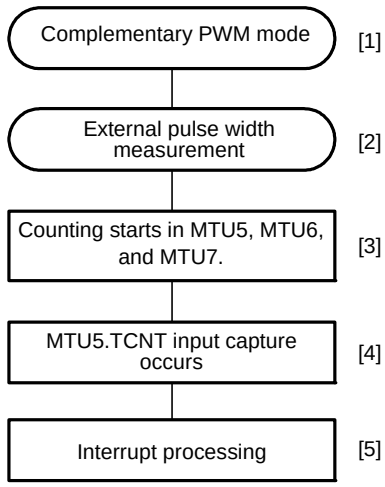


Figure 24.107 Delay in Dead Time in Complementary PWM Operation

(1) Example of Dead Time Compensation Setting Procedure

Figure 24.108 shows an example of dead time compensation setting procedure by using three counters in MTU5.



[1] Set MTU6 and MTU7 to complementary PWM mode. For details, refer to section 24.3.8, Complementary PWM Mode.

[2] Specify the external pulse width measurement function for the target TIOR in MTU5. For details, refer to section 24.3.11, External Pulse Width Measurement.

[3] Set bits CST6 and CST7 in TSTRB and bits CSTU5, CSTV5, and CSTW5 in MTU5.TSTR to 1 to start count operation.

[4] When the capture condition specified in TIOR is satisfied, the MTU5.TCNT value is captured in MTU5.TGR.
To prevent the generation of the MTU5.TGIm5 interrupt (m = U, V, W) request under the capture condition specified in TIOR, do not enable interrupt requests in MTU5.TIER.

[5] For U-phase dead time compensation, when an interrupt is generated at the crest (TGIA6) or trough (TCIV7) in complementary PWM mode, read the MTU5.TGRU value, calculate the difference in time between MTU6.TGRB and MTU5.TGRU, and write the corrected value to MTU6.TGRD in the interrupt processing.
For the V phase and W phase, read the MTU5.TGRV and MTU5.TGRW values and write the corrected values to MTU7.TGRC and TGRD, respectively, in the same way as for U-phase compensation.
The MTU5.TCNT value should be cleared through the TCNTCMPCLR setting or by software.

Note: The MPC and port mode registers (PMR) of the I/O ports should be set up in advance.
As the interrupt flag is set under the capture condition specified in TIOR, do not enable interrupt requests in MTU5.TIER.

Figure 24.108 Example of Dead Time Compensation Setting Procedure

24.3.13 TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode

The MTU5 external pulse width measurement function can be used to transfer the value in TCNTU, TCNTV, and TCNTW to TGRU, TGRV, and TGRW at the crest, or trough, or crest and trough. The transfer timing is set in TIORU, TIORV, and TIORW. When the CMPCLR5U, CMPCLR5V, and CMPCLR5W bits in the TCNTCMPCLR register are set to 1, TCNTU, TCNTV, and TCNTW are cleared to 0 at the transfer timing for TGRU, TGRV, and TGRW.
Figure 24.109 shows an operation example in which TCNTU is used as a free-running counter without being cleared, and the value is captured in TGRU at the crest or trough in complementary PWM mode.

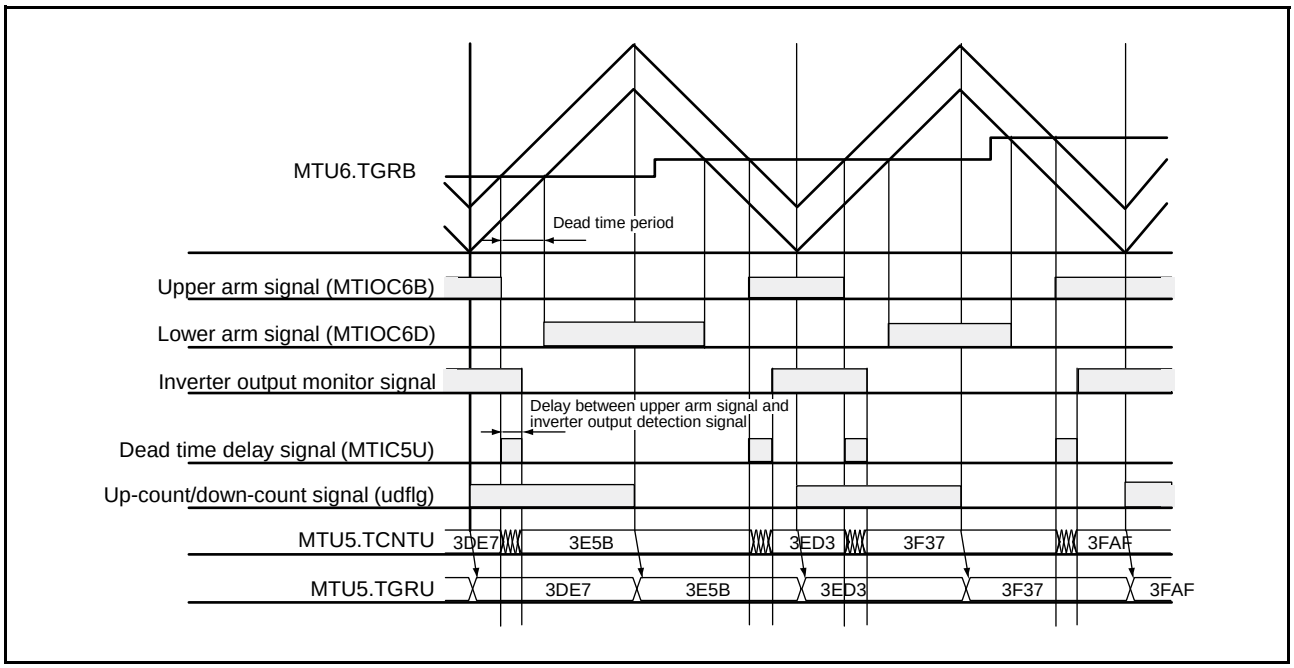


Figure 24.109 TCNTU Capture at Crest and/or Trough in Complementary PWM Operation

24.3.14 Noise Filter Function

The input capture input pins and external pulse input pins have a noise filter function.
Set the NFCRn register (n = 0 to 7, C) to enable or disable the noise filter function and set the sampling clock. The noise filter for each pin can be enabled or disabled individually, and the sampling clock can be set for each channel.
Figure 24.110 shows the timing of noise filtering.

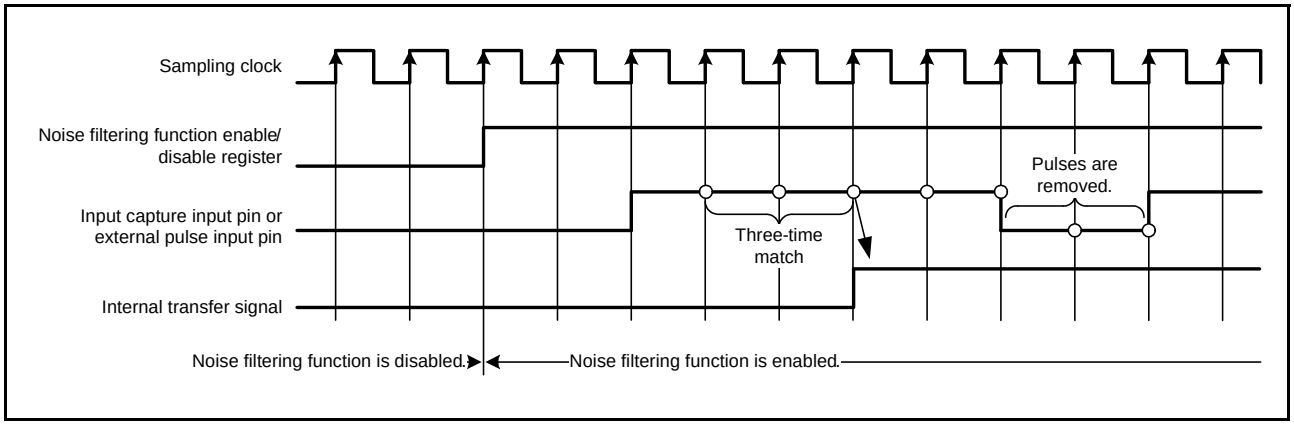


Figure 24.110 Timing of Noise Filtering

24.4 Interrupt Sources

24.4.1 Interrupt Sources and Priorities

There are three kinds of interrupt source; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own enable/disable bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt source is generated, if the corresponding enable/disable bit in TIER is set to 1, an interrupt is requested.

Relative channel priorities can be changed by the interrupt controller; however the priority within a channel is fixed. For details, refer to section 15, Interrupt Controller (ICUA). Table 24.78 lists the MTU interrupt sources.

Table 24.78 MTU Interrupt Sources

Channel	Name	Interrupt Source	DMAC/DTC Activation
MTU0	TGIA0	MTU0.TGRA input capture/compare match	Possible
	TGIB0	MTU0.TGRB input capture/compare match	Possible
	TGIC0	MTU0.TGRC input capture/compare match	Possible
	TGID0	MTU0.TGRD input capture/compare match	Possible
	TCIV0	MTU0.TCNT overflow	Not possible
	TGIE0	MTU0.TGRE compare match	Not possible
	TGIF0	MTU0.TGRF compare match	Not possible
MTU1	TGIA1	MTU1.TGRA input capture/compare match	Possible
	TGIB1	MTU1.TGRB input capture/compare match	Possible
	TCIV1	MTU1.TCNT overflow	Not possible
	TCIU1	MTU1.TCNT underflow	Not possible
MTU2	TGIA2	MTU2.TGRA input capture/compare match	Possible
	TGIB2	MTU2.TGRB input capture/compare match	Possible
	TCIV2	MTU2.TCNT overflow	Not possible
	TCIU2	MTU2.TCNT underflow	Not possible
MTU3	TGIA3	MTU3.TGRA input capture/compare match	Possible
	TGIB3	MTU3.TGRB input capture/compare match	Possible
	TGIC3	MTU3.TGRC input capture/compare match	Possible
	TGID3	MTU3.TGRD input capture/compare match	Possible
	TCIV3	MTU3.TCNT overflow	Not possible
MTU4	TGIA4	MTU4.TGRA input capture/compare match	Possible
	TGIB4	MTU4.TGRB input capture/compare match	Possible
	TGIC4	MTU4.TGRC input capture/compare match	Possible
	TGID4	MTU4.TGRD input capture/compare match	Possible
	TCIV4	MTU4.TCNT overflow/underflow*1	Possible
MTU5	TGIU5	MTU5.TGRU input capture/compare match	Possible
	TGIV5	MTU5.TGRV input capture/compare match	Possible
	TGIW5	MTU5.TGRW input capture/compare match	Possible
MTU6	TGIA6	MTU6.TGRA input capture/compare match	Possible
	TGIB6	MTU6.TGRB input capture/compare match	Possible
	TGIC6	MTU6.TGRC input capture/compare match	Possible
	TGID6	MTU6.TGRD input capture/compare match	Possible
	TCIV6	MTU6.TCNT overflow	Not possible
MTU7	TGIA7	MTU7.TGRA input capture/compare match	Possible
	TGIB7	MTU7.TGRB input capture/compare match	Possible
	TGIC7	MTU7.TGRC input capture/compare match	Possible
	TGID7	MTU7.TGRD input capture/compare match	Possible
	TCIV7	MTU7.TCNT overflow/underflow*1	Possible
MTU8	TGIA8	MTU8.TGRA input capture/compare match	Possible
	TGIB8	MTU8.TGRB input capture/compare match	Possible
	TGIC8	MTU8.TGRC input capture/compare match	Possible
	TGID8	MTU8.TGRD input capture/compare match	Possible
	TCIV8	MTU8.TCNT overflow	Possible

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

Note 1. Only in complementary PWM mode

(1) Input Capture/Compare Match Interrupt

If the TIER.TGIE bit is set to 1 when a TGR input capture/compare match occurs on a channel, an interrupt is requested. The MTU has 33 input capture/compare match interrupts (six for MTU0, four each for MTU3, MTU4, MTU6, MTU7, and MTU8, two each for MTU1 and MTU2, and three for MTU5).

(2) Overflow Interrupt

If the TIER.TCIEV bit is set to 1 when a TCNT overflow occurs on a channel, clearing an interrupt is requested. The MTU has eight overflow interrupts (one for each channel except MTU5).

(3) Underflow Interrupt

If the TIER.TCIEU bit is set to 1 when a TCNT underflow occurs on a channel, an interrupt is requested. The MTU has two underflow interrupts (one each for MTU1, and MTU2).

24.4.2 DTC/DMAC Activation

(1) DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt in each channel or the overflow interrupt in MTU4 and MTU7. For details, refer to section 20, Data Transfer Controller (DTCa).
The MTU provides a total of 33 input capture/compare match interrupts and overflow interrupts that can be used as DTC activation sources: four each for MTU0, MTU3, MTU6, and MTU8, two each for MTU1 and MTU2, five each for MTU4 and MTU7, and three for MTU5.

(2) DMAC Activation

The DMAC can be activated by the TGRA input capture/compare match interrupt and the overflow interrupt in MTU4 and MTU7. For details, refer to section 18, DMA Controller (DMACAa).
The MTU provides a total of 33 input capture/compare match interrupts that can be used as DMAC activation sources: four each for MTU0, MTU3, MTU6, and MTU8, two each for MTU1 and MTU2, five each for MTU4 and MTU7, and three for MTU5.
In DMAC activation by the MTU, the activation trigger signal is cleared when the DMAC requests mastership of the internal bus. Accordingly, the state of the internal bus may lead to a wait for the start of DMAC transfer.

(3) EXDMAC Activation

The EXDMAC can be activated by MTU interrupts. For details, refer to section 19, EXDMA Controller (EXDMACa).

24.4.3 A/D Converter Activation

The A/D converter can be activated by one of the following three methods in the MTU. Table 24.79 shows the relationship between interrupt sources and A/D converter start request signals.

(1) A/D Converter Activation by TGRA Input Capture/Compare Match or at MTU4.TCNT (MTU7.TCNT) Trough in Complementary PWM Mode

The A/D converter can be activated by the occurrence of a TGRA input capture/compare match in each channel. In addition, if complementary PWM operation is performed while the TTGE2 bit in MTU4.TIER (MTU7.TIER) is set to 1, the A/D converter can be activated at the trough of MTU4.TCNT (MTU7.TCNT) count (MTU4.TCNT (MTU7.TCNT) = 0000h).
A/D converter start request signal TRGAnN is issued to the A/D converter under either of the following conditions (n = 0 to 4, 6, 7).

- When a TGRA input capture/compare match occurs on a channel while the TIER.TTGE bit is set to 1
- When the MTU4.TCNT (MTU7.TCNT) count reaches the trough (MTU4.TCNT (MTU7.TCNT) = 0000h) during complementary PWM operation while the TTGE2 bit in MTU4.TIER (MTU7.TIER) is set to 1

When either condition is satisfied, if A/D converter start signal TRGAnN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(2) A/D Converter Activation by Compare Match between MTU0.TCNT and MTU0.TGRE

A/D converter start request signal TRG0N is issued to the A/D converter when a compare match occurs between MTU0.TCNT and MTU0.TGRE.
When a compare match occurs between MTU0.TCNT and MTU0.TGRE while the TTGE2 bit in MTU0.TIER2 is set to 1, A/D converter start request TRG0N is issued to the A/D converter. If A/D converter start signal TRG0N from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(3) A/D Converter Activation by A/D Converter Start Request Delaying Function

The A/D converter can be activated by generating A/D converter start request signal TRG4AN or TRG4BN (TRG7AN or TRG7BN) when the MTU4.TCNT (MTU7.TCNT) count matches the MTU4.TADCORA or MTU4.TADCORB (MTU7.TADCORA or MTU7.TADCORB) value if the UT4AE, DT4AE, UT4BE, or DT4BE (UT7AE, DT7AE, UT7BE, or DT7BE) bit in the A/D converter start request control register (MTU4.TADCR (MTU7.TADCR)) is set to 1. For details, refer to section 24.3.9, A/D Converter Start Request Delaying Function.
A/D conversion will start when TRG4AN (TRG7AN) is generated if A/D converter start signal TRG4AN (TRG7AN) from the MTU is selected as the trigger in the A/D converter, when TRG4BN (TRG7BN) is generated if TRG4BN (TRG7BN) from the MTU is selected as the trigger in the A/D converter, or when TRG4ABN (TRG7ABN) is generated if TRG4ABN (TRG7ABN) from the MTU is selected as the trigger in the A/D converter.

Table 24.79 Interrupt Sources and A/D Converter Start Request Signals

Target Registers	Interrupt Source	A/D Converter Start Request Signal
MTU0.TGRA and MTU0.TCNT	Input capture/compare match	TRGA0N
MTU1.TGRA and MTU1.TCNT		TRGA1N
MTU2.TGRA and MTU2.TCNT		TRGA2N
MTU3.TGRA and MTU3.TCNT		TRGA3N
MTU4.TGRA and MTU4.TCNT*1		TRGA4N
MTU4.TCNT	MTU4.TCNT trough in complementary PWM mode	
MTU6.TGRA and MTU6.TCNT	Input capture/compare match	TRGA6N
MTU7.TGRA and MTU7.TCNT*1		TRGA7N
MTU7.TCNT	MTU7.TCNT trough in complementary PWM mode	
MTU0.TGRE and MTU0.TCNT	Compare match	TRG0N
MTU4.TADCORA and MTU4.TCNT		TRG4AN
MTU4.TADCORB and MTU4.TCNT		TRG4BN
MTU7.TADCORA and MTU7.TCNT		TRG7AN
MTU7.TADCORB and MTU7.TCNT		TRG7BN
MTU4.TADCORA and MTU4.TCNT, MTU4.TADCORB and MTU4.TCNT	Compare match (interrupt skipping function 2)	TRG4ABN
MTU7.TADCORA and MTU7.TCNT, MTU7.TADCORB and MTU7.TCNT		TRG7ABN

Note 1. Since PWM waveforms are generated in complementary PWM mode, MTU4.TGRA (MTU7.TGRA) compare match not only with MTU4.TCNT (MTU7.TCNT) but also with MTU3.TCNT (MTU6.TCNT) and TCNTSA (TCNTSB) is detected. Accordingly, when compare match with MTU3.TCNT (MTU6.TCNT) and TCNTSA (TCNTSB) occurs, TRGA4N (TRGA7N) is also generated. When MTU3 and MTU 4 (MTU 6 and MTU7) are made to operate in complementary PWM mode for generating an A/D converter start request, use the A/D converter start request by compare match between MTU4.TCNT (MTU7.TCNT) and MTU4.TADCORA/TADCORB (MTU7.TADCORA/TADCORB).

24.5 Operation Timing

24.5.1 Input/Output Timing

(1) TCNT Count Timing

Figure 24.111 and Figure 24.112 show the TCNT count timing in internal clock operation, Figure 24.113 shows the TCNT count timing in external clock operation (normal mode), and Figure 24.114 shows the TCNT count timing in external clock operation (phase counting mode).

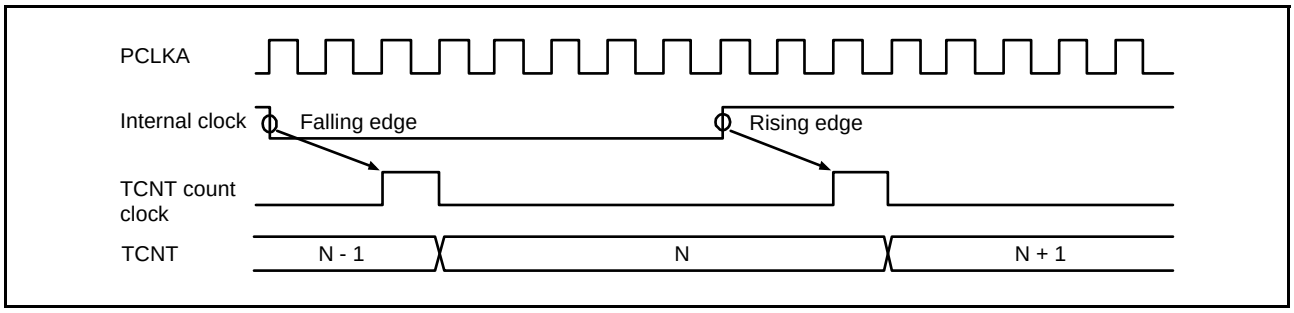


Figure 24.111 Count Timing in Internal Clock Operation (MTU0 to MTU4 and MTU6 to MTU8)

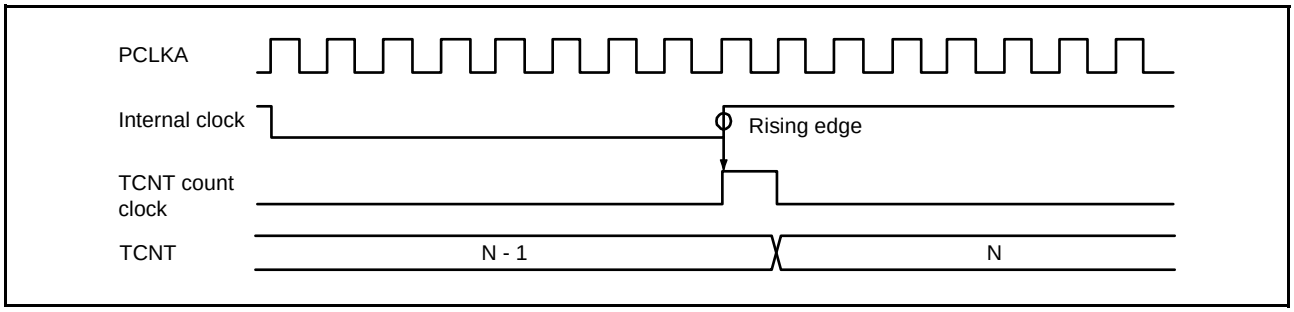


Figure 24.112 Count Timing in Internal Clock Operation (MTU5)

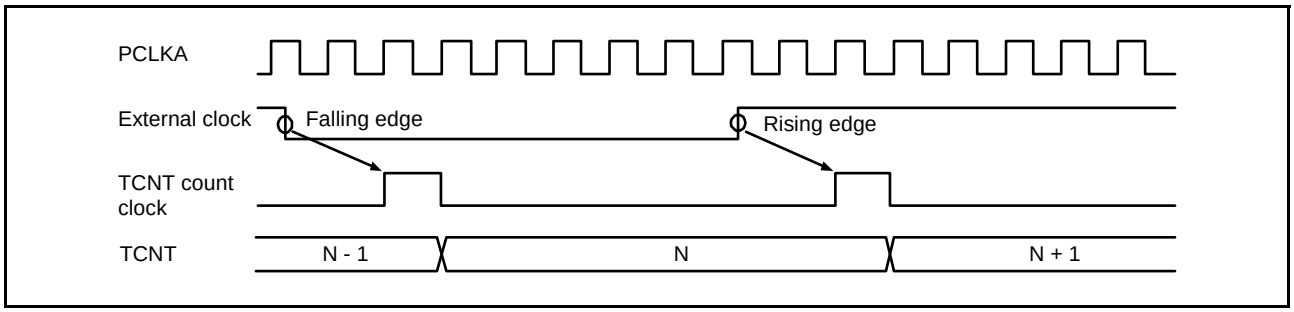


Figure 24.113 Count Timing in External Clock Operation (MTU0 to MTU4 and MTU6 to MTU8)

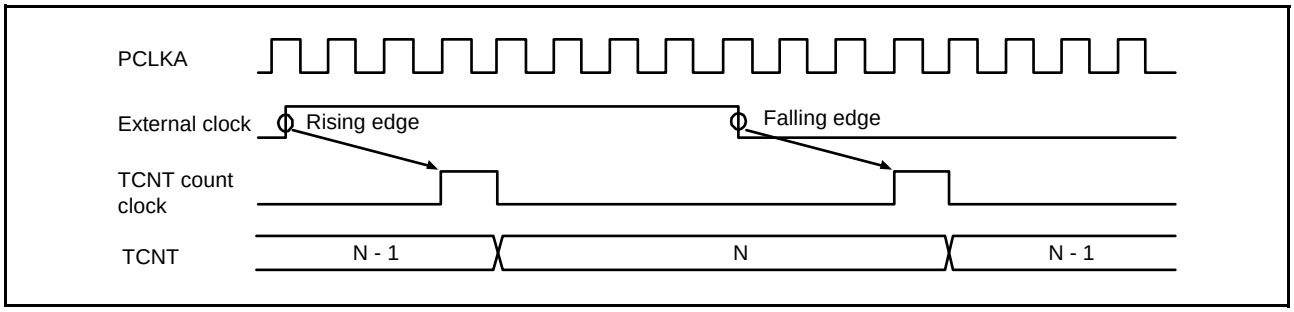


Figure 24.114 Count Timing in External Clock Operation (Phase Counting Mode)

(2) Output Compare Output Timing

A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the value set in TIOR is output from MTIOCnm pin (n = 0 to 4, 6, 7, 8; m = A to D). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT count clock is generated.

Figure 24.115 shows the output compare output timing (normal mode or PWM mode) and Figure 24.116 shows the output compare output timing (complementary PWM mode or reset-synchronized PWM mode).

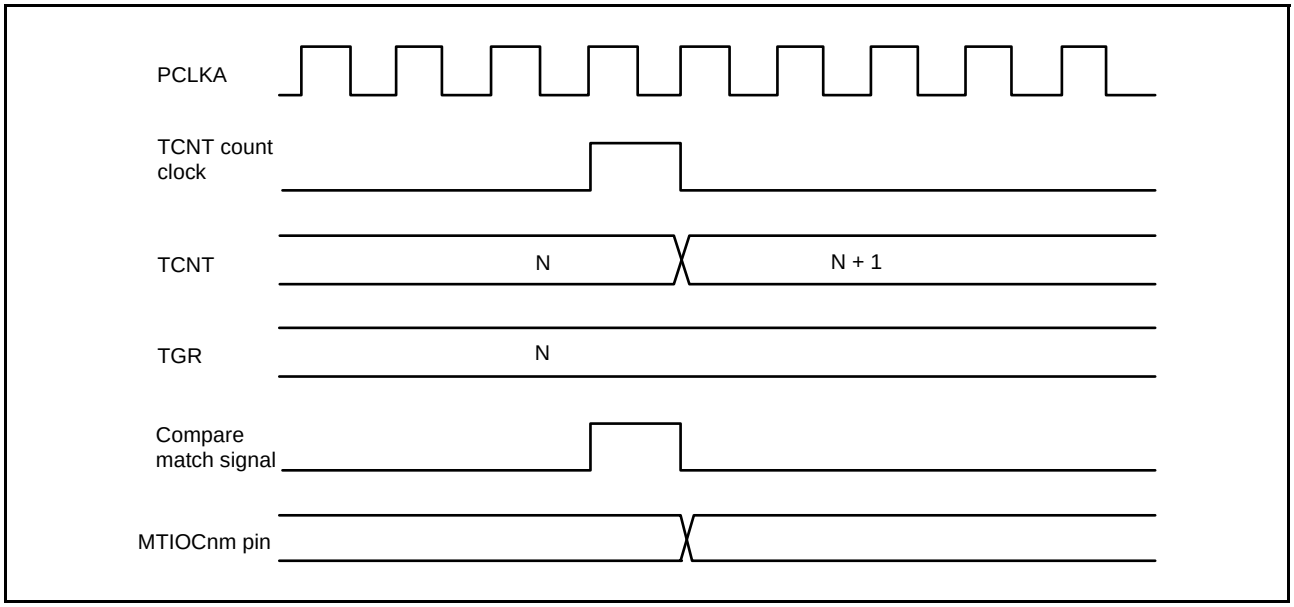


Figure 24.115 Output Compare Output Timing (Normal Mode or PWM Mode) (n = 0 to 4, 6, 7, 8; m = A to D)

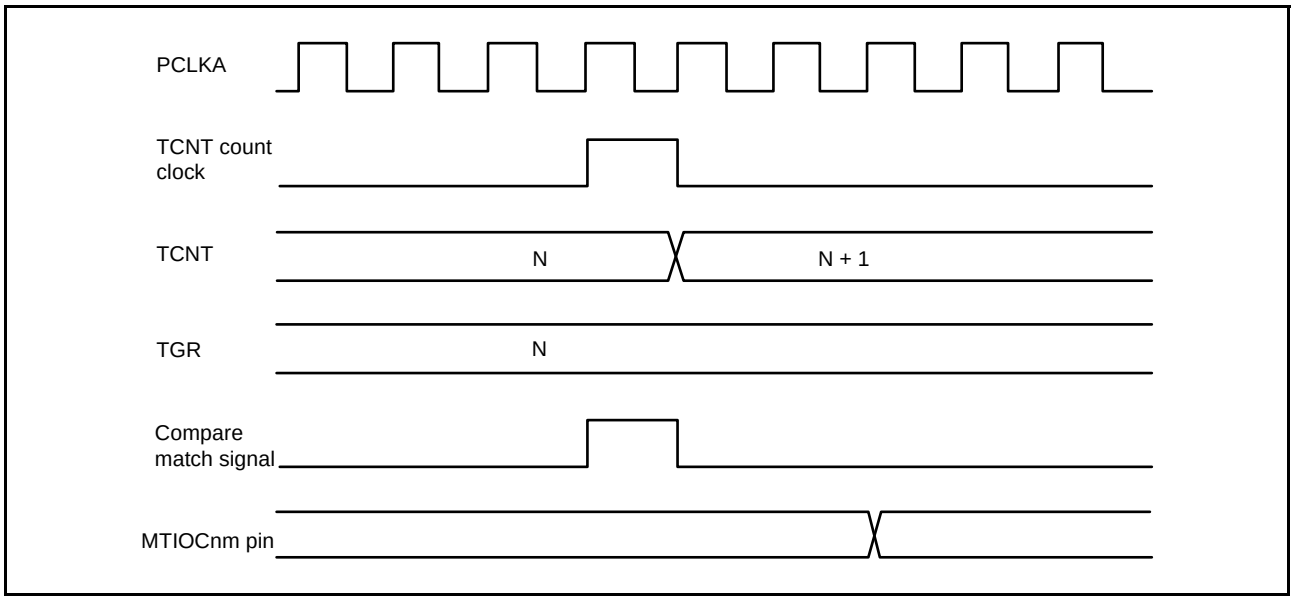


Figure 24.116 Output Compare Output Timing (Complementary PWM Mode or Reset-Synchronized PWM Mode) (n = 0 to 4, 6, 7, 8; m = A to D)

(3) Input Capture Signal Timing

Figure 24.117 shows the input capture signal timing.

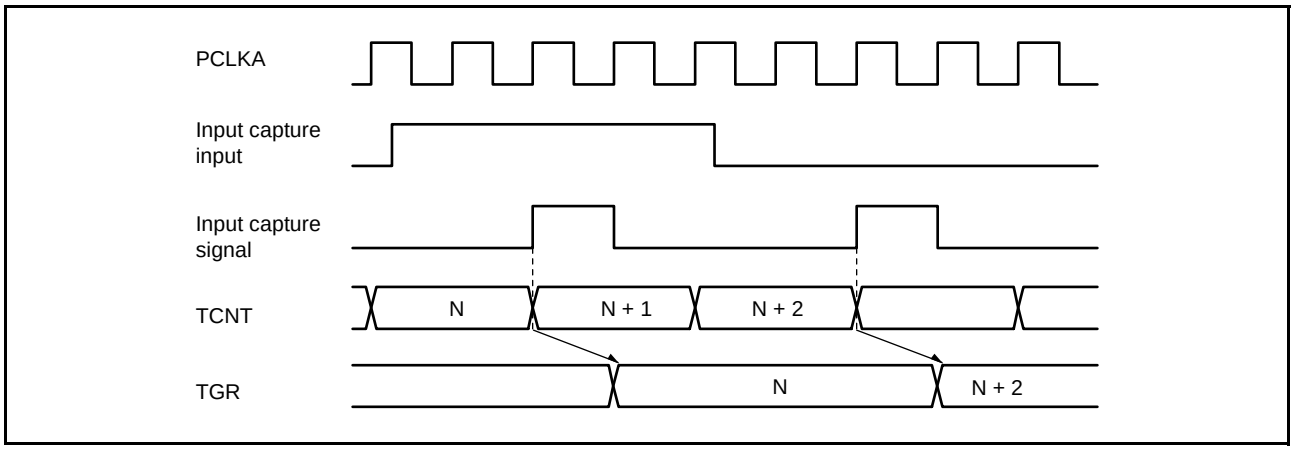


Figure 24.117 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

FFigure 24.118 and Figure 24.119 show the timing when counter clearing on compare match is specified, and Figure 24.120 shows the timing when counter clearing on input capture is specified.

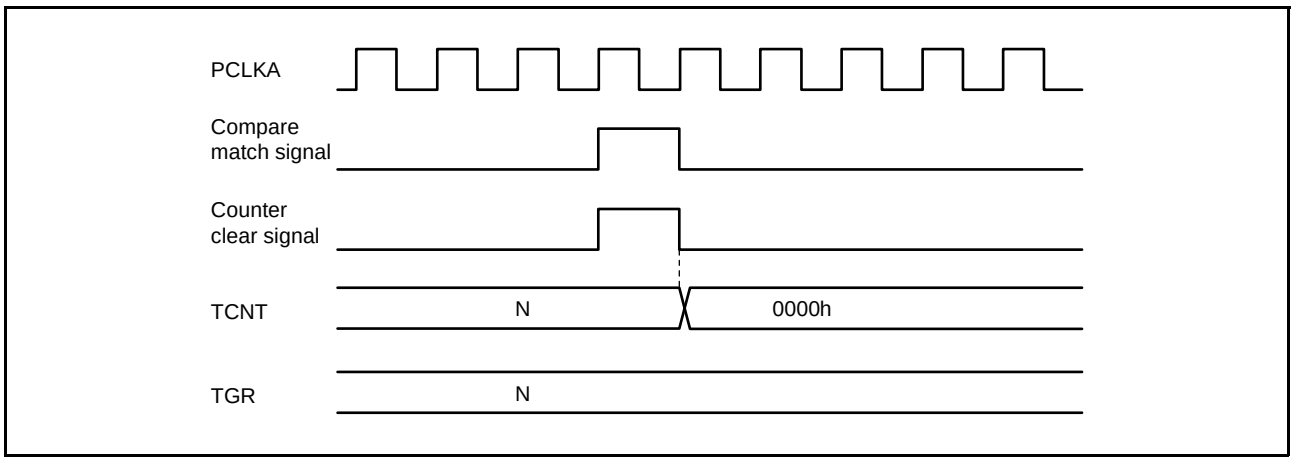


Figure 24.118 Counter Clear Timing (Compare Match) (MTU0 to MTU4 and MTU6 to MTU8)

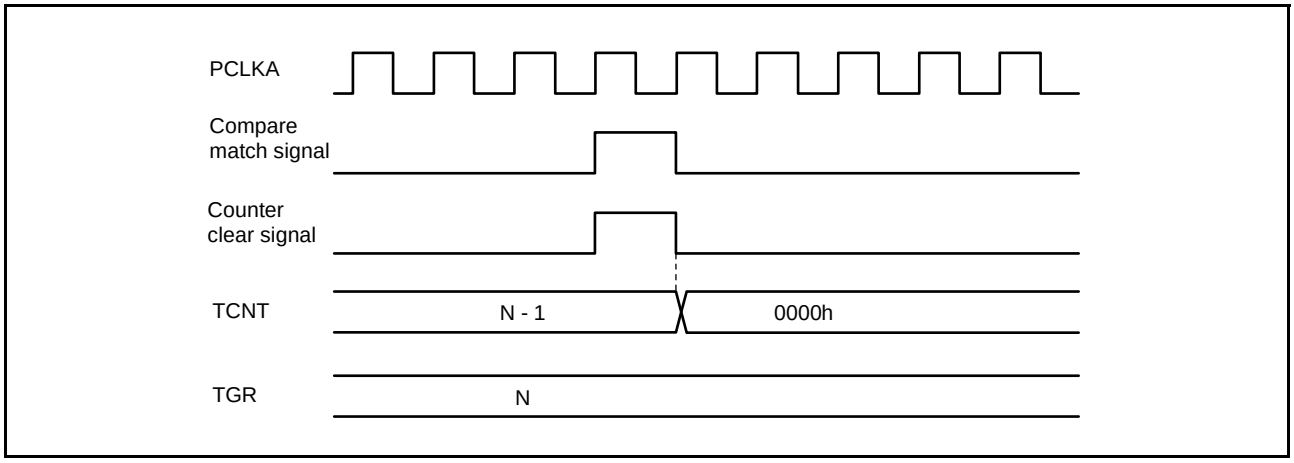


Figure 24.119 Counter Clear Timing (Compare Match) (MTU5)

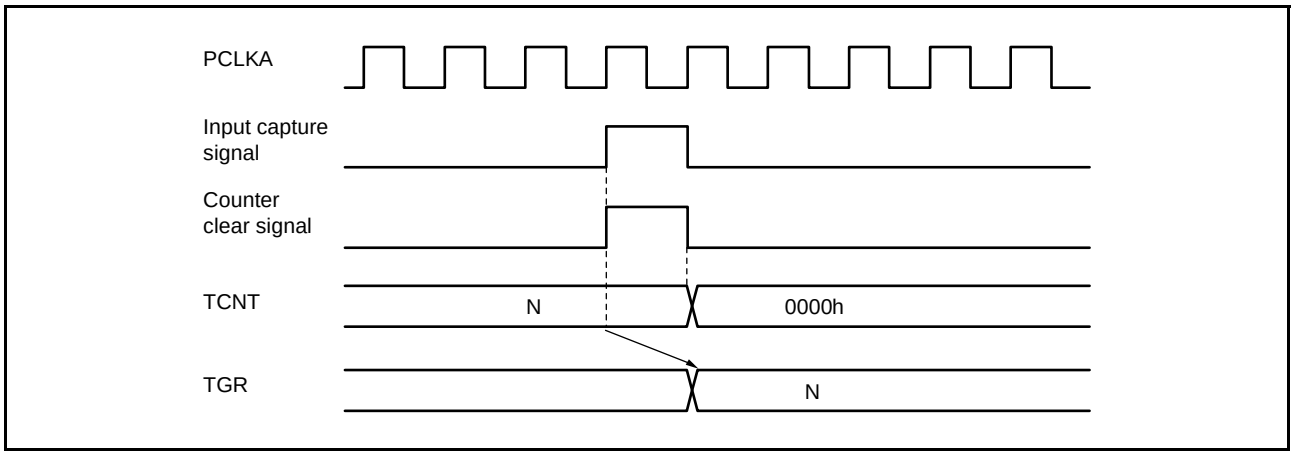


Figure 24.120 Counter Clear Timing (Input Capture) (MTU0 to MTU8)

(5) Buffer Operation Timing

Figure 24.121 to Figure 24.123 show the timing in buffer operation.

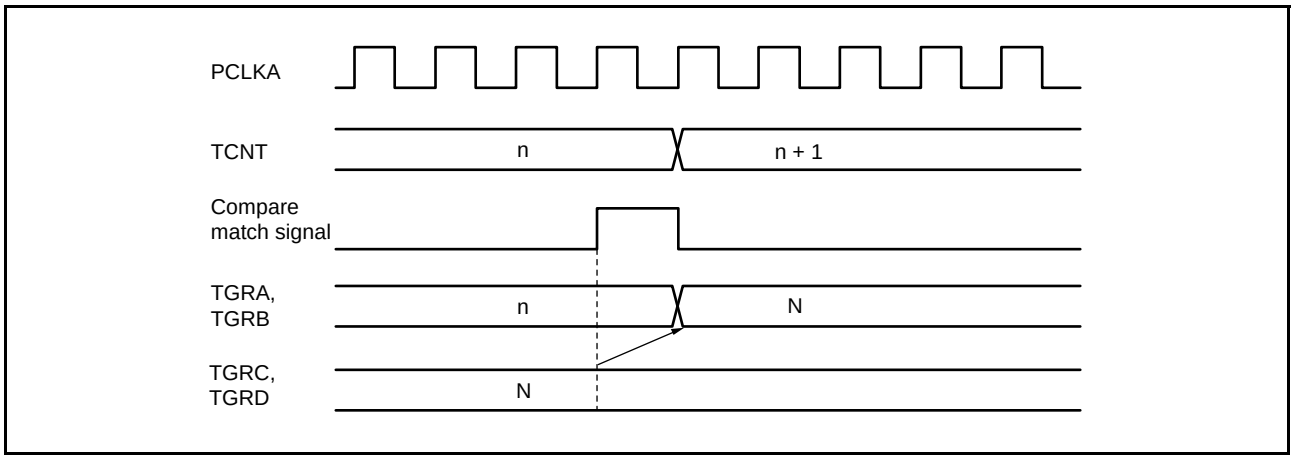


Figure 24.121 Buffer Operation Timing (Compare Match)

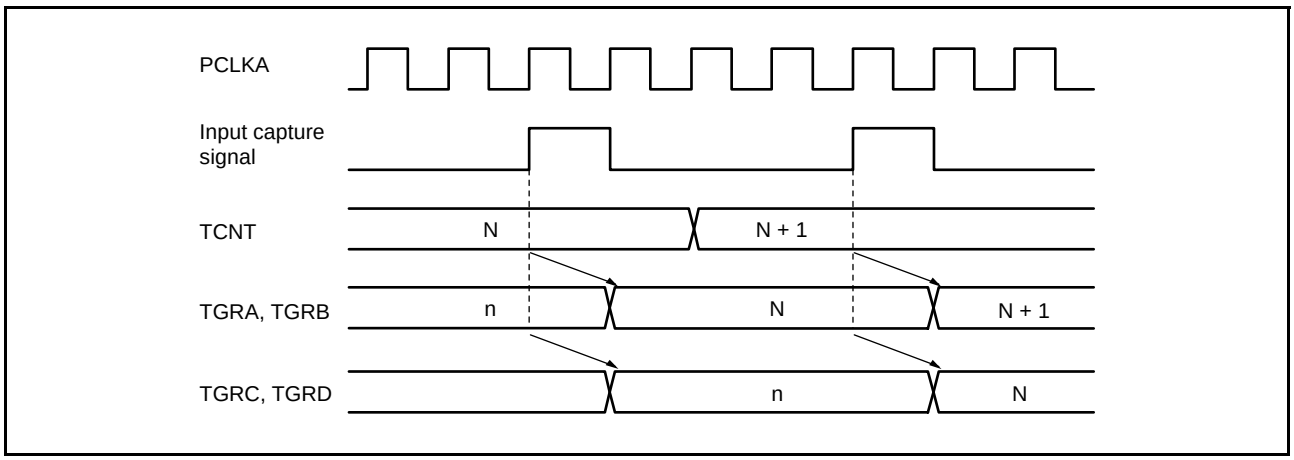


Figure 24.122 Buffer Operation Timing (Input Capture)

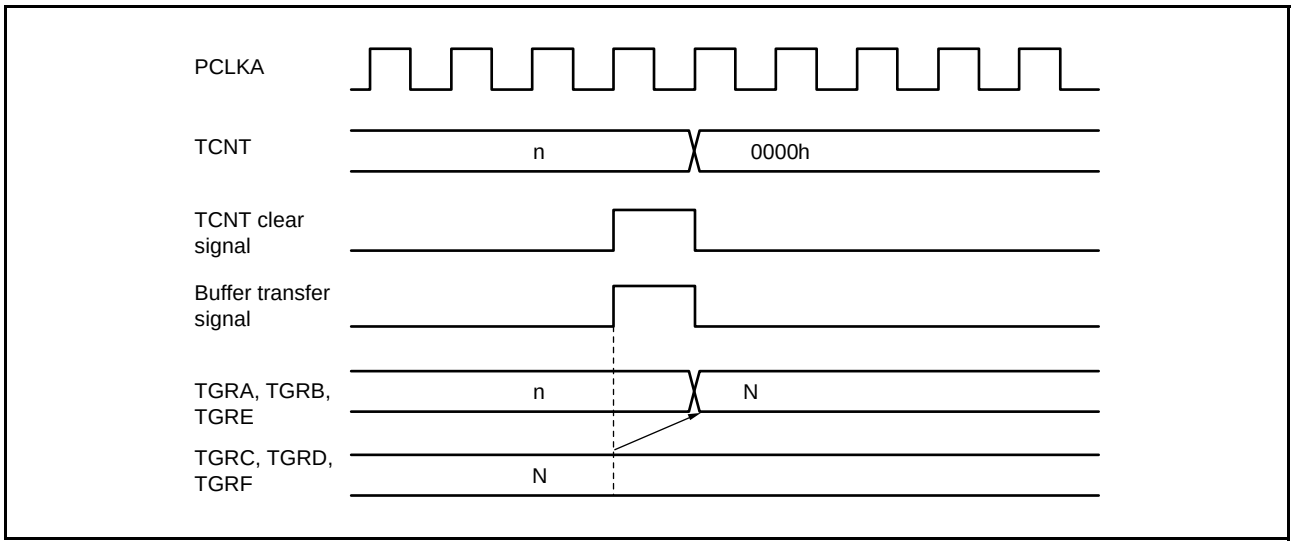


Figure 24.123 Buffer Operation Timing (When TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figure 24.124 to Figure 24.126 show the buffer transfer timing in complementary PWM mode.

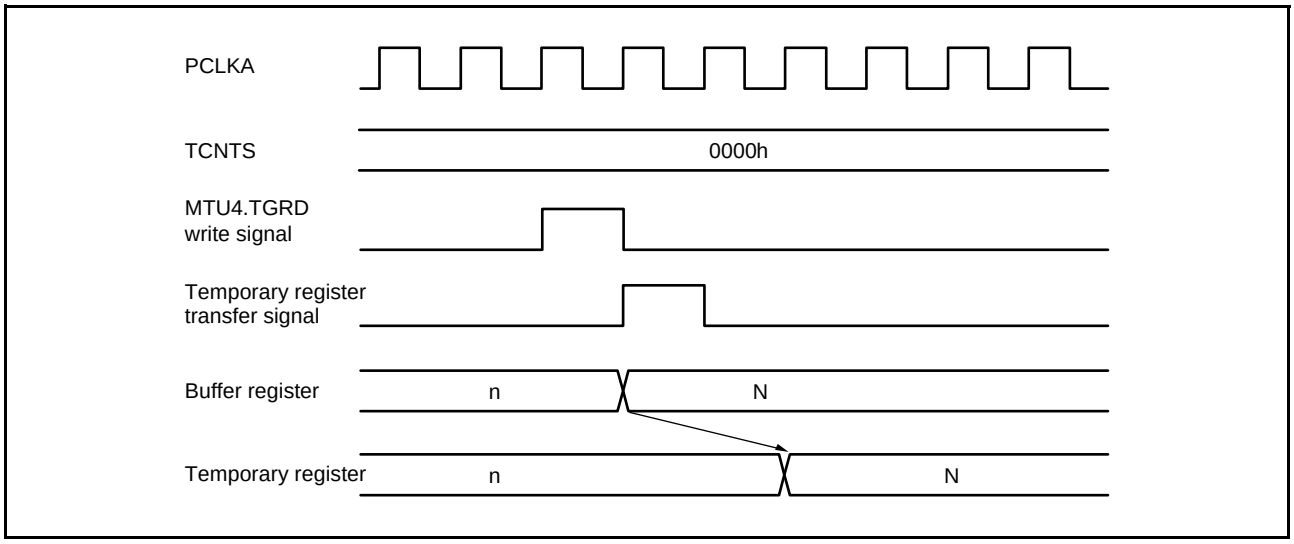


Figure 24.124 Transfer Timing from Buffer Register to Temporary Register (TCNTSA Stopped)

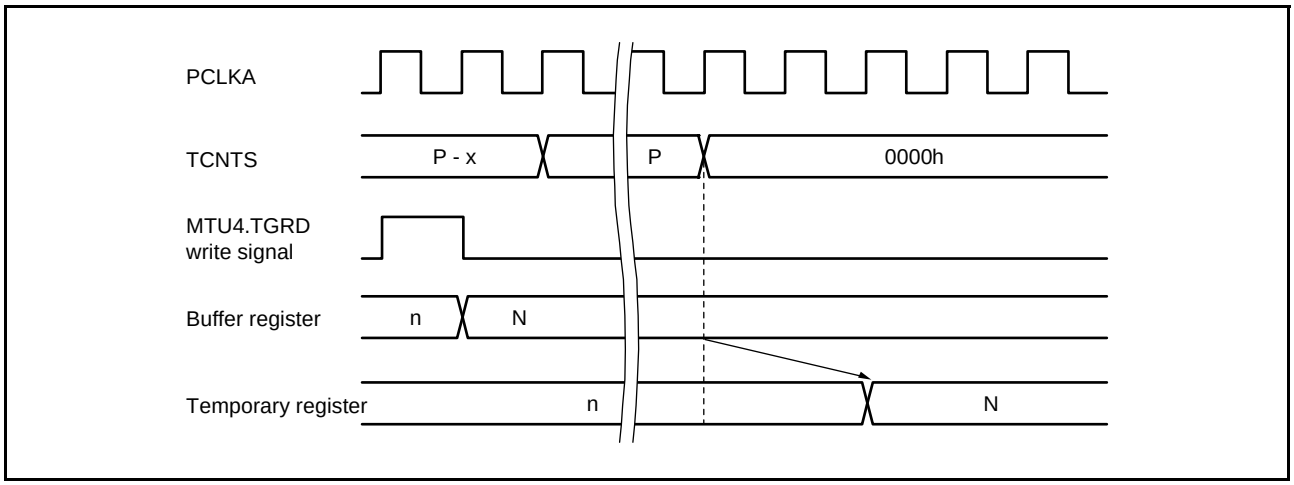


Figure 24.125 Transfer Timing from Buffer Register to Temporary Register (TCNTSA Operating)

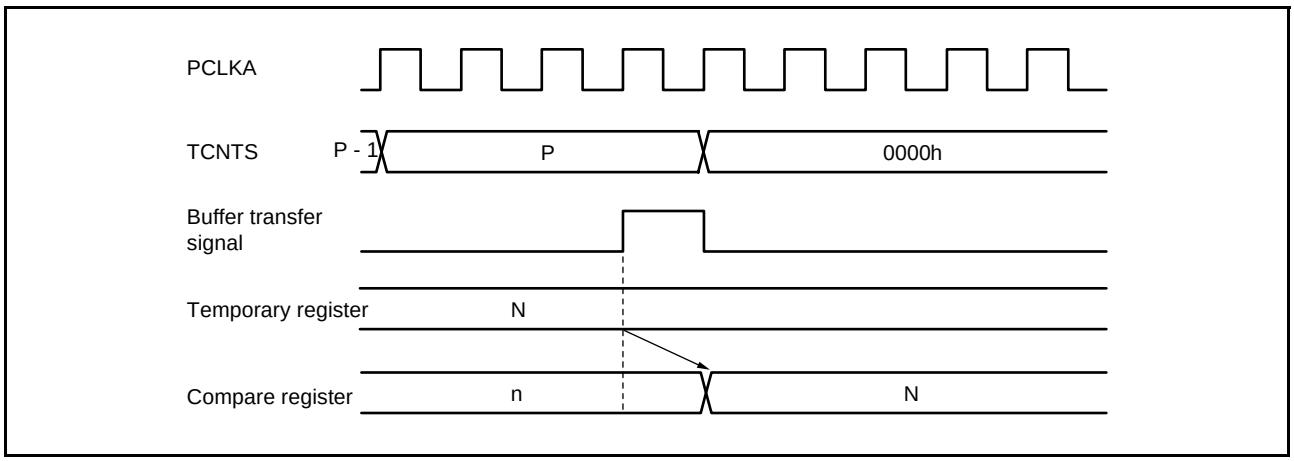


Figure 24.126 Transfer Timing from Temporary Register to Compare Register

24.5.2 Interrupt Signal Timing

(1) TGI Interrupt Timing by Compare Match

Figure 24.127 and Figure 24.128 show the TGI interrupt request signal timing when a compare match occurs.

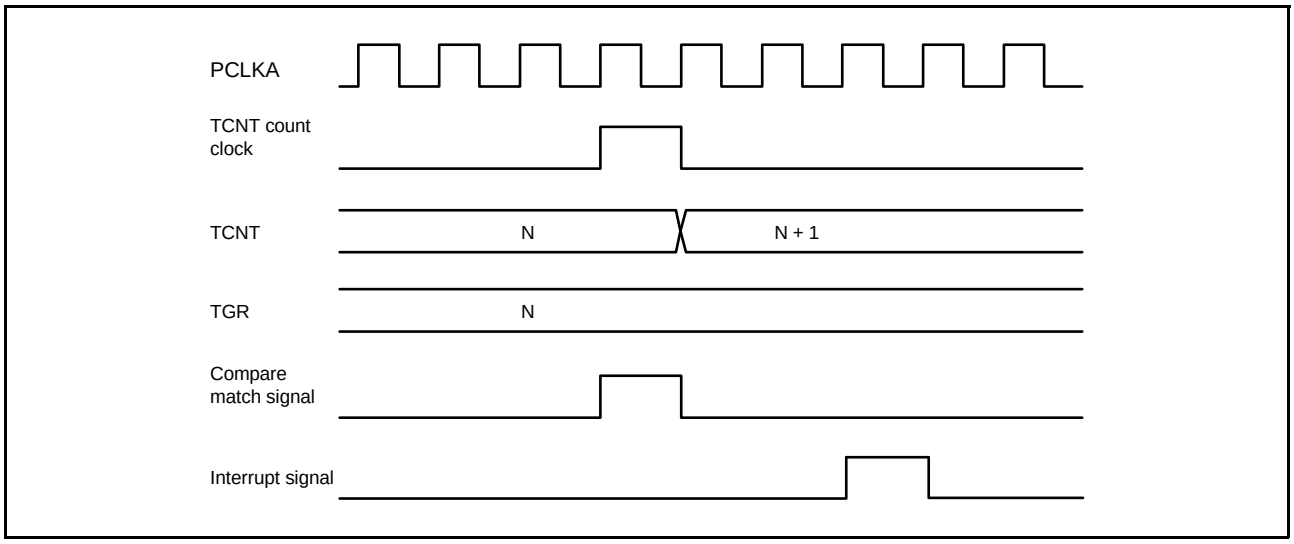


Figure 24.127 TGI Interrupt Timing (Compare Match) (MTU0 to MTU4 and MTU6 to MTU8)

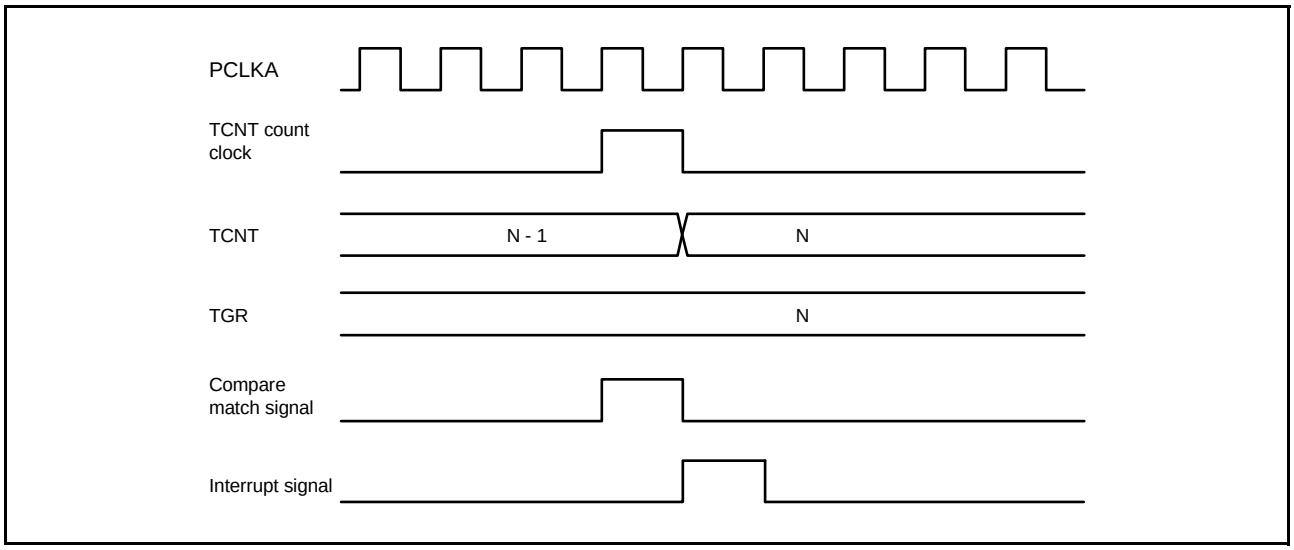


Figure 24.128 TGI Interrupt Timing (Compare Match) (MTU5)

(2) TGI Interrupt Timing by Input Capture

Figure 24.129 and Figure 24.130 show the TGI interrupt request signal timing when an input capture occurs.

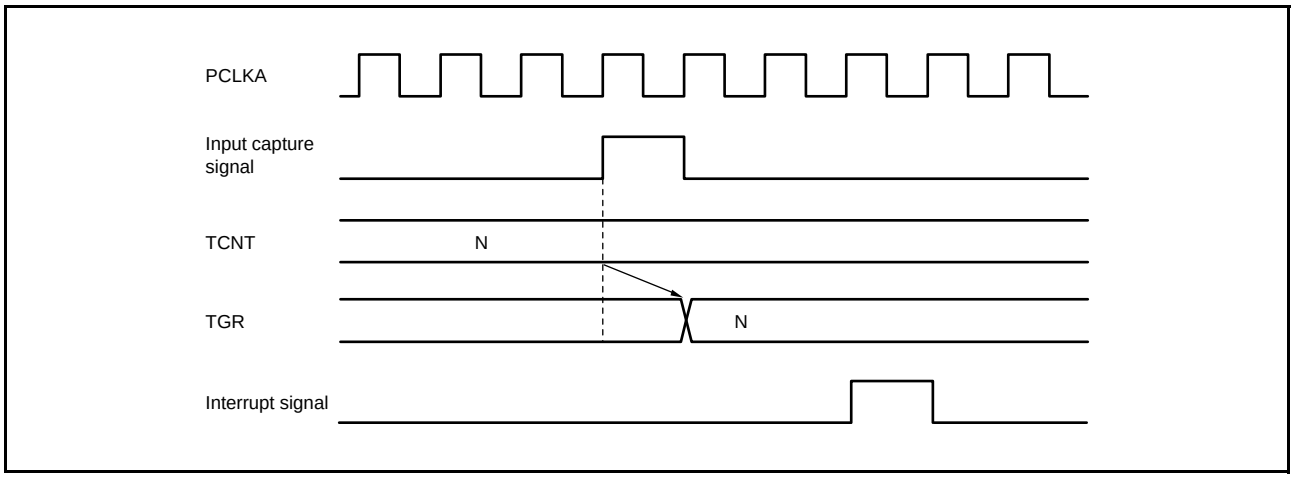


Figure 24.129 TGI Interrupt Timing (Input Capture) (MTU0 to MTU4 and MTU6 to MTU8)

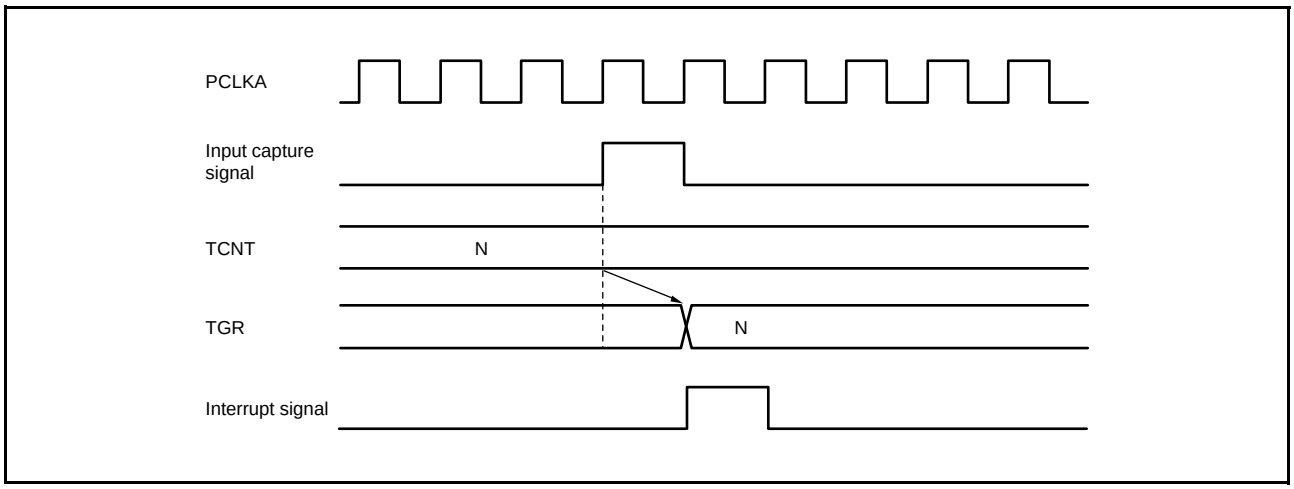


Figure 24.130 TGI Interrupt Timing (Input Capture) (MTU5)

(3) TCIV and TCIU Interrupt Timing

Figure 24.131 shows the TCIV interrupt request signal timing when an overflow is generated.
Figure 24.132 shows the TCIU interrupt request signal timing when an underflow is generated.

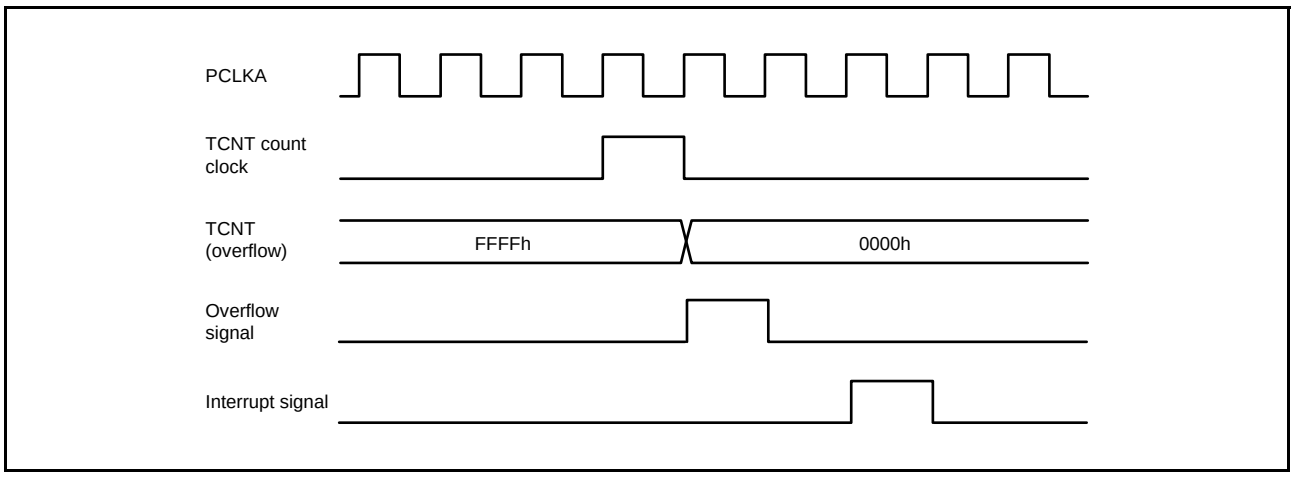


Figure 24.131 TCIV Interrupt Timing

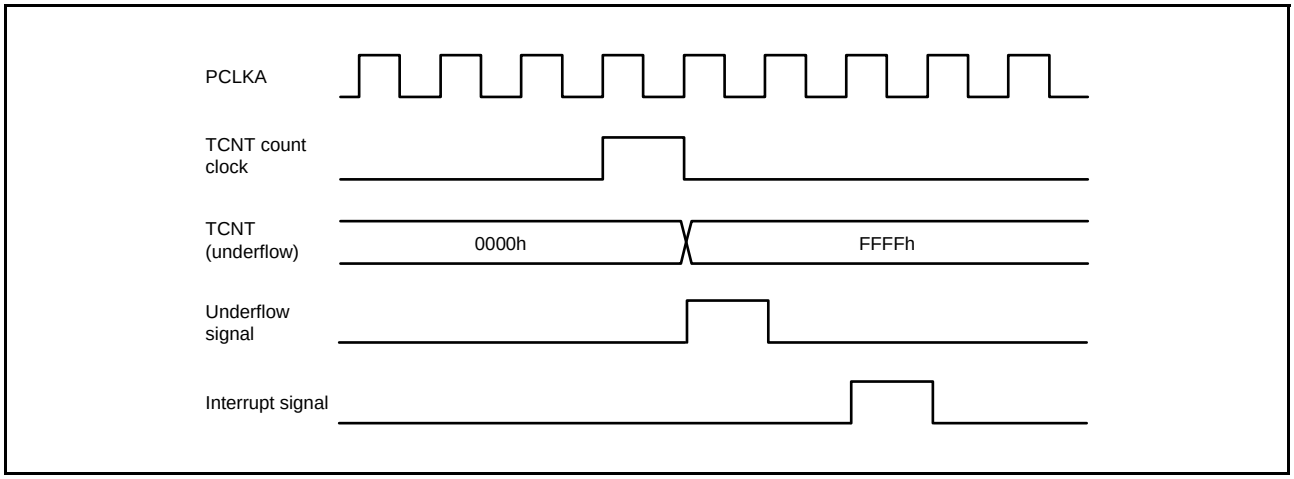


Figure 24.132 TCIU Interrupt Timing

24.6 Usage Notes

24.6.1 Module Stop Function Setting

MTU operation can be disabled or enabled using the module stop control register. MTU operation is stopped with the initial setting. Register access is enabled by releasing the module clock stop state. For details, refer to section 11, Low Power Consumption.

24.6.2 Count Clock Restrictions

The count clock source pulse width must be at least three PCLKA clocks for single-edge detection, and at least five PCLKA clocks for both-edge detection. The MTU will not operate properly at narrower pulse widths.
In phase counting mode, the phase difference and overlap between the two input clocks must be at least three PCLKA clocks, and the pulse width must be at least five PCLKA clocks. Figure 24.133 shows the input clock conditions in phase counting mode.

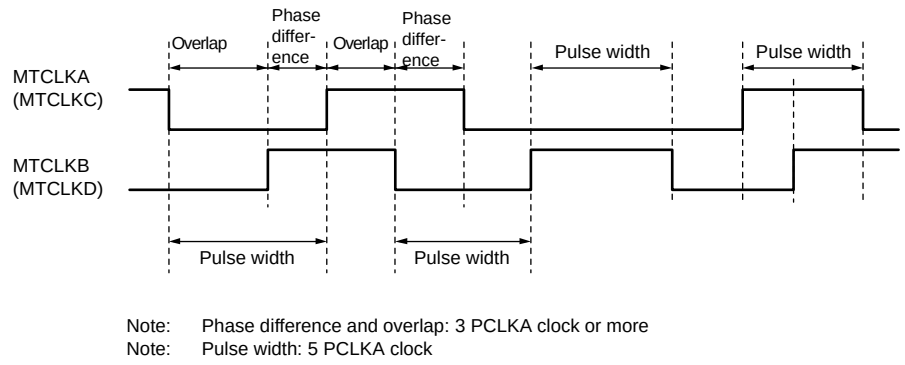


Figure 24.133 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

24.6.3 Note on Cycle Setting

When counter clearing on compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which TCNT updates the matched count value). Consequently, the actual counter frequency is given by the following formula:

- MTU0 to MTU4 and MTU6 to MTU8

$$f = \frac{CNTCLK}{N + 1}$$

- MTU5

$$f = \frac{CNTCLK}{N}$$

f: Counter frequency
CNTCLK: The count clock frequency set by TCR.TPSC[2:0] and TCR2.TPSC2[2:0]
N: TGR setting

24.6.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in the TCNT write cycle, TCNT clearing takes precedence and TCNT write operation is not performed.
Figure 24.134 shows the timing in this case.

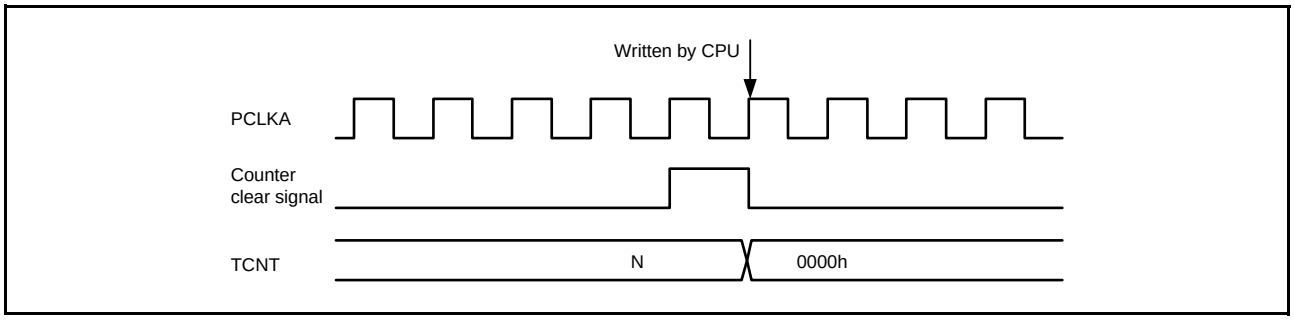


Figure 24.134 Contention between TCNT Write and Clear Operations

24.6.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in a TCNT write cycle, TCNT write operation takes precedence and TCNT is not incremented.
Figure 24.135 shows the timing in this case.

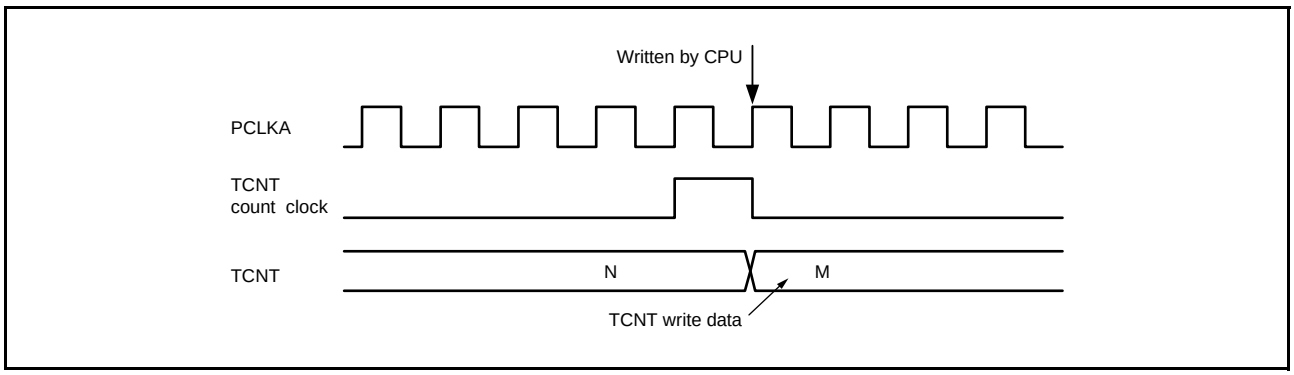


Figure 24.135 Contention between TCNT Write and Increment Operations

24.6.6 Contention between TGR Write Operation and Compare Match

If a compare match occurs in a TGR write cycle, TGR write operation is executed and the compare match signal is also generated.
Figure 24.136 shows the timing in this case.

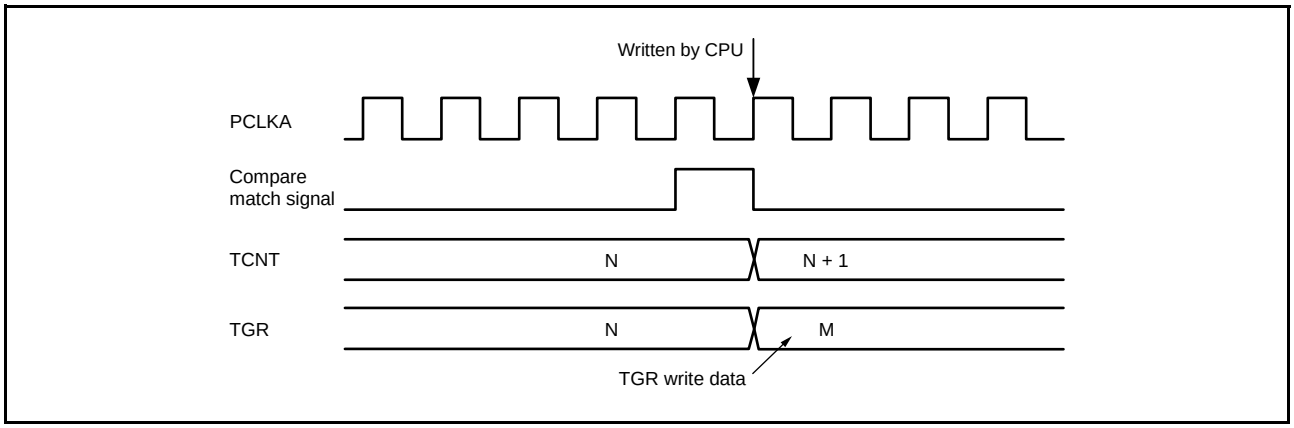


Figure 24.136 Contention between TGR Write Operation and Compare Match

24.6.7 Contention between Buffer Register Write Operation and Compare Match

If a compare match occurs in the T2 state in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.
Figure 24.137 shows the timing in this case.

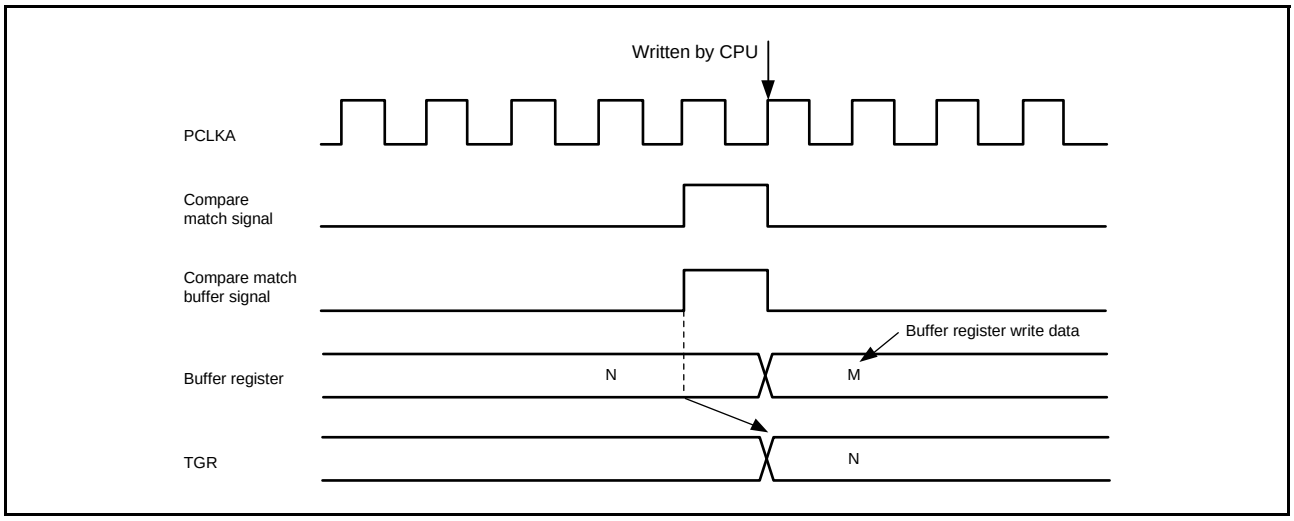


Figure 24.137 Contention between Buffer Register Write Operation and Compare Match

24.6.8 Contention between Buffer Register Write and TCNT Clear Operations

When the buffer transfer timing is set at the TCNT clear timing by the timer buffer transfer mode register (TBTM), if TCNT clearing occurs in the TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.
Figure 24.138 shows the timing in this case.

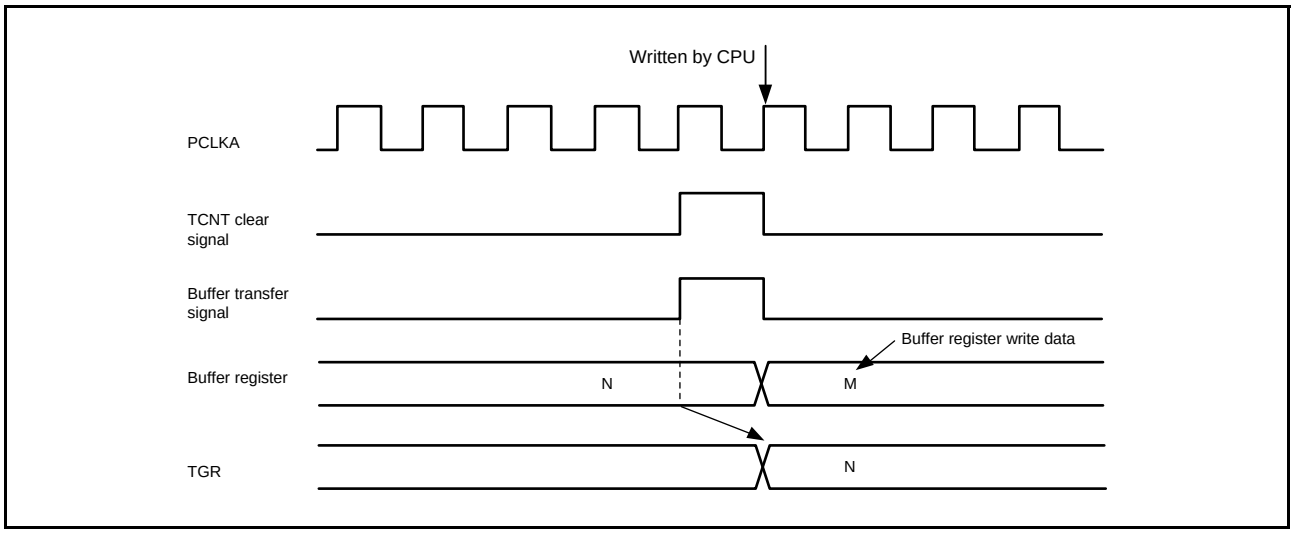


Figure 24.138 Contention between Buffer Register Write and TCNT Clear Operations

24.6.9 Contention between TGR Read Operation and Input Capture

If an input capture signal is generated in a TGR read cycle, the data before input capture transfer is read.
Figure 24.139 shows the timing in this case.

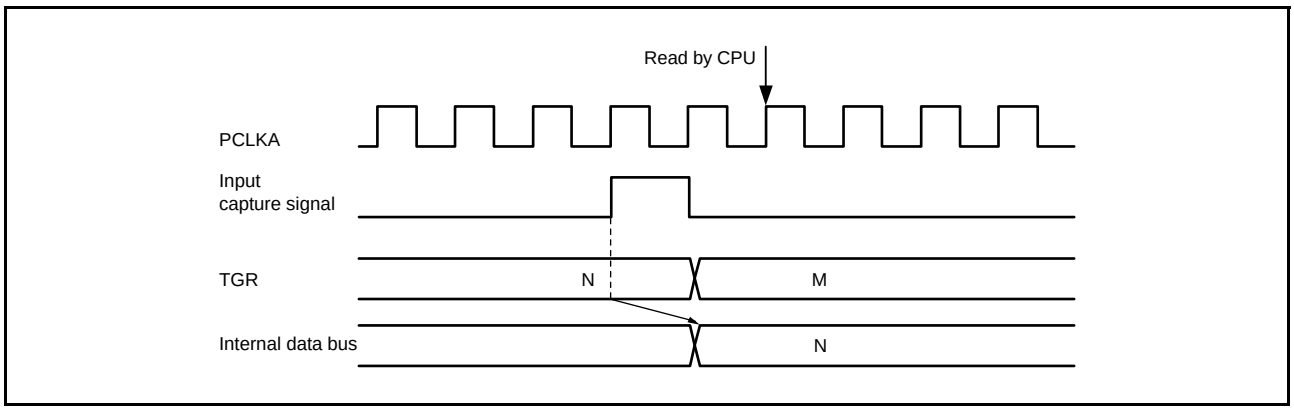


Figure 24.139 Contention between TGR Read Operation and Input Capture (MTU0 to MTU8)

24.6.10 Contention between TGR Write Operation and Input Capture

If an input capture signal is generated in the TGR write cycle, the input capture operation takes precedence and the TGR write operation is not performed in MTU0 to MTU4 and MTU6 to MTU8. In MTU5, the TGR write operation is performed and the input capture signal is generated.
Figure 24.140 and Figure 24.141 show the timing in this case.

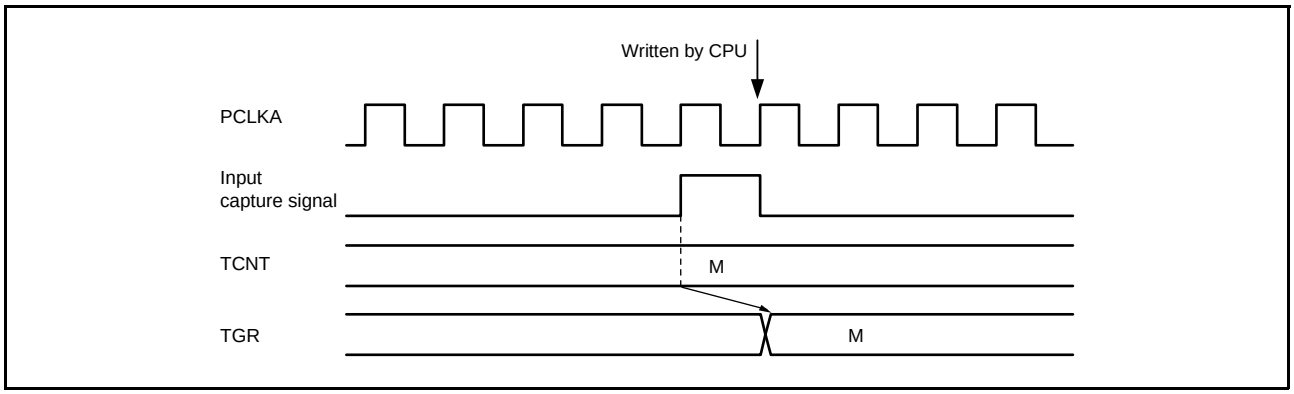


Figure 24.140 Contention between TGR Write Operation and Input Capture (MTU0 to MTU4 and MTU6 to MTU8)

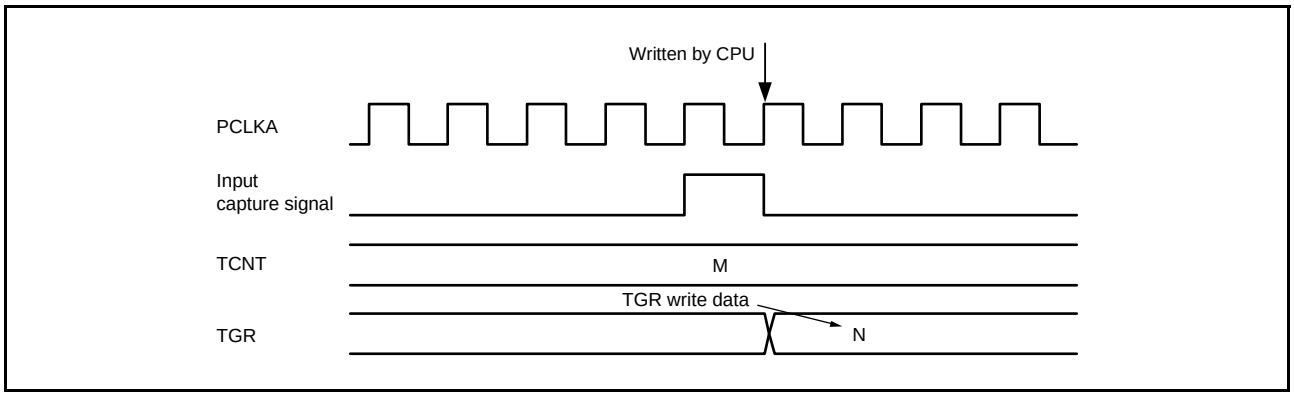


Figure 24.141 Contention between TGR Write Operation and Input Capture (MTU5)

24.6.11 Contention between Buffer Register Write Operation and Input Capture

If an input capture signal is generated in the buffer register write cycle, the buffer operation takes precedence and the buffer register write operation is not performed.
Figure 24.142 shows the timing in this case.

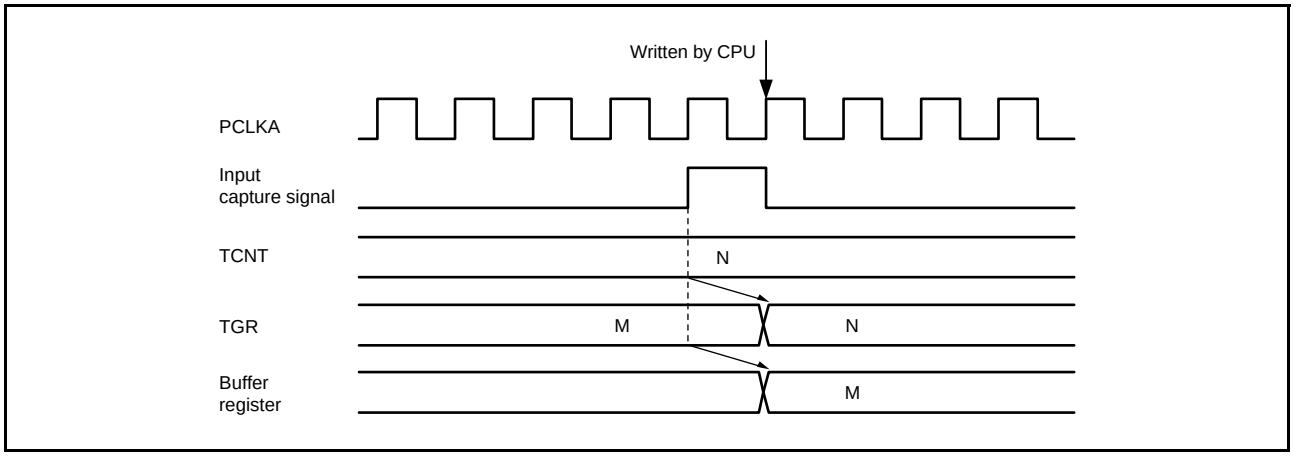


Figure 24.142 Contention between Buffer Register Write Operation and Input Capture

24.6.12 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

With timer counters MTU1.TCNT and MTU2.TCNT in a cascade, when a contention occurs between MTU1.TCNT counting (an MTU2.TCNT overflow/underflow) and the MTU2.TCNT write cycle, the MTU2.TCNT write operation is performed and the MTU1.TCNT count signal is disabled. In this case, if MTU1.TGRA works as a compare match register and there is a match between the MTU1.TGRA and MTU1.TCNT values, a compare match signal is issued. Furthermore, when the MTU1.TCNT count clock is selected as the input capture source of MTU0, MTU0.TGRA to MTU0.TGRD work in input capture mode. In addition, when the MTU0.TGRC compare match/input capture is selected as the input capture source of MTU1.TGRB, MTU1.TGRB works in input capture mode.

Figure 24.143 shows the timing in this case.

When setting the TCNT clearing function in cascaded operation, be sure to synchronize MTU1 and MTU2.

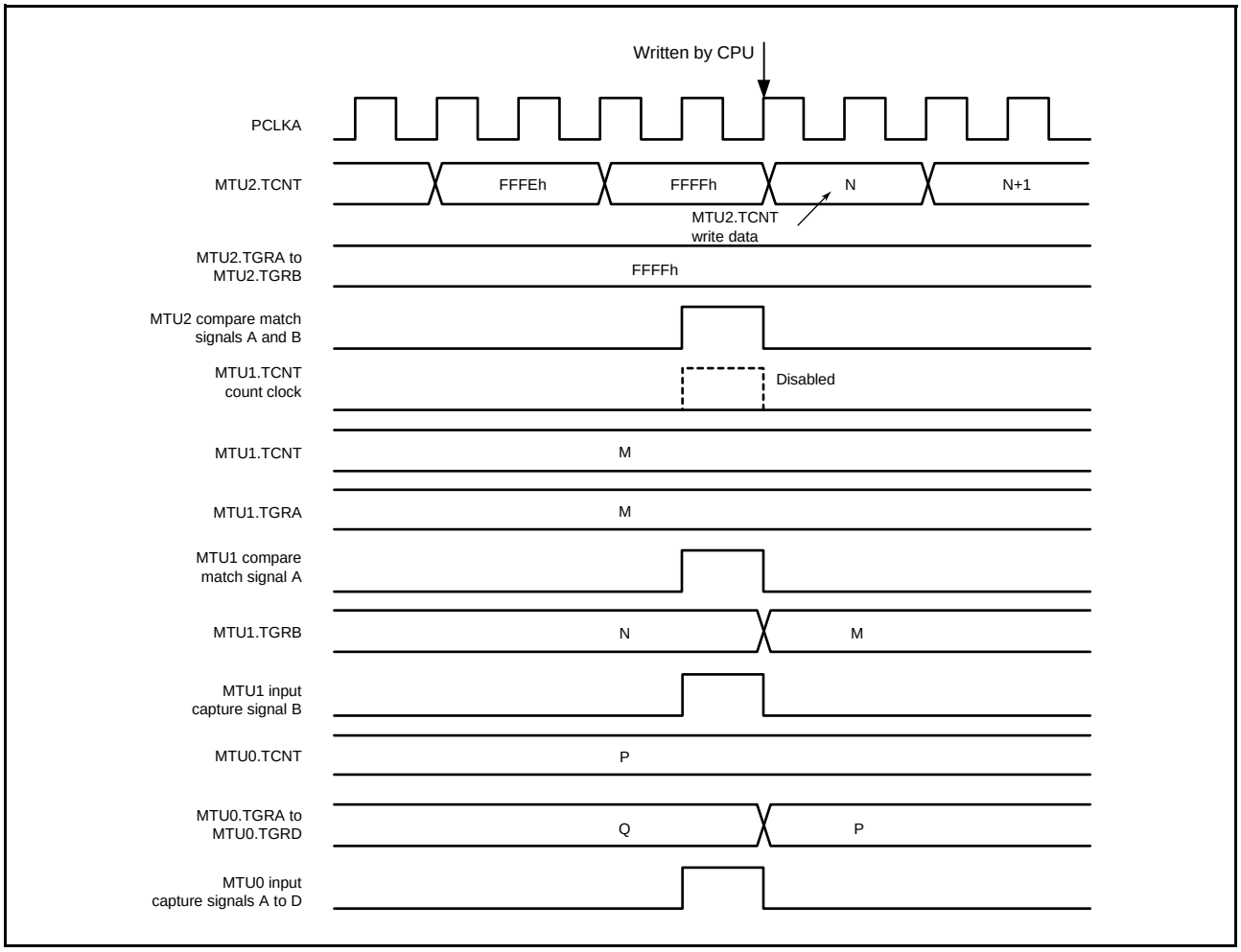


Figure 24.143 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

24.6.13 Counter Value When Count Operation is Stopped in Complementary PWM Mode

When counting operation in MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) is stopped in complementary PWM mode, the MTU3.TCNT (MTU6.TCNT) value is set to the timer dead time register (TDDRA (TDDRB)) value and MTU4.TCNT (MTU7.TCNT) is set to 0000h.
When operation is restarted in complementary PWM mode, counting begins automatically from the initial setting state. Figure 24.144 shows this operation.
When counting begins in another operating mode, be sure to make initial settings in MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT).

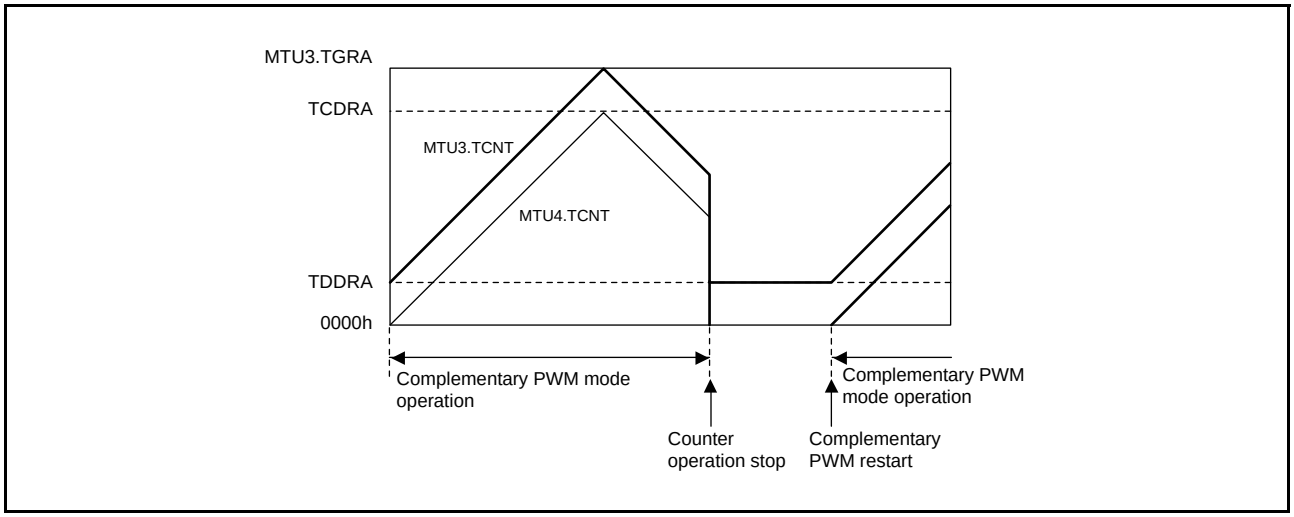


Figure 24.144 Counter Value When Stopped in Complementary PWM Mode

24.6.14 Buffer Operation Setting in Complementary PWM Mode

When modifying the PWM cycle set register (MTU3.TGRA or MTU6.TGRA), timer cycle data register (TCDRA or TCDRB), and duty set registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB (MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB)) in complementary PWM mode, be sure to use buffer operation. In addition, set the BFA and BFB bits in MTU4.TMDR1 (MTU7.TMDR1) to 0. The MTIOC4C (MTIOC7C) pin cannot output waveforms if the BFA bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1. Likewise, the MTIOC4D (MTIOC7D) pin cannot output waveforms if the BFB bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1.
In complementary PWM mode, buffer operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the settings in bits BFA and BFB of MTU3.TMDR1 (MTU6.TMDR1). When the BFA bit in MTU3.TMDR1 (MTU6.TMDR1) is set to 1, MTU3.TGRC (MTU6.TGRC) functions as a buffer register for MTU3.TGRA (MTU6.TGRA). At the same time, MTU4.TGRC (MTU7.TGRC) functions as a buffer register for MTU4.TGRA (MTU7.TGRA), and TCBRA (TCBRB) functions as a buffer register for TCDRA (TCDRB).

24.6.15 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode

When setting buffer operation in reset-synchronized PWM mode, set the BFA and BFB bits in MTU4.TMDR1 (MTU7.TMDR1) to 0. The MTIOC4C (MTIOC7C) pin cannot output waveforms if the BFA bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1. Likewise, the MTIOC4D (MTIOC7D) pin cannot output waveforms if the BFB bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1.

In reset-synchronized PWM mode, buffer operation in MTU3 and MTU4 (or MTU 6 and MTU7) depends on the settings in the BFA and BFB bits of MTU3.TMDR1 (MTU6.TMDR1). For example, if the BFA bit in MTU3.TMDR1 (MTU6.TMDR1) is set to 1, MTU3.TGRC (MTU6.TGRC) functions as a buffer register for MTU3.TGRA (MTU6.TGRA). At the same time, MTU4.TGRC (MTU7.TGRC) functions as a buffer register for MTU4.TGRA (MTU7.TGRA).

While the MTU3.TGRC (MTU6.TGRC) and MTU3.TGRD (MTU6.TGRD) are operating as buffer registers, a TGImn interrupt (m = C, D; n = 3, 4 or 6, 7) is not generated.

Figure 24.145 shows an example of MTU3.TGR (MTU6.TGR), MTU4.TGR (MTU7.TGR), MTIOC3 (MTIOC6), and MTIOC4 (MTIOC7) operation with the BFA and BFB bits in MTU3.TMDR1 (MTU6.TMDR1) set to 1 and the BFA and BFB bits in MTU4.TMDR1 (MTU7.TMDR1) set to 0.

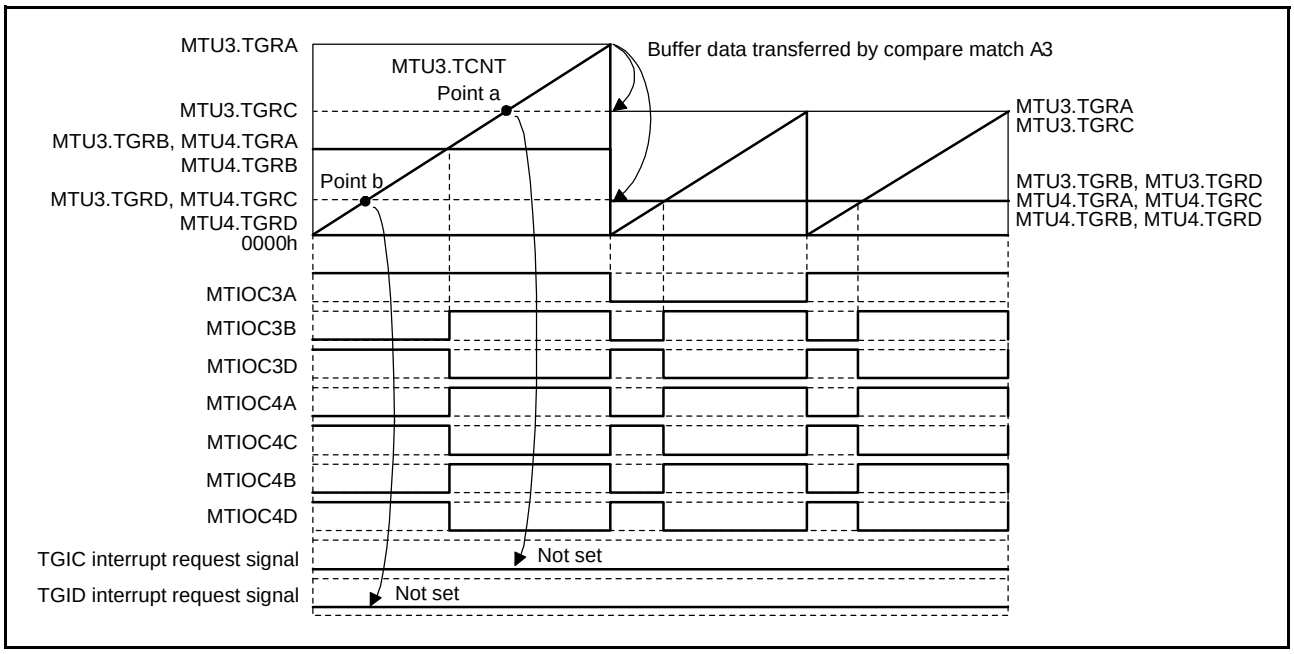


Figure 24.145 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode

24.6.16 Overflow in Reset-Synchronized PWM Mode

After reset-synchronized PWM mode is selected, MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) start counting when the CST3 (CST6) bit of TSTRA (TSTRB) is set to 1. In this state, the MTU4.TCNT (MTU7.TCNT) count clock source and count edge are determined by the MTU3.TCR (MTU6.TCR) setting.

In reset-synchronized PWM mode, with cycle register MTU3.TGRA (MTU6.TGRA) set to FFFFh and the MTU3.TGRA (MTU6.TGRA) compare match selected as the counter clearing source, MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) count up to FFFFh, then a compare match occurs with MTU3.TGRA (MTU6.TGRA), and MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) are both cleared. In this case, a TCIVn interrupt (n = 3, 4 or 6, 7) is not generated.

Figure 24.146 shows an example of operation in reset-synchronized PWM mode with cycle register MTU3.TGRA (MTU6.TGRA) set to FFFFh and the MTU3.TGRA (MTU6.TGRA) compare match specified for the counter clearing source.

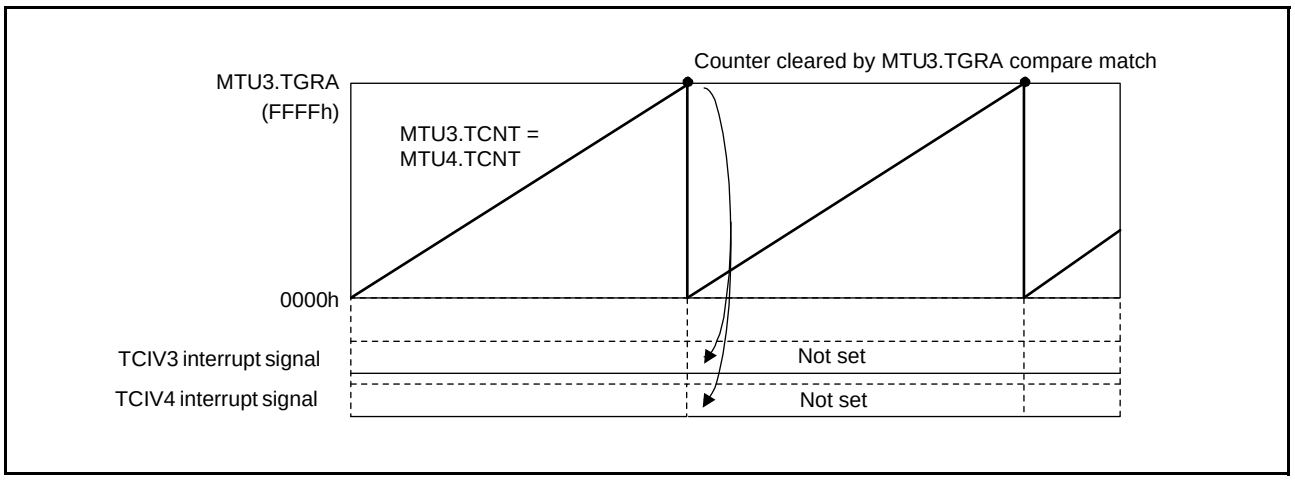


Figure 24.146 Overflow in Reset-Synchronized PWM Mode

24.6.17 Contention between Overflow/Underflow and Counter Clearing

If an overflow/underflow and counter clearing occur simultaneously, a TCIVn interrupt (n = 0 to 4, 6 to 8) nor a TCIUn interrupt (n = 1, 2) is not generated and TCNT clearing takes precedence.
Figure 24.147 shows the operation timing when a TGR compare match is specified as the clearing source and TGR is set to FFFFh.

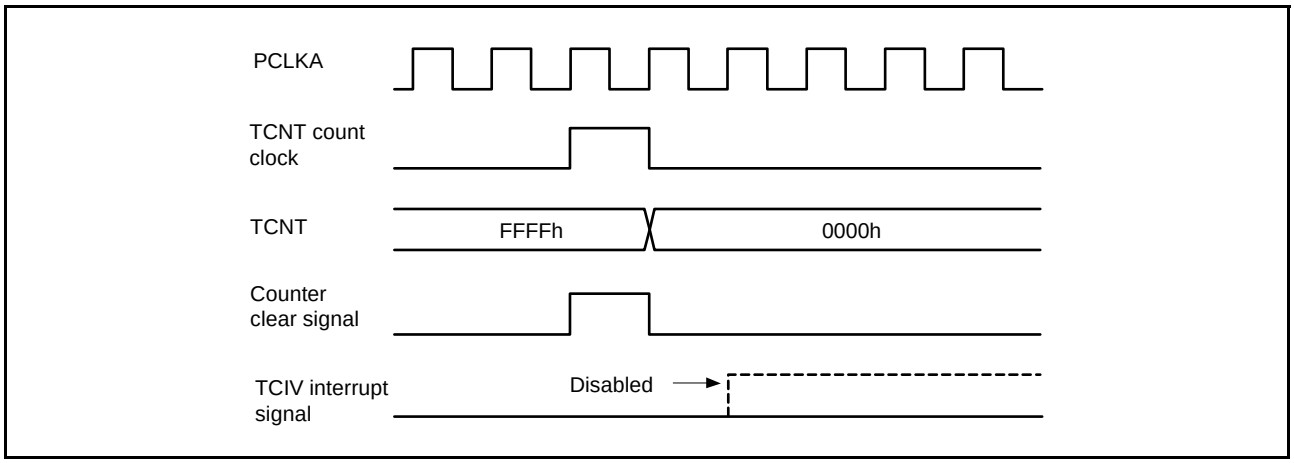


Figure 24.147 Contention between Overflow and Counter Clearing

24.6.18 Contention between TCNT Write Operation and Overflow/Underflow

If TCNT counts up or down in a TCNT write cycle and an overflow or an underflow occurs, the TCNT write operation takes precedence. A TCIVn interrupt (n = 0 to 4, 6 to 8) nor a TCIUn interrupt (n = 1, 2) is not generated.
Figure 24.148 shows the operation timing when there is contention between TCNT write operation and overflow.

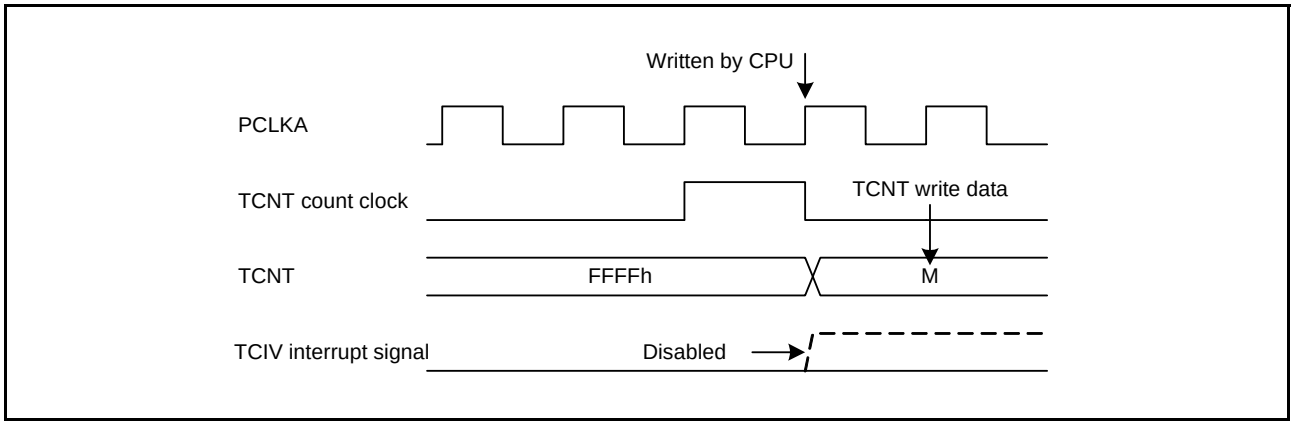


Figure 24.148 Contention between TCNT Write Operation and Overflow

24.6.19 Note on Transition from Normal Mode or PWM Mode 1 to Reset-Synchronized PWM Mode

When making a transition from normal mode or PWM mode 1 to reset-synchronized PWM mode in MTU3 and MTU4 (or MTU6 and MTU7), if the counter is stopped while the output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, and MTIOC7D) are held at a high level and then operation is started after a transition to reset-synchronized PWM mode, the initial pin output will not be correct.

When making a transition from normal mode to reset-synchronized PWM mode, write 11h to MTU3.TIORH, MTU3.TIORL, MTU4.TIORH, and MTU4.TIORL (MTU6.TIORH, MTU6.TIORL, MTU7.TIORH, and MTU7.TIORL) to initialize the output pin state to a low level, then set the registers to the initial value (00h) before making the mode transition.

When making a transition from PWM mode 1 to reset-synchronized PWM mode, switch to normal mode, initialize the output pin state to a low level, and then set the registers to the initial value (00h) before making the transition to reset-synchronized PWM mode.

24.6.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode

When MTU3 and MTU4 (or MTU6 and MTU7) are in complementary PWM mode or reset-synchronized PWM mode, the PWM waveform output level is determined by the OLSP and OLSN bits in the timer output control register (TOCR1A or TOCR1B). In complementary PWM mode or reset-synchronized PWM mode, TIOR should be set to 00h. The output level in negative phase when the TDER.TDER bit is set to 0 in complementary PWM mode (the dead time is not generated) does not depend on the setting of the TOCR1.OLSN bit. It is equivalent to the inverted level of positive phase output based on the setting of the TOCR1.OLSP bit.

24.6.21 Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection

When timer counters 1 and 2 (MTU1.TCNT and MTU2.TCNT) operate as a 32-bit counter in cascade connection, the cascaded counter value cannot be captured successfully in some cases even if input-capture input is simultaneously done to MTIOC1A and MTIOC2A or to MTIOC1B and MTIOC2B. This is because the input timing of MTIOC1A and MTIOC2A or to MTIOC1B and MTIOC2B may not be the same when external input-capture signals input into MTU1.TCNT and MTU2.TCNT are taken in synchronization with the internal clock.

For example, MTU1.TCNT (the counter for upper 16 bits) does not capture the count-up value by an overflow from MTU2.TCNT (the counter for lower 16 bits) but captures the count value before the up-counting. In this case, the values of MTU1.TCNT = FFF1h and MTU2.TCNT = 0000h should be transferred to MTU1.TGRA and MTU2.TGRA or to MTU1.TGRB and MTU2.TGRB, but the values of MTU1.TCNT = FFF0h and MTU2.TCNT = 0000h are erroneously transferred.

The MTU has a function that allows simultaneous capture of MTU1.TCNT and MTU2.TCNT with a single input capture input. This function can be used to read the 32-bit counter such that MTU1.TCNT and MTU2.TCNT are captured at the same time. For details, refer to section 24.2.11, Timer Input Capture Control Register (TICCR).

24.6.22 Interrupt Skipping Function 2

When interrupt skipping function 2 is in use and the difference between the values in MTU4.TADCORA and MTU4.TADCORB is small, correct counting of the number skipped may not be possible, in which case requests for A/D conversion will not be generated with the expected timing. The conditions listed below thus apply to these settings.
For MTU6 and MTU7, the same conditions apply to the settings of MTU7.TADCORA and MTU7.TADCORB.

(1) When the number skipped is zero for skipping function 2

- The difference between the values in MTU4.TADCORA and MTU4.TADCORB must be at least four.
- The interval of comparison for MTU4.TADCORA must be at least four cycles of PCLKA clock (the updated value of MTU4.TADCORA is set to the previous value plus or minus at least four).
- The interval of comparison for MTU4.TADCORB must be at least four cycles of PCLKA clock (the updated value of MTU4.TADCORB is set to the previous value plus or minus at least four).

(2) When the number skipped is one or more for skipping function 2

- The difference between the values in MTU4.TADCORA and MTU4.TADCORB must be at least two.
- The interval of comparison for MTU4.TADCORB must be at least two cycles of PCLKA (the updated value of MTU4.TADCORB is set to the previous value plus or minus at least two).

24.6.23 Notes When Complementary PWM Mode Output Protection Function is Not Used

The complementary PWM mode output protection function is initially enabled. For details, refer to section 25, Port Output Enable 3 (POE3a).

24.6.24 Notes Regarding Timer Counter (MTU5.TCNT) and Timer General Register (MTU5.TGR)

Do not set an MTU5.TGRj (j = U, V, W) bit to the value of the corresponding MTU5.TCNTj (j = U, V, W) plus one while counting by the MTU5.TCNTj (j = U, V, W) register is stopped. If an MTU5.TGRj (j = U, V, W) bit is set to the value of the corresponding MTU5.TCNTj (j = U, V, W) plus one while counting by the MTU5.TCNTj (j = U, V, W) is stopped, a compare-match will be generated even though counting is stopped.
In this case, if the compare match enable bit (MTU5.TIER.TGIE5j (j = U, V, W) bit is set to 1 (enabling interrupts), a compare-match interrupt will also be generated. If the value of the timer compare match clear register is 1 (enabled), the timer is automatically cleared to 0000h when the compare-match is generated, regardless of whether interrupts from the MTU5.TCNTj (j = U, V, W) are enabled or disabled.

24.6.25 Notes to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode

If control of the output waveform is enabled (TWCRA.WRE bit = 1 or TWCRB.WRE bit = 1) at the time of synchronous counter clearing in complementary PWM mode, satisfaction of either condition 1 or 2 below has the following effects.

- Dead time on the PWM output pins is shortened (or disappears).
- The active level is output on the negative phase PWM output pins beyond the period for active-level output.

Condition 1: In portion (10) of the initial output inhibition period in Figure 24.149, synchronous clearing occurs within the dead-time period for PWM output.

Condition 2: In portions (10) and (11) of the initial output inhibition period in Figure 24.150, synchronous clearing occurs when any condition from among MTU3.TGRB (MTU6.TGRB) ≤ TDDR (TDDRB), MTU4.TGRA (MTU7.TGRA) ≤ TDDR (TDDRB), or MTU4.TGRB (MTU7.TGRB) ≤ TDDR (TDDRB) is satisfied.

The following method avoids the above phenomena.

Ensure that synchronous clearing proceeds with the value of each comparison register (MTU3.TGRB (MTU6.TGRB), MTU4.TGRA (MTU7.TGRA), and MTU4.TGRB (MTU7.TGRB) ≤ TDDR (TDDRB)) set to at least double the value of the TDDRA register (TDDRB register).

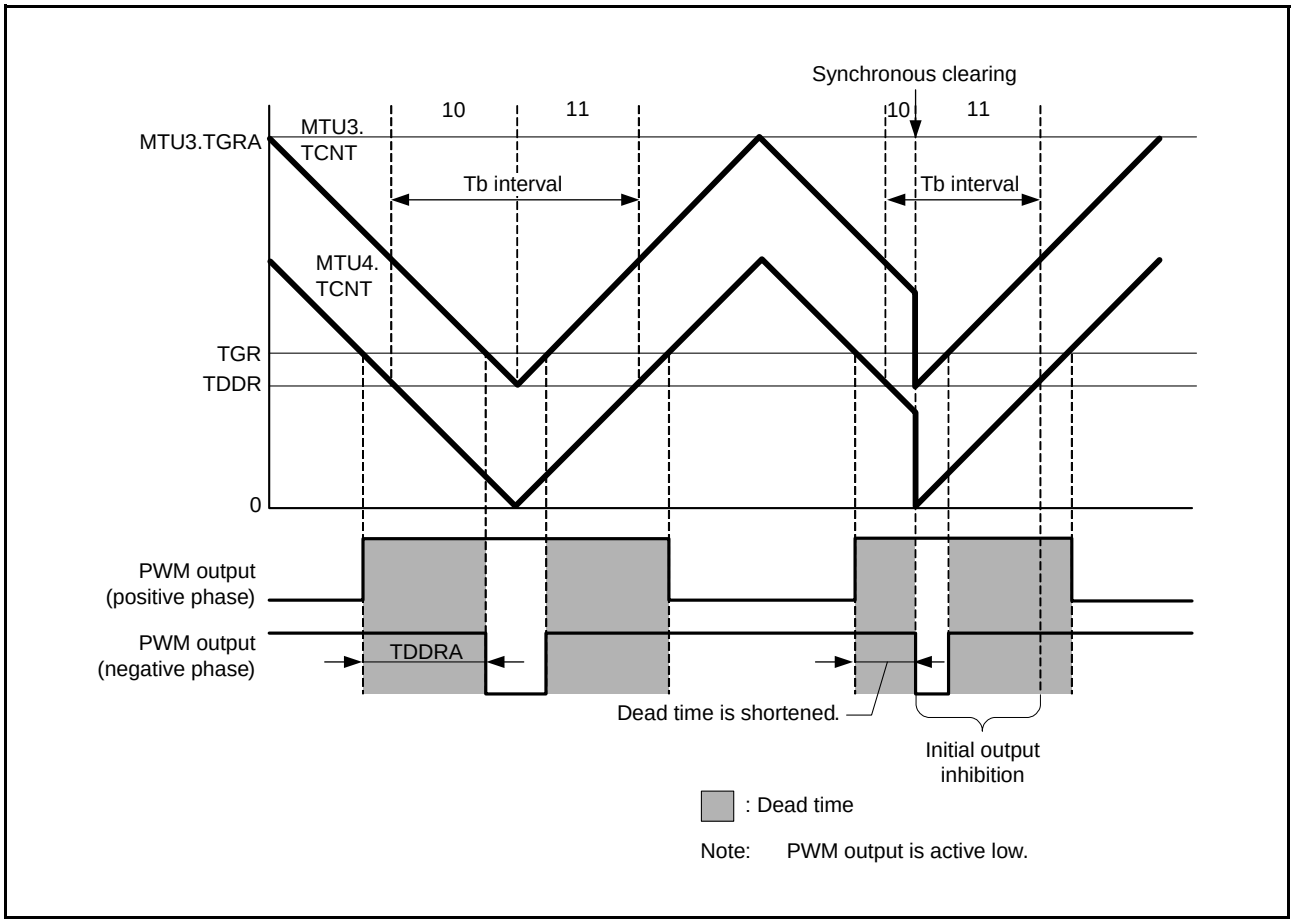


Figure 24.149 Example of Synchronous Clearing (When Condition 1 Applies)

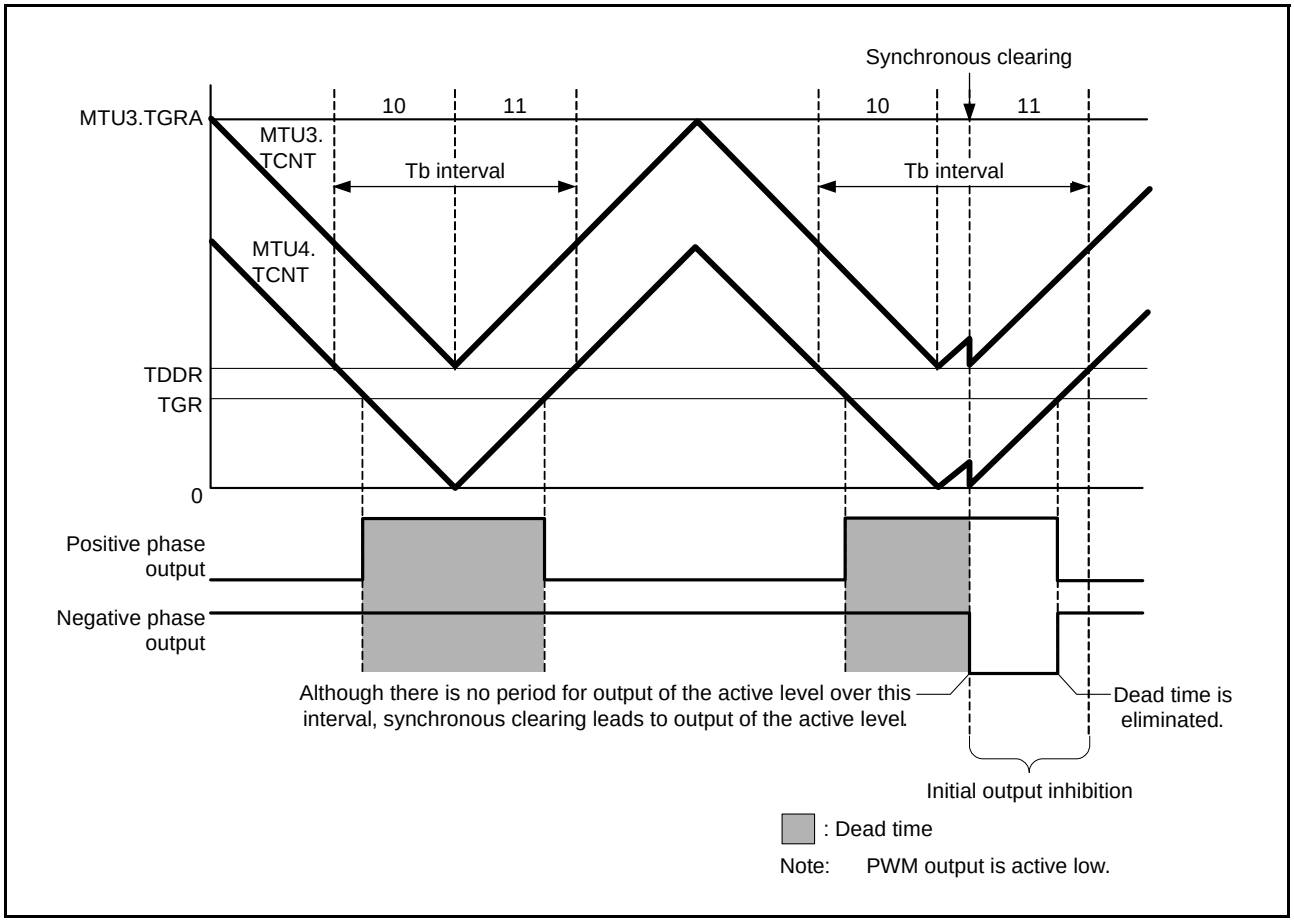


Figure 24.150 Example of Synchronous Clearing (When Condition 2 Applies)

24.6.26 Notes on Timer Mode Register Setting for ELC Event Input

When MTU is used in ELC operation, set the timer mode register (TMDR) of the relevant channel to its initial value (00h).

24.6.27 Continuous Output of Interrupt Signal in Response to a Compare Match

When the TGR register is set to 0000h, the PCLKA/1 clock is set as the count clock, and compare match is set as the trigger for clearing of the count clock, the value of the TCNT counter remains 0000h, and the interrupt signal will be output continuously (i.e. its level will be flat) rather than output over a single cycle. Consequently, interrupts will not be detected in response to second and subsequent compare matches.
Figure 24.151 shows the timing for continuous output of the interrupt signal in response to a compare match.

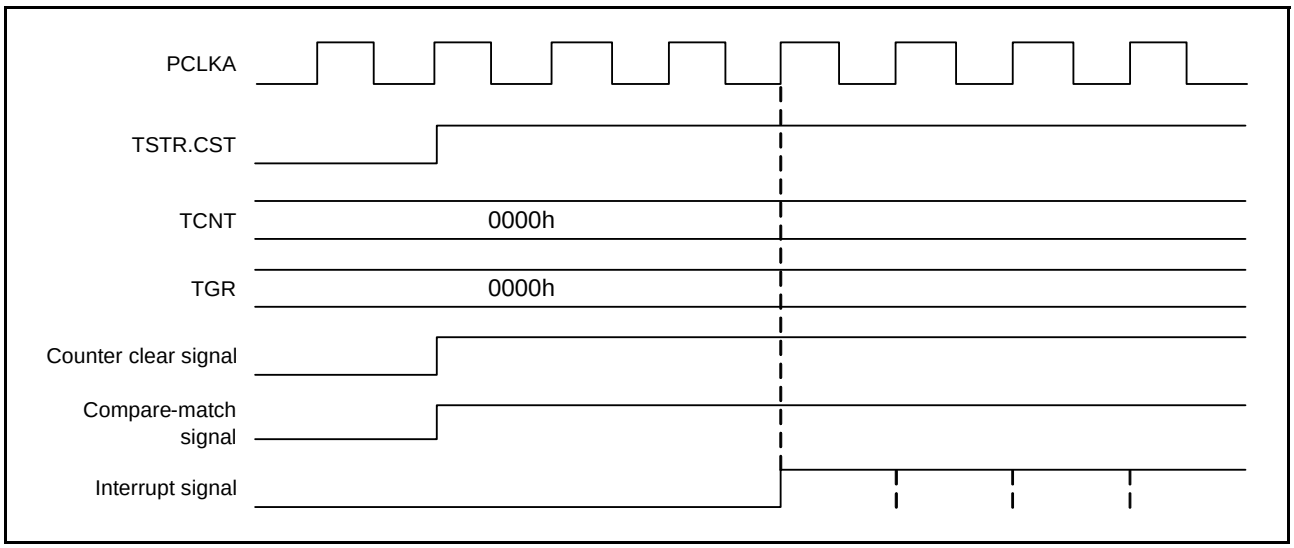


Figure 24.151 Continuous Output of Interrupt Signal in Response to a Compare Match

24.6.28 Usage Notes on A/D Converter Delaying Function in Complementary PWM Mode

- When data is transferred from a buffer register at the trough of the MTU4.TCNT (MTU7.TCNT) counter while the MTU4.TADCOBRA and MTU4.TADCOBRB (MTU7.TADCOBRA and MTU7.TADCOBRB) registers are set to 0 and the UT4AE and UT4BE (UT7AE and UT7BE) bits in the MTU4.TADCR (MTU7.TADCR) register are set to 1, no A/D converter start request is issued during up-counting immediately after transfer. Refer to Figure 24.152.
- When data is transferred from a buffer register at the crest of the MTU4.TCNT (MTU7.TCNT) counter while the MTU4.TADCOBRA and MTU4.TADCOBRB (MTU7.TADCOBRA and MTU7.TADCOBRB) registers are set to the same value as the TCDR and the UT4AE and UT4BE (UT7AE and UT7BE) bits in the MTU4.TADCR (MTU7.TADCR) register are set to 1, no A/D converter start request is issued during down-counting immediately after transfer. Refer to Figure 24.153.
- To issue an A/D converter start request linked with the interrupt skipping function, set the MTU4.TADCORA and MTU4.TADCORB (MTU7.TADCORA and MTU7.TADCORB) registers so that $2 \leq$ MTUn.TADCORA/ TADCORB $\leq$ TCDR – 2 is satisfied (n = 4, 7).

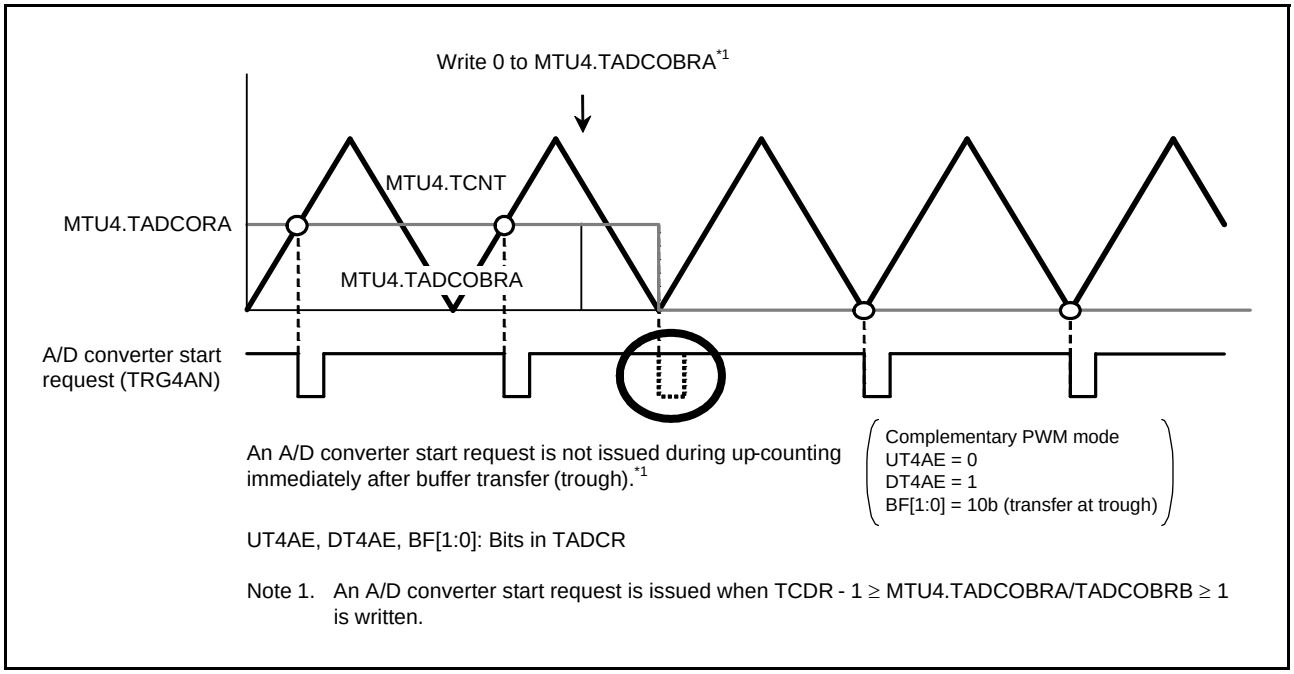


Figure 24.152 A/D Converter Start Request When 0 is Written to MTU4.TADCOBRA

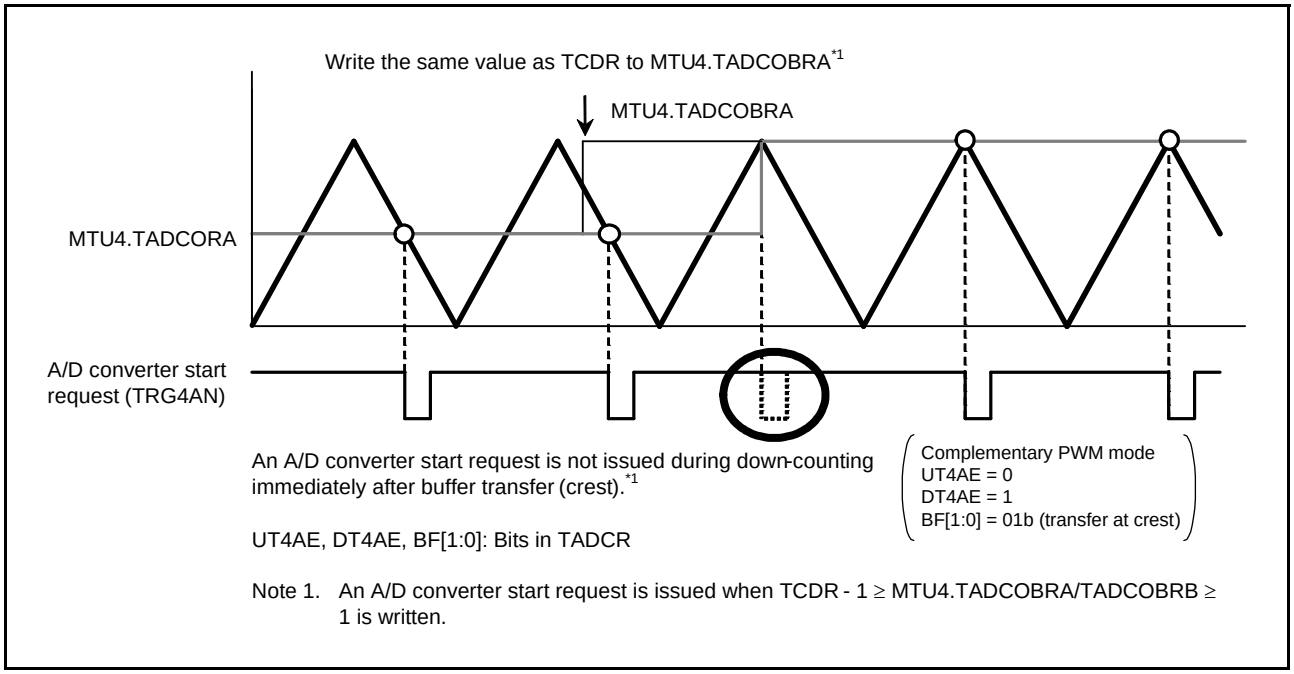


Figure 24.153 A/D Converter Start Request When the Same Value as TCDR is Written to MTU4.TADCOBRA

24.7 MTU Output Pin Initialization

24.7.1 Operating Modes

The MTU has the following six operating modes. Waveforms can be output in any of these modes.

- Normal mode (MTU0 to MTU4 and MTU6 to MTU8)
- PWM mode 1 (MTU0 to MTU4, MTU6, and MTU7)
- PWM mode 2 (MTU0 to MTU2)
- Phase counting modes 1 to 5 (MTU1 and MTU2)
- Complementary PWM mode (MTU3, MTU4, MTU6, and MTU7)
- Reset-synchronized PWM mode (MTU3, MTU4, MTU6, and MTU7)

This section describes how to initialize the MTU output pins in each of these modes.

24.7.2 Operation in Case of Re-Setting Due to Error during Operation

If an error occurs during MTU operation, MTU output should be cut off by the system. The output can be cut off by allowing non-active level output from the pins by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports. MTU output can be disabled through TIOR settings. Complementary PWM output (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC7C, and MTIOC7D) should be specified through TOERA and TOERB settings. For PWM output pins, output can also be cut by hardware, using port output enable 3 (POE3). The pin initialization procedures for re-setting due to an error during operation and the procedures for restarting in a different mode after re-setting are described below.

The MTU has six operating modes, as stated above. There are thus 36 mode transition combinations, but some transitions are not available with certain channel and mode combinations. Available mode transition combinations are shown in Table 24.80.

Table 24.80 Mode Transition Combinations

	Normal	PWM1	PWM2	PCM	CPWM	RPWM
Normal	(1)	(2)	(3)	(4)	(5)	(6)
PWM1	(7)	(8)	(9)	(10)	(11)	(12)
PWM2	(13)	(14)	(15)	(16)	Not available	Not available
PCM	(17)	(18)	(19)	(20)	Not available	Not available
CPWM	(21)	(22)	Not available	Not available	(23) (24)	(25)
RPWM	(26)	(27)	Not available	Not available	(28)	(29)

Normal: Normal mode
PWM1: PWM mode 1
PWM2: PWM mode 2
PCM: Phase counting modes 1 to 5
CPWM: Complementary PWM mode
RPWM: Reset-synchronized PWM mode

24.7.3 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation

- When making a transition to a mode (Normal, PWM1, PWM2, or PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of TIOR setting.
- In PWM mode 1, waveforms are not output to the MTIOCnB and MTIOCnD (n = 3, 4, 6, 7) pins. When a pin is configured for MTIOCnB or MTIOCnD, it enters high-impedance state. To output a specified level, set the pin to general output port.
- In PWM mode 2, waveforms are not output to the cycle register pins. When a pin is configured for MTIOCnm (n = 0 to 2; m = A to D), it enters high-impedance state. To output a specified level, set the pin to general output port.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, waveforms are not output to the corresponding pins (MTIOCnC or MTIOCnD (n = 0, 3, 4, 6, 7)). When a pin is configured for MTIOCnC or MTIOCnD, it enters high-impedance state. To output a specified level, set the pin to general output port.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, waveforms are not output to the corresponding pins (MTIOCnC or MTIOCnD (n = 0, 3, 4, 6, 7)). When a pin is configured for MTIOCnC or MTIOCnD, it enters high-impedance state. To output a specified level, set the pin to general output port.
- When making a transition to a mode (CPWM or RPWM) in which the pin output level is selected by the timer output control register (TOCR1A, TOCR2A, TOCR1B, or TOCR2B) setting, temporarily disable output in MTU3 and MTU4 (or MTU6 and MTU7) with the timer output master enable register (TOERA or TOERB). At this time, when a pin is configured for MTIOCnm (n = 3, 4, 6, 7; m = A to D), it enters high-impedance state. To output a specified level, set the pin to general output port. Switch to normal mode, perform initialization with TIOR, restore TIOR to its initial value, then operate the MTU in accordance with the mode setting procedure (TOCR1A setting, TOCR2A setting, TMDR1 setting, and TOERA setting (TOCR1B setting, TOCR2B setting, TMDR1 setting, and TOERB setting)).

Note: Channel number is substituted for “n” indicated in this section.

Pin initialization procedures are described below for the numbered combinations in Table 24.80. The active level is assumed to be low.

(1) Operation When Error Occurs in Normal Mode and Operation is Restarted in Normal Mode

Figure 24.154 shows a case in which an error occurs in normal mode and operation is restarted in normal mode after re-setting.

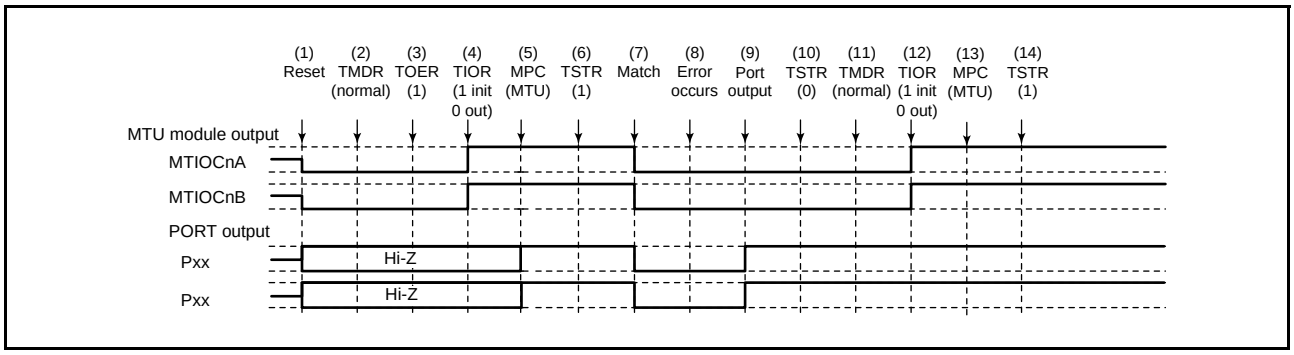


Figure 24.154 Error Occurrence in Normal Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) After a reset, the TMDR1 setting is for normal mode.
(3) For MTU3 and MTU4, enable output with the TOERA register before initializing the pins with the TIOR register.
(4) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)
(5) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting the TSTRA register.
(7) Output goes low on compare match occurrence.
(8) An error occurs.
(9) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting the TSTRA register.
(11) This step is not necessary when restarting in normal mode.
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

(2) Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 1

Figure 24.155 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

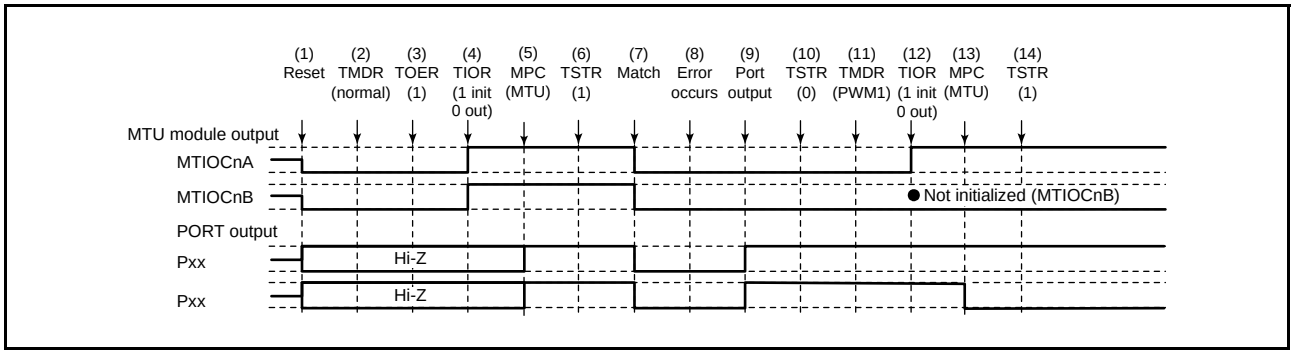


Figure 24.155 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 24.154.
(11) Set PWM mode 1.
(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

(3) Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM mode 2

Figure 24.156 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

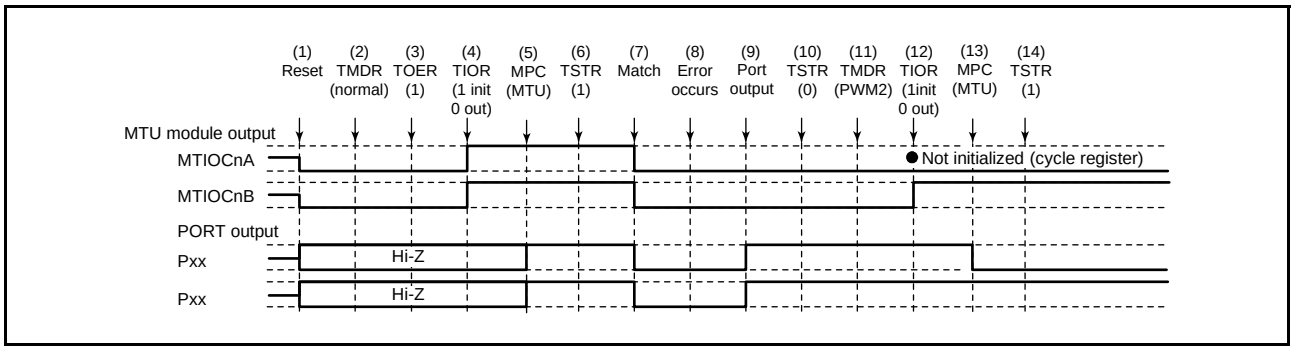


Figure 24.156 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

(1) to (10) are the same as in Figure 24.154.
(11) Set PWM mode 2.
(12) Initialize the pins with the TIOR register. (In PWM mode 2, waveforms are not output to the cycle register pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

Note: PWM mode 2 can only be selected for MTU0 to MTU2, and therefore the TOERA register setting is not necessary.

(4) Operation When Error Occurs in Normal Mode and Operation is Restarted in Phase Counting Mode

Figure 24.157 shows a case in which an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

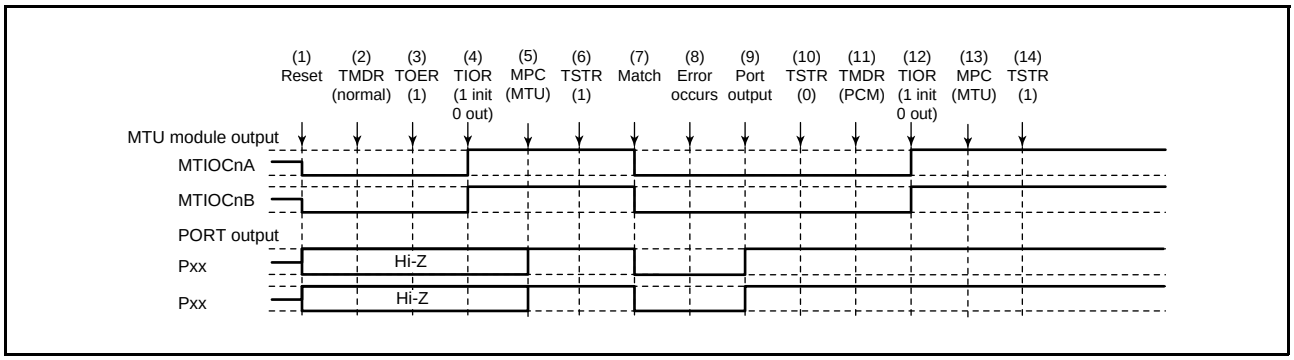


Figure 24.157 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

(1) to (10) are the same as in Figure 24.154.
(11) Set the phase counting mode.
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

Note: The phase counting mode can only be selected for MTU1 and MTU2, and therefore the TOERA register setting is not necessary.

(5) Operation When Error Occurs in Normal Mode and Operation is Restarted in Complementary PWM Mode

Figure 24.158 shows a case in which an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

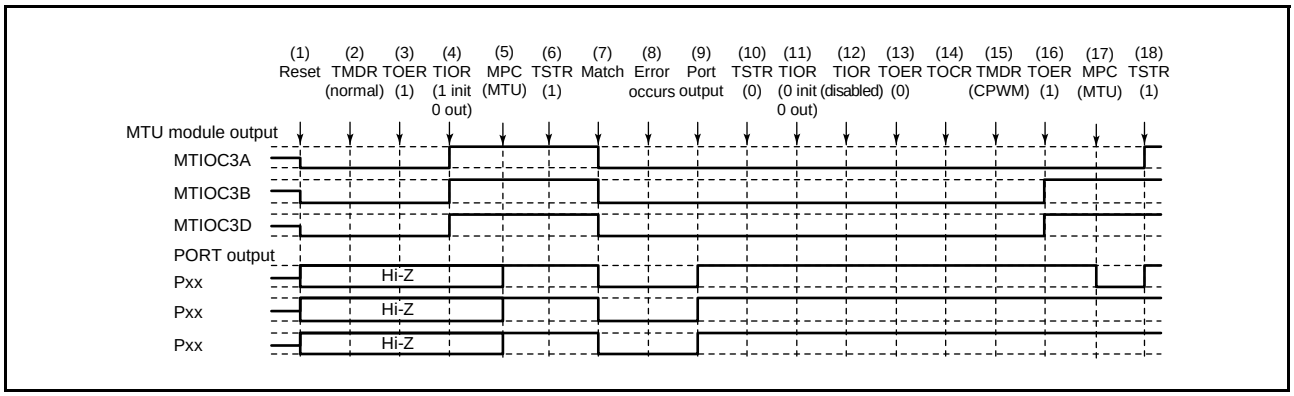


Figure 24.158 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 24.154.
(11) Initialize the normal mode waveform generation block with the TIOR register.
(12) Disable operation of the normal mode waveform generation block with the TIOR register.
(13) Disable output in MTU3 and MTU4 with the TOERA register.
(14) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A.
(15) Set complementary PWM mode.
(16) Enable output in MTU3 and MTU4 with the TOERA register.
(17) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(18) Restart operation by setting the TSTRA register.

(6) Operation When Error Occurs in Normal Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 24.159 shows a case in which an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

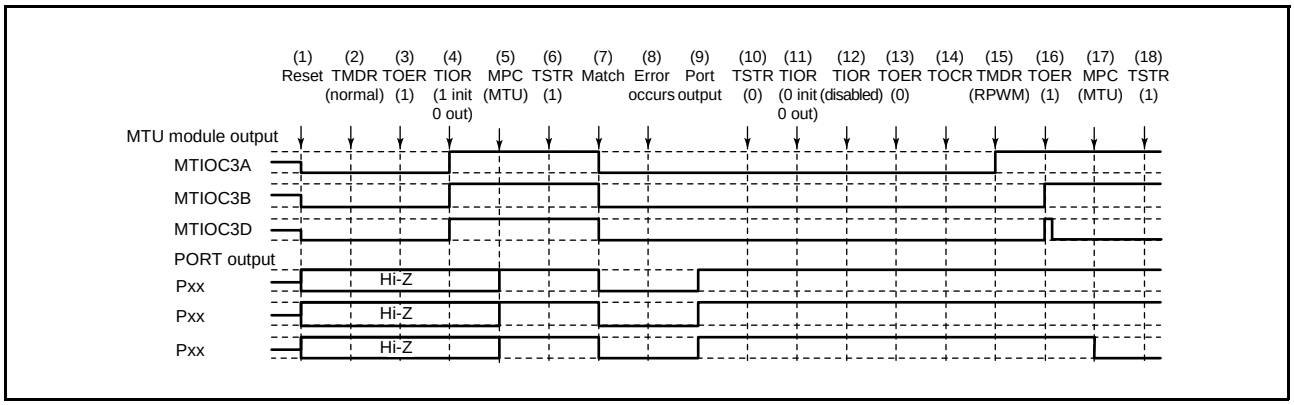


Figure 24.159 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (13) are the same as in Figure 24.158.

(14) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A.

(15) Set reset-synchronized PWM mode.

(16) Enable output in channels 3 and 4 with the TOERA register.

(17) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(18) Restart operation by setting the TSTRA register.

(7) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Normal Mode

Figure 24.160 shows a case in which an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

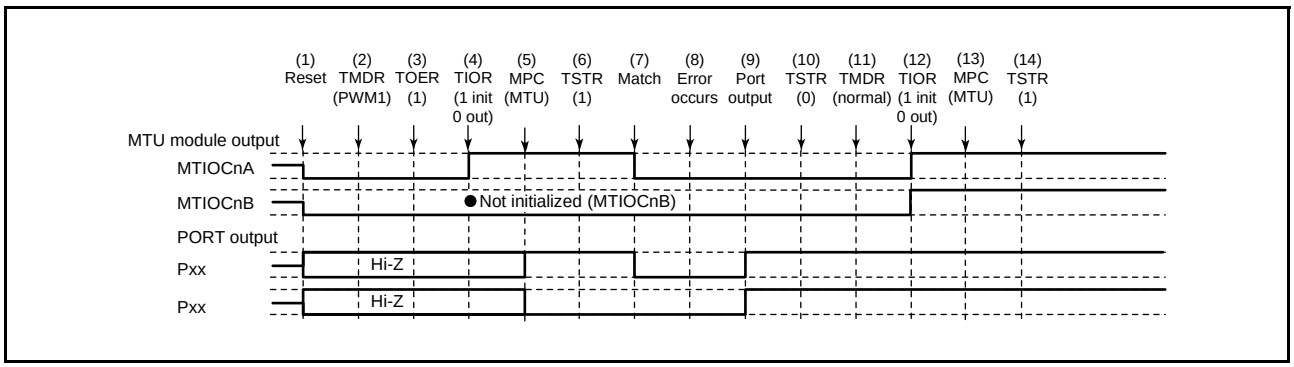


Figure 24.160 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) Set PWM mode 1.
(3) For MTU3 and MTU4, enable output with the TOERA register before initializing the pins with the TIOR register.
(4) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence. In PWM mode 1, the MTIOCnB side is not initialized.)
(5) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting the TSTRA register.
(7) Output goes low on compare match occurrence.
(8) An error occurs.
(9) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting the TSTRA register.
(11) Set normal mode.
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

(8) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 1

Figure 24.161 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

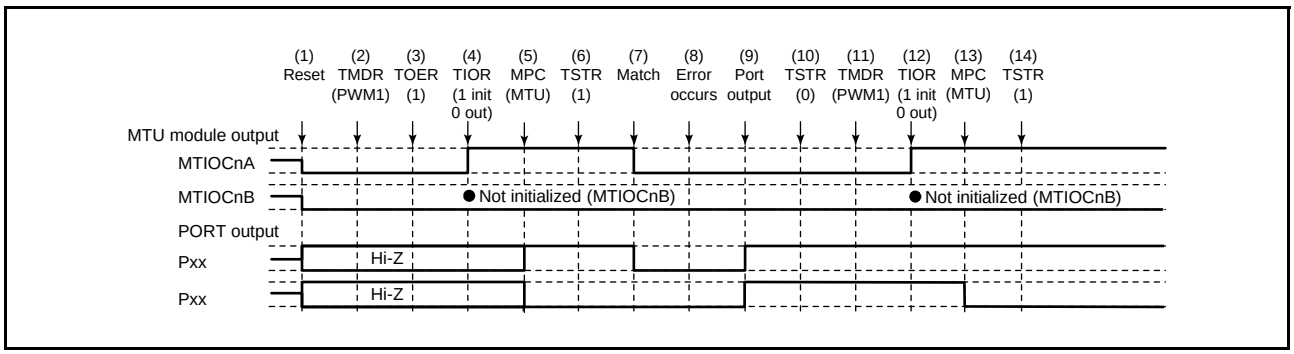


Figure 24.161 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 24.160.
(11) This step is not necessary when restarting in PWM mode 1.
(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

(9) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 2

Figure 24.162 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

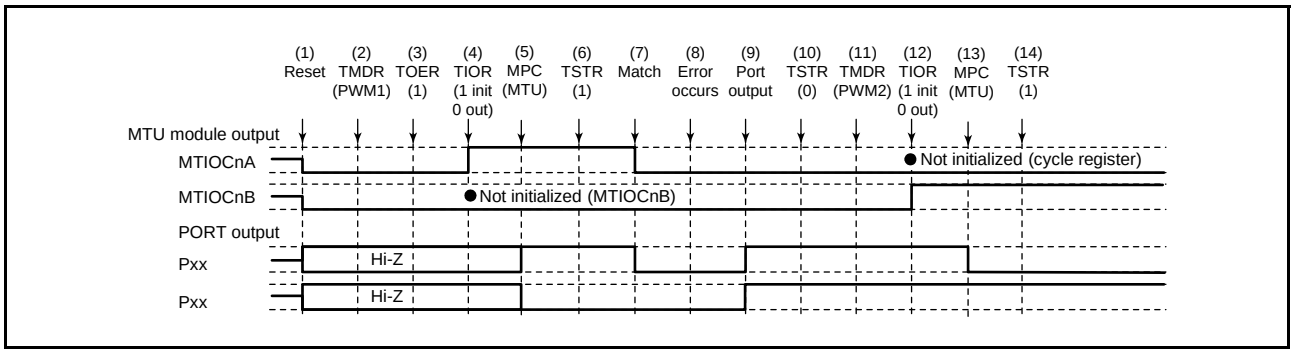


Figure 24.162 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

(1) to (10) are the same as in Figure 24.160.
(11) Set PWM mode 2.
(12) Initialize the pins with the TIOR register. (In PWM mode 2, waveforms are not output to the cycle register pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

Note: PWM mode 2 can only be selected for MTU0 to MTU2, and therefore the TOERA register setting is not necessary.

(10) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Phase Counting Mode

Figure 24.163 shows a case in which an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

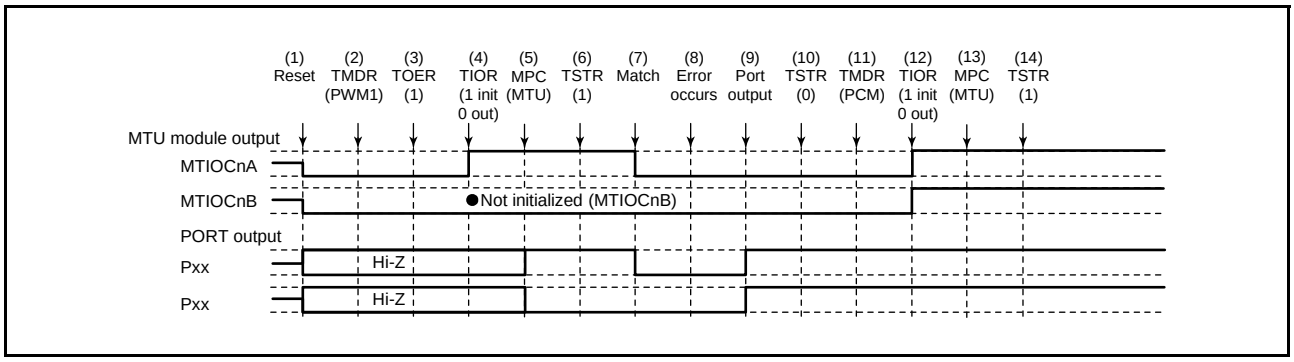


Figure 24.163 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

(1) to (10) are the same as in Figure 24.160.
(11) Set the phase counting mode.
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

Note: The phase counting mode can only be selected for MTU1 and MTU2, and therefore the TOERA register setting is not necessary.

(11) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Complementary PWM Mode

Figure 24.164 shows a case in which an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

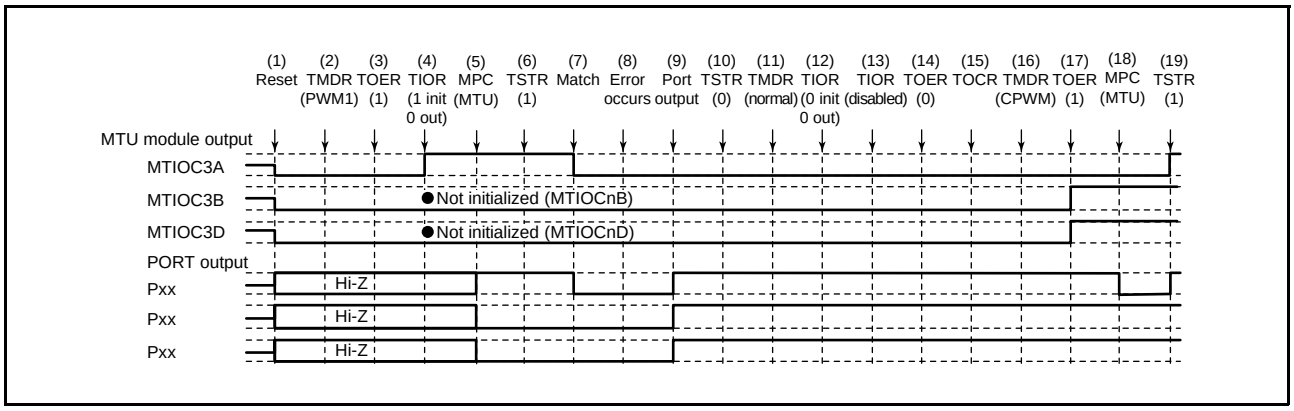


Figure 24.164 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 24.160.
(11) Set normal mode to initialize the normal mode waveform generation block.
(12) Initialize the PWM mode 1 waveform generation block with the TIOR register.
(13) Disable operation of the PWM mode 1 waveform generation block with the TIOR register.
(14) Disable output in MTU3 and MTU4 with the TOERA register.
(15) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A.
(16) Set complementary PWM mode.
(17) Enable output in MTU3 and MTU4 with the TOERA register.
(18) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(19) Restart operation by setting the TSTRA register.

(12) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 24.165 shows a case in which an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

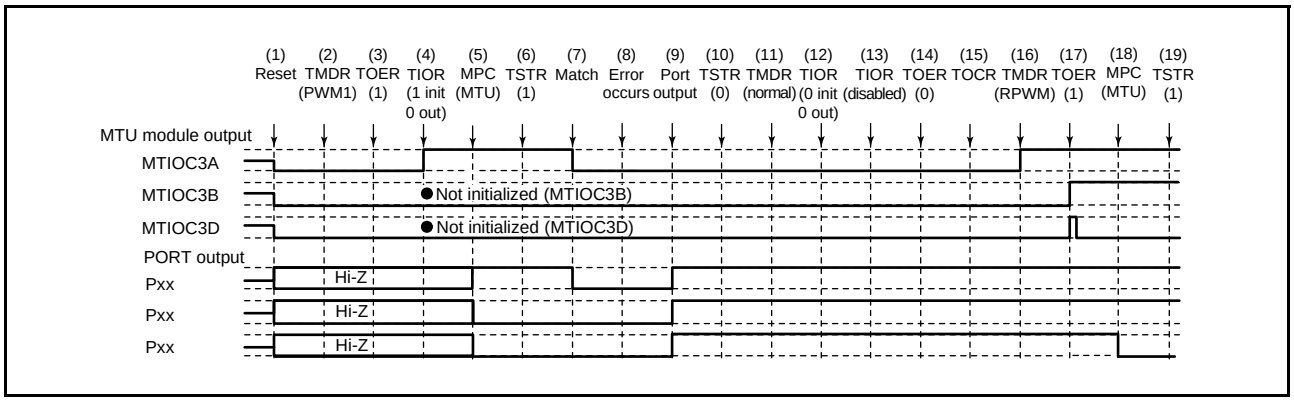


Figure 24.165 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

(1) to (14) are the same as in Figure 24.164.

(15) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A.

(16) Set reset-synchronized PWM mode.

(17) Enable output in MTU3 and MTU4 with the TOERA register.

(18) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(19) Restart operation by setting the TSTRA register.

(13) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Normal Mode

Figure 24.166 shows a case in which an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

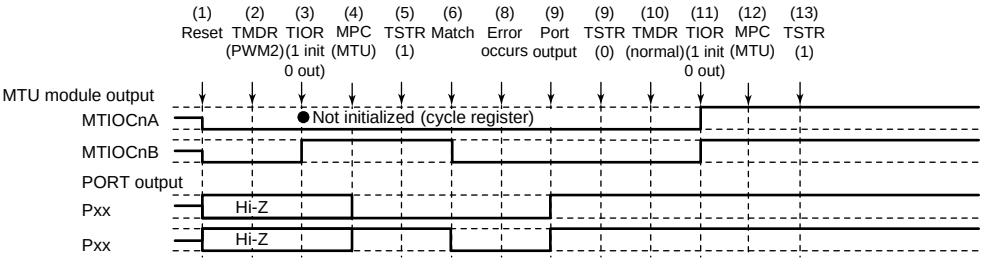


Figure 24.166 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) Set PWM mode 2.
(3) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence. In PWM mode 2, the cycle register pins are not initialized. In the example, the MTIOCnA pin is the cycle register pin.)
(4) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(5) Start count operation by setting the TSTRA register.
(6) Output goes low on compare match occurrence.
(7) An error occurs.
(8) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(9) Stop count operation by setting the TSTRA register.
(10) Set normal mode.
(11) Initialize the pins with the TIOR register.
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(14) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 1

Figure 24.167 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

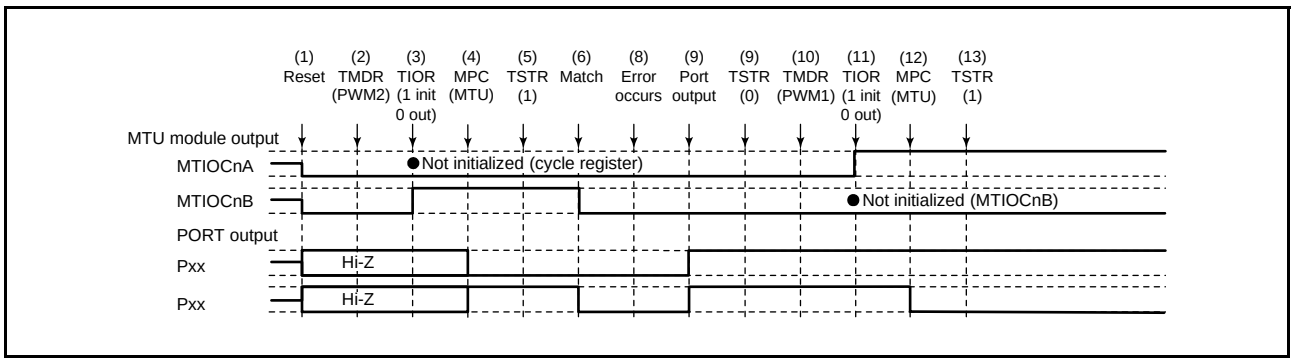


Figure 24.167 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 24.166.
(10) Set PWM mode 1.
(11) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins.
To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(15) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 2

Figure 24.168 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

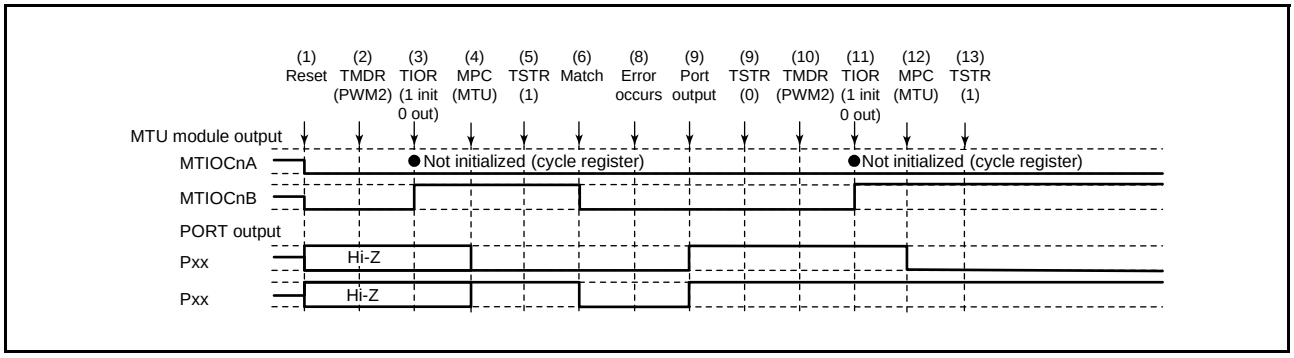


Figure 24.168 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

(1) to (9) are the same as in Figure 24.166.
(10) This step is not necessary when restarting in PWM mode 2.
(11) Initialize the pins with the TIOR register. (In PWM mode 2, waveforms are not output to the cycle register pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(16) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Phase Counting Mode

Figure 24.169 shows a case in which an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

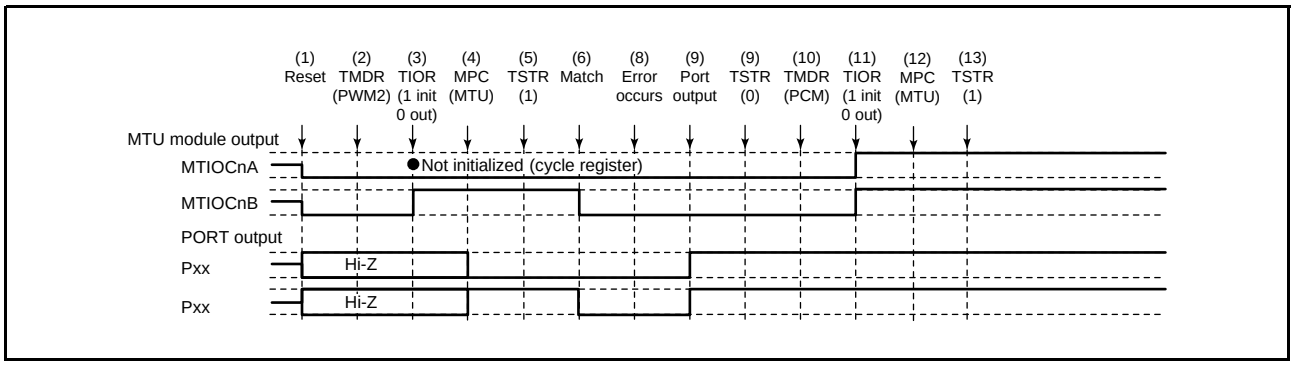


Figure 24.169 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

(1) to (9) are the same as in Figure 24.166.
(10) Set the phase counting mode.
(11) Initialize the pins with the TIOR register.
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(17) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Normal Mode

Figure 24.170 shows a case in which an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

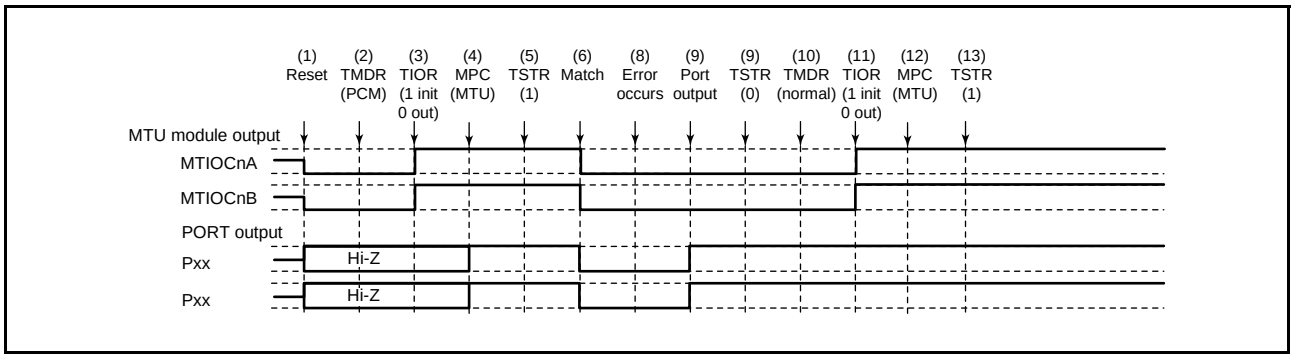


Figure 24.170 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) Set phase counting mode.
(3) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)
(4) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(5) Start count operation by setting the TSTRA register.
(6) Output goes low on compare match occurrence.
(7) An error occurs.
(8) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(9) Stop count operation by setting the TSTRA register.
(10) Set normal mode.
(11) Initialize the pins with the TIOR register.
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(18) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 1

Figure 24.171 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

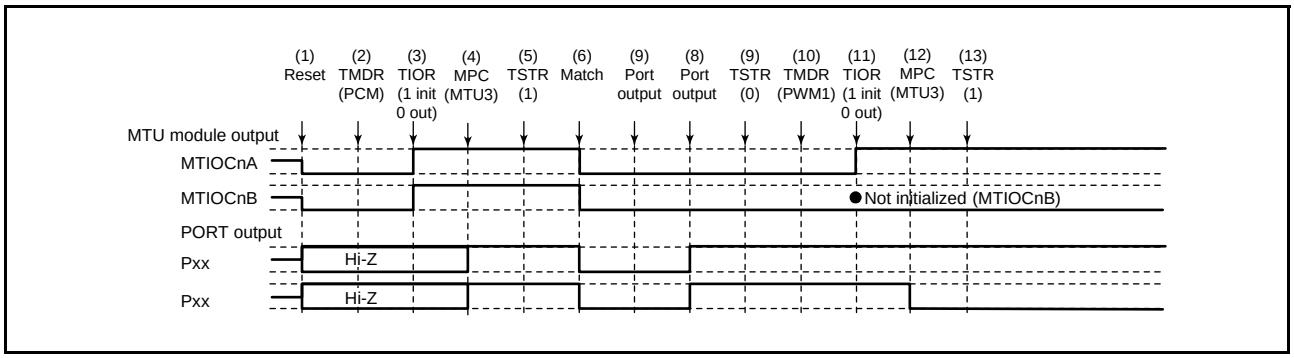


Figure 24.171 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 24.170.
(10) Set PWM mode 1.
(11) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins.
To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(19) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 2

Figure 24.172 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

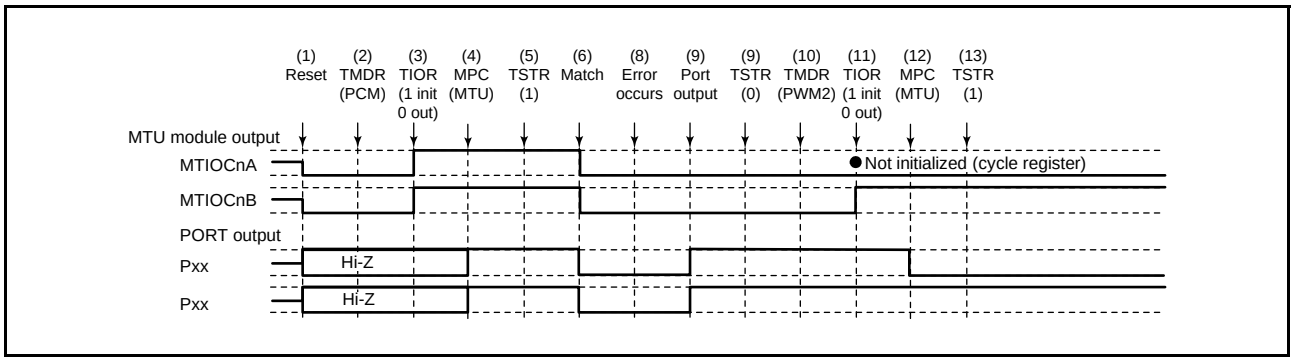


Figure 24.172 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

(1) to (9) are the same as in Figure 24.170.
(10) Set PWM mode 2.
(11) Initialize the pins with the TIOR register. (In PWM mode 2, waveforms are not output to the cycle register pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(20) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Phase Counting Mode

Figure 24.173 shows a case in which an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

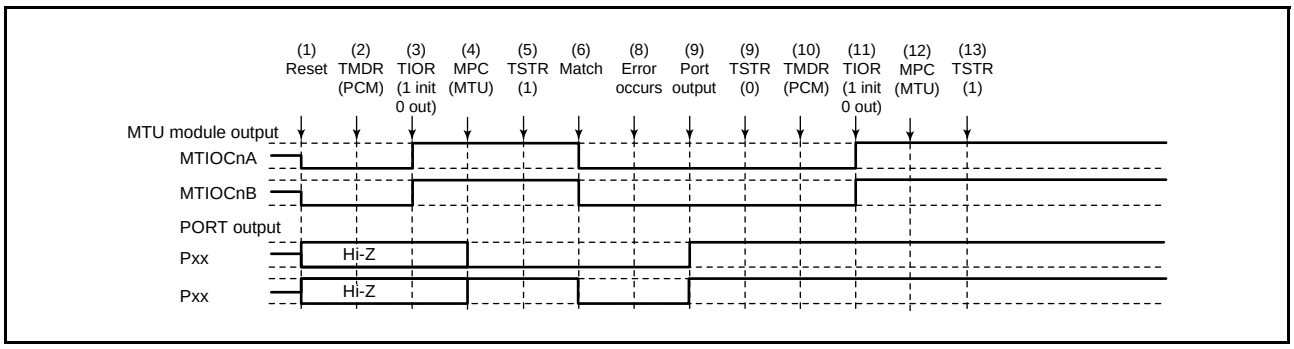


Figure 24.173 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

(1) to (9) are the same as in Figure 24.170.
(10) This step is not necessary when restarting in phase counting mode.
(11) Initialize the pins with the TIOR register.
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(21) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Normal Mode

Figure 24.174 shows a case in which an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

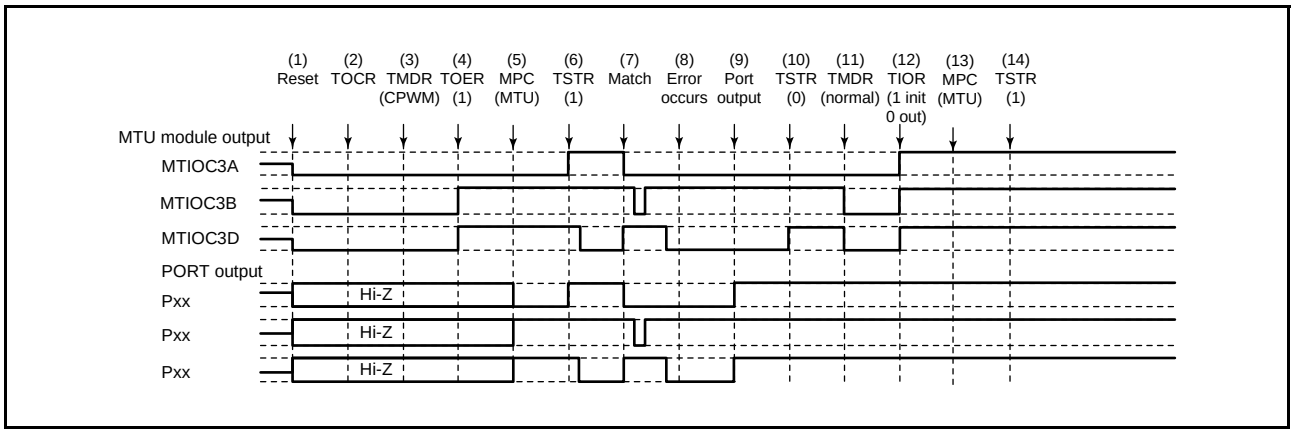


Figure 24.174 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

(1) After a reset, the MTU3 output goes low and the ports enter high-impedance state.
(2) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A.
(3) Set complementary PWM mode.
(4) Enable output in MTU3 and MTU4 with the TOERA register.
(5) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting the TSTRA register.
(7) The complementary PWM waveform is output on compare match occurrence.
(8) An error occurs.
(9) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting the TSTRA register. (MTU output becomes the initial complementary PWM output value).
(11) Set normal mode (MTU output goes low).
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

(22) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in PWM Mode 1

Figure 24.175 shows a case in which an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

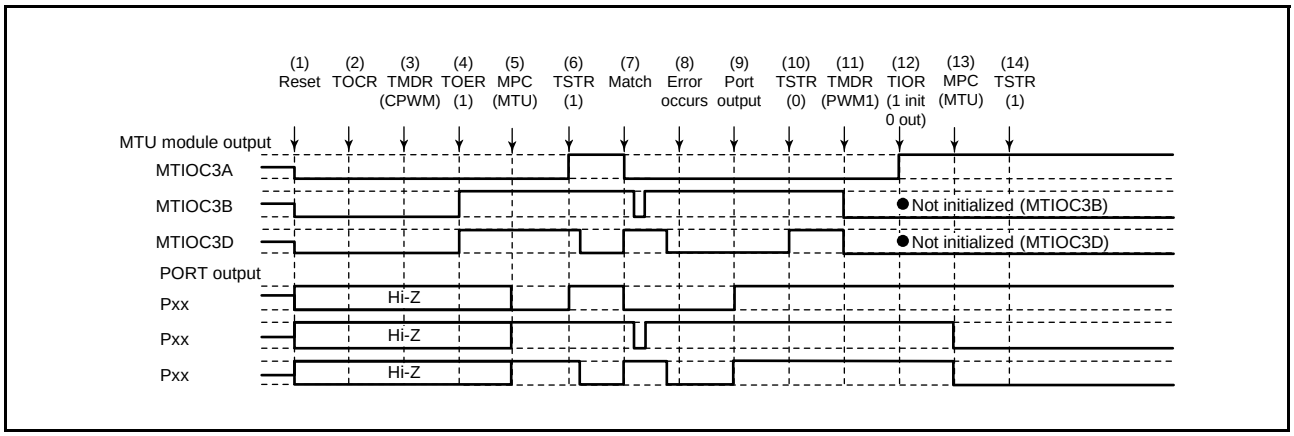


Figure 24.175 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 24.174.
(11) Set PWM mode 1 (MTU output goes low).
(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

(23) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 24.176 shows a case in which an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using the cycle and duty settings at the time of stopping the counter).

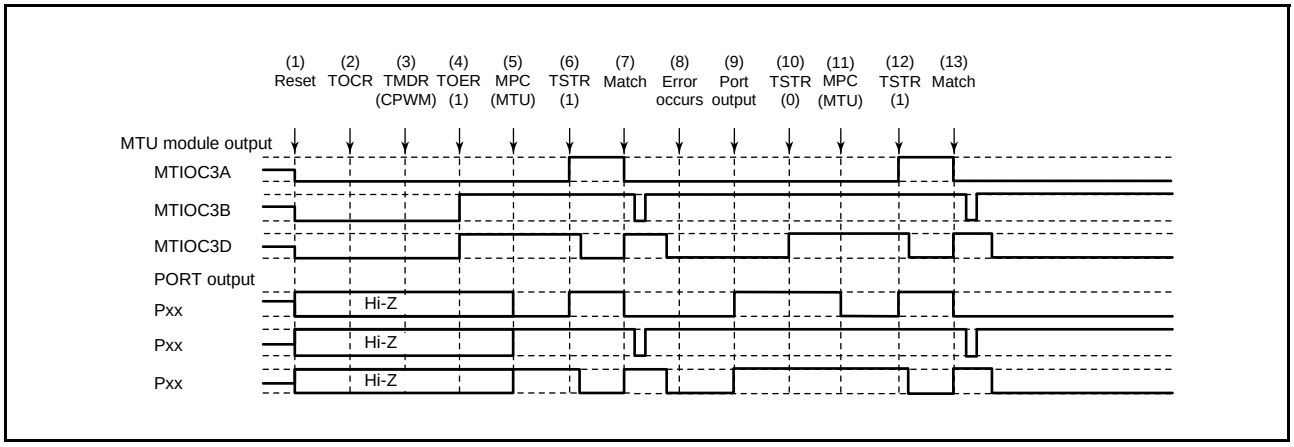


Figure 24.176 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode (1)

(1) to (10) are the same as in Figure 24.174.
(11) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(12) Restart operation by setting the TSTRA register.
(13) The complementary PWM waveform is output on compare match occurrence.

(24) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode with New Settings

Figure 24.177 shows a case in which an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (operation is restarted using new cycle and duty ratio settings).

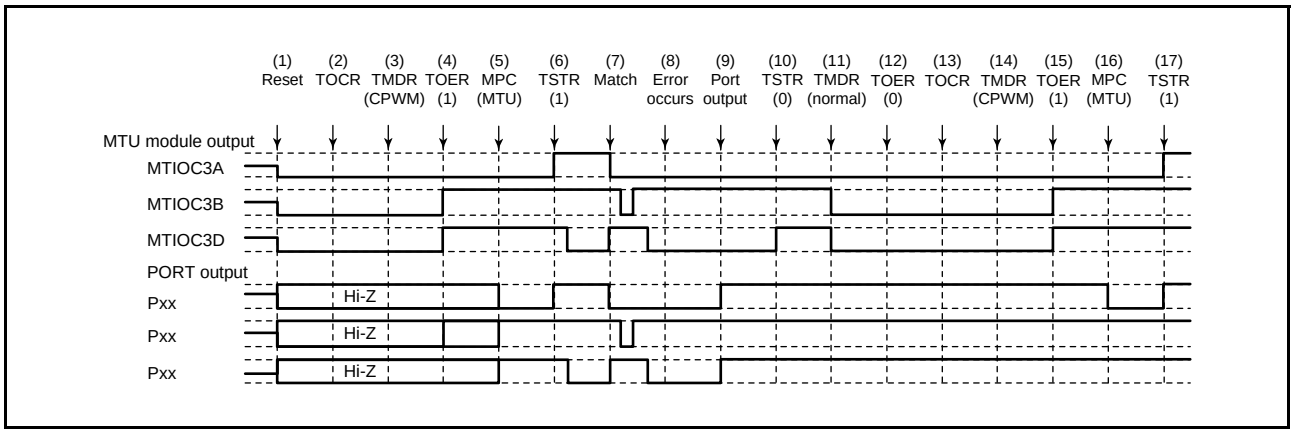


Figure 24.177 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode (2)

(1) to (10) are the same as in Figure 24.174.
(11) Set normal mode and make new settings (MTU output goes low).
(12) Disable output in MTU3 and MTU4 with the TOERA register.
(13) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A.
(14) Set complementary PWM mode.
(15) Enable output in MTU3 and MTU4 with the TOERA register.
(16) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(17) Restart operation by setting the TSTRA register.

(25) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 24.178 shows a case in which an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

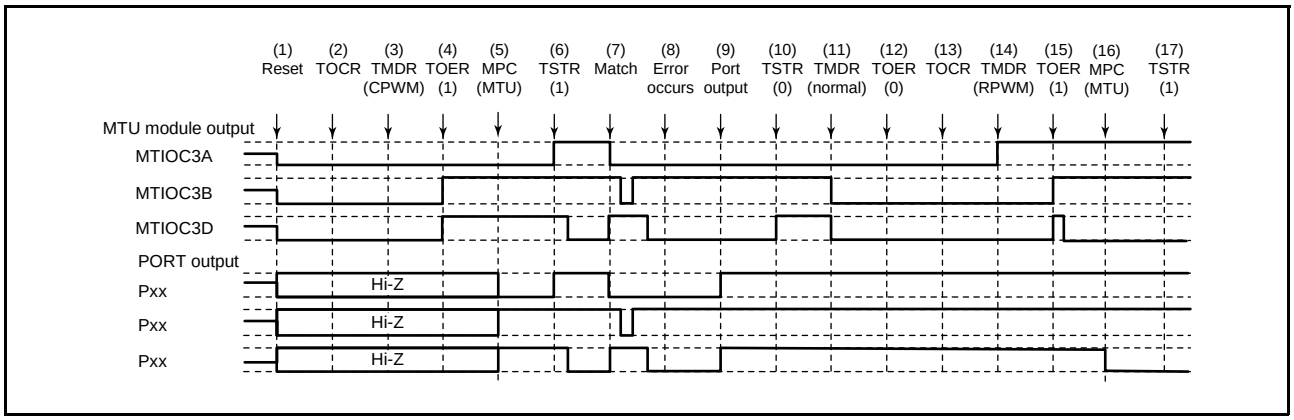


Figure 24.178 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (10) are the same as in Figure 24.174.
(11) Set normal mode (MTU output goes low).
(12) Disable output in MTU3 and MTU4 with the TOERA register.
(13) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A.
(14) Set reset-synchronized PWM mode.
(15) Enable output in MTU3 and MTU4 with the TOERA register.
(16) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(17) Restart operation by setting the TSTRA register.

(26) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Normal Mode

Figure 24.179 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

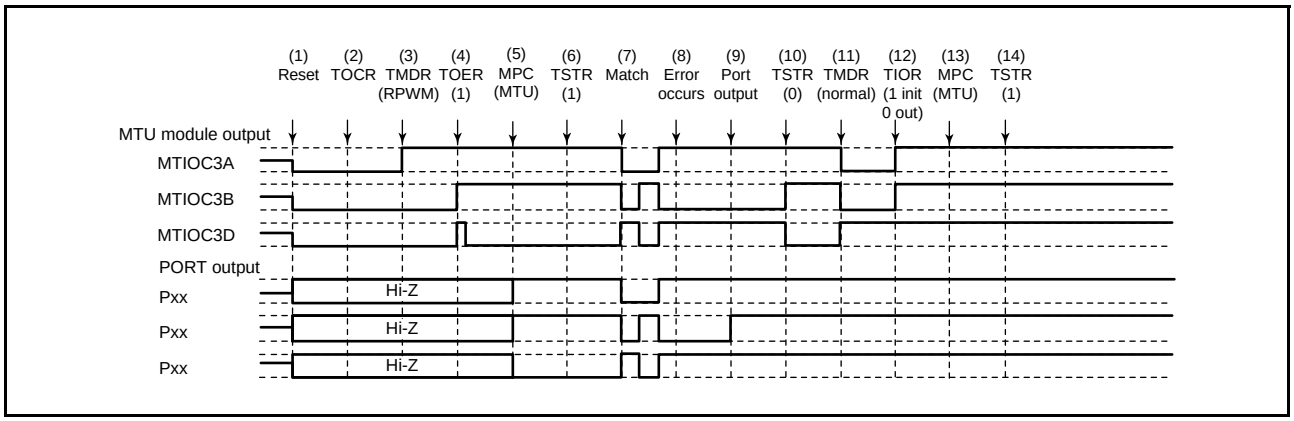


Figure 24.179 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A.
(3) Set reset-synchronized PWM mode.
(4) Enable output in MTU3 and MTU4 with the TOERA register.
(5) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting the TSTRA register.
(7) The reset-synchronized PWM waveform is output on compare match occurrence.
(8) An error occurs.
(9) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting the TSTRA register. (MTU output becomes the initial reset-synchronized PWM output value.)
(11) Set normal mode (positive-phase MTU output goes low, and negative-phase output goes high).
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

(27) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in PWM Mode 1

Figure 24.180 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

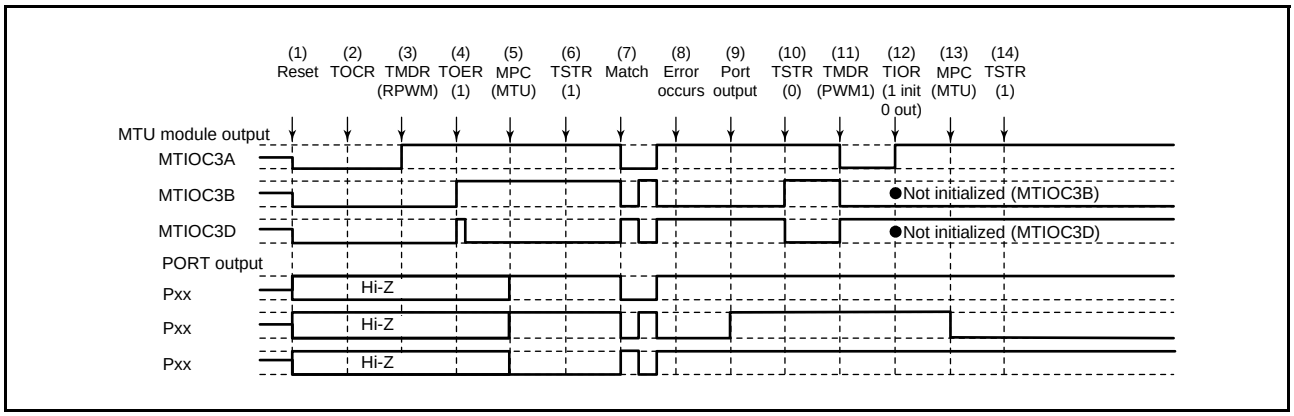


Figure 24.180 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 24.179.

(11) Set PWM mode 1 (positive-phase MTU output goes low, and negative-phase output goes high).

(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)

(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA register.

(28) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 24.181 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

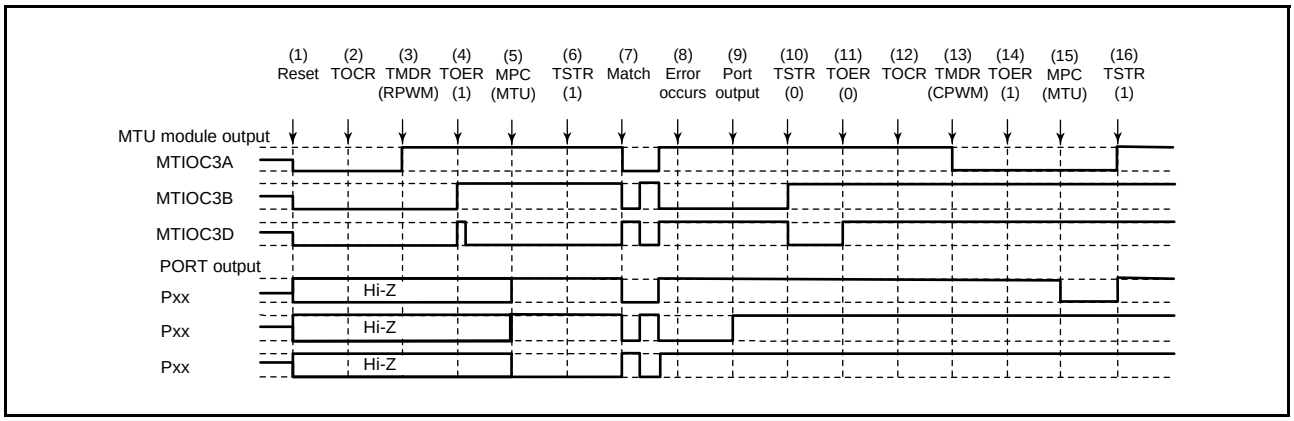


Figure 24.181 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 24.179.
(11) Disable output in MTU3 and MTU4 with the TOERA register.
(12) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A.
(13) Set complementary PWM mode (MTU cyclic output pin goes low).
(14) Enable output in MTU3 and MTU4 with the TOERA register.
(15) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(16) Restart operation by setting the TSTRA register.

(29) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 24.182 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

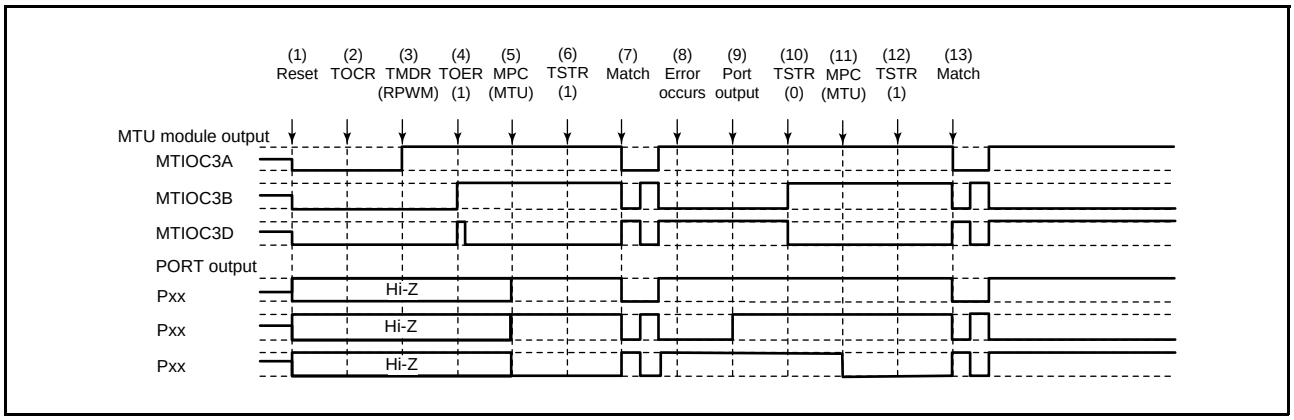


Figure 24.182 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (10) are the same as in Figure 24.179.

(11) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(12) Restart operation by setting the TSTRA register.

(13) The reset-synchronized PWM waveform is output on compare match occurrence.

24.8 Operations Linked by the ELC

24.8.1 Event Signal Output to the ELC

The MTU is capable of operation linked with another module set in advance when its interrupt request signal is used as an event signal by the event link controller (ELC).
The MTU outputs the event signal regardless of the setting of the corresponding interrupt request enable bit.

24.8.2 MTU Operations in Response to Receiving Event Signals from the ELC

The MTU can perform the following operations in response to the event set in advance in the ELSRn register of the event link controller (ELC).

(1) Start Counting Operation

Counting by the MTU starts in response to the event when this is selected by the setting of the ELOPA or ELOPB register of the ELC. The ELOPA register handles control for MTU0 and MTU3 and the ELOPB register handles control for MTU4. When the event specified in the ELSRn register occurs, the CSTn bit in the TSTRA register (timer start register) shown in Table 24.81 is set to 1, and the MTU counter starts.
However, when the specified event is generated while the CSTn bit in the TSTRA register (timer start register) has already been set to 1, the event has no effect. Table 24.81 lists the TSTRA register bits used for each channel.

Table 24.81 Operation of the TSTRA register (timer start register) in response to the ELC

Channel No.	TSTRA register (timer start register)
MTU0	TSTRA.CST0 bit
MTU3	TSTRA.CST3 bit
MTU4	TSTRA.CST4 bit

(2) Input Capture Operation

Input capture by the MTU proceeds when this is selected by the setting of the ELOPA or ELOPB register of the ELC. The ELOPA register handles control for MTU channels 1 to 3 and the ELOPB register handles control for MTU channel 4. When the event specified in the ELSRn register occurs, the TGR register (timer general register) captures the value of the TCNT register (timer counter register). When using input capture in response to an event, the corresponding bit of the TIOR register (timer I/O control register) in the MTU should be set for input capture and the CSTn bit of TSTRA register (timer start register) should be set to 1 to start counting by the counter.
In this case, the TIOCnA pin (input capture pin) input has no effect.
Table 24.82 lists the timer general register and timer I/O control register used for each channel in input capture operations in response to the ELC.

Table 24.82 Timer General Register and Timer I/O Control Register Used in the Input Capture Operation

Channel No.	Timer General Register Name	Timer I/O Control Register Bit Name
MTU0	MTU0.TGRA	MTU0.TIORH.IOA[3:0] bits
MTU3	MTU3.TGRA	MTU3.TIORH.IOA[3:0] bits
MTU4	MTU4.TGRA	MTU4.TIORH.IOA[3:0] bits

(3) Restart [Clear] Counting Operation

The counter is cleared in response to the event when the ELOPA or ELOPB register of the ELC is set for “restart counting” as the operation for the MTU. The ELOPA register handles control for MTU0 and MTU3 and the ELOPB register handles control for MTU4. When the event specified in the ELSRn register occurs, the TCNT register (timer counter register) is returned to its initial value. If the corresponding CSTn bit in the TSTRA register (timer start register) is set to 1, counting will continue. For the CSTn bits in the TSTRA register (timer start register), refer to Table 24.81.

24.8.3 Usage Notes on MTU Operation by Event Signal Reception from the ELC

The following notes on usage apply when the MTU is used in event link operation.

(1) Start Counting

If the event specified in the ELSRn register occurs during a cycle of writing to a CSTn bit in the TSTRA/TSTRB register, writing to the CSTn bit in the TSTRA/TSTRB register does not proceed because setting of the bit to 1 due to the event takes priority.

(2) Restart [Clear] Counting

If the event specified in the ELSRn register occurs during a cycle of writing to the TCNT counter, writing to the TCNT counter does not proceed because initialization of the counter’s value due to the event takes priority.
In addition, for MTU3 and MTU4 in complementary PWM mode, disable the counter clearing by the ELC.

25. Port Output Enable 3 (POE3a)

The registers of the port output enable 3 (POE3a) can be used to, under various conditions, switch output pins for the MTU and the GPT to the high-impedance state.
In this section, “PCLK” is used to refer to PCLKB.

25.1 Overview

Table 25.1 lists the specifications of the POE, and Figure 25.1 shows a block diagram of the POE.

Table 25.1 POE Specifications

Item	Description
Target pins for switching to high-impedance state	• MTU output pins MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D) MTU3 pins (MTIOC3B, MTIOC3D) MTU4 pins (MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D) MTU6 pins (MTIOC6B, MTIOC6D) MTU7 pins (MTIOC7A, MTIOC7B, MTIOC7C, MTIOC7D) • GPT output pins GPT0 pins (GTIOC0A, GTIOC0B) GPT1 pins (GTIOC1A, GTIOC1B) GPT2 pins (GTIOC2A, GTIOC2B) GPT3 pins (GTIOC3A, GTIOC3B)
Generating conditions of request for switching to high-impedance state	• Input signal detection: Detection of the POE0#, POE4#, POE8#, POE10#, and POE11# signal level. • Short-circuits between output pins: A match (short circuit) between the output signal levels at the active level over one or more cycles on the following combination of pins (1) MTU Complementary PWM Output Pins: 1 MTIOC3B and MTIOC3D; 2 MTIOC4A and MTIOC4C; 3 MTIOC4B and MTIOC4D; 4 MTIOC6B and MTIOC6D; 5 MTIOC7A and MTIOC7C; 6 MTIOC7B and MTIOC7D (2) GPT Output Pins: 1 GTIOC0A and GTIOC0B; 2 GTIOC1A and GTIOC1B; 3 GTIOC2A and GTIOC2B • Register setting for high-impedance being made • Detection that the main clock oscillator had stopped oscillating
Function	• Each of the POE0#, POE4#, POE8#, POE10#, and POE11# input pins can be set for falling edge, PCLK/8 × 16, PCLK/16 × 16, or PCLK/128 × 16 low-level sampling. • Pins for the MTU complementary PWM output, MTU0, GPT output, and GPT3 pins can be switched to the high-impedance state by falling-edge or low-level sampling of the POE0#, POE4#, POE8#, POE10#, or POE11# pin. • Pins for the MTU complementary PWM output, MTU0, GPT output, and GPT3 pins can be switched to the high-impedance state when oscillation stop is detected by the oscillation stop detection function of the clock generator. • Pins for the MTU complementary PWM output and GPT output pins can be switched to the high-impedance state when output levels of the MTU complementary PWM output pins and the GPT output pins (GPT0, GPT1, and GPT2) are compared and simultaneous active-level output continues for one cycle or more. • Pins for the MTU complementary PWM output, MTU0, GPT output, and GPT3 pins can be switched to the high-impedance state by modifying the settings of the POE registers. • Interrupts can be generated by input-level sampling or output-level comparison results.

The POE has input-level detection circuits, pin selection circuits, output-level comparison circuits, and a high-impedance request/interrupt request generating circuit as shown in Figure 25.1.

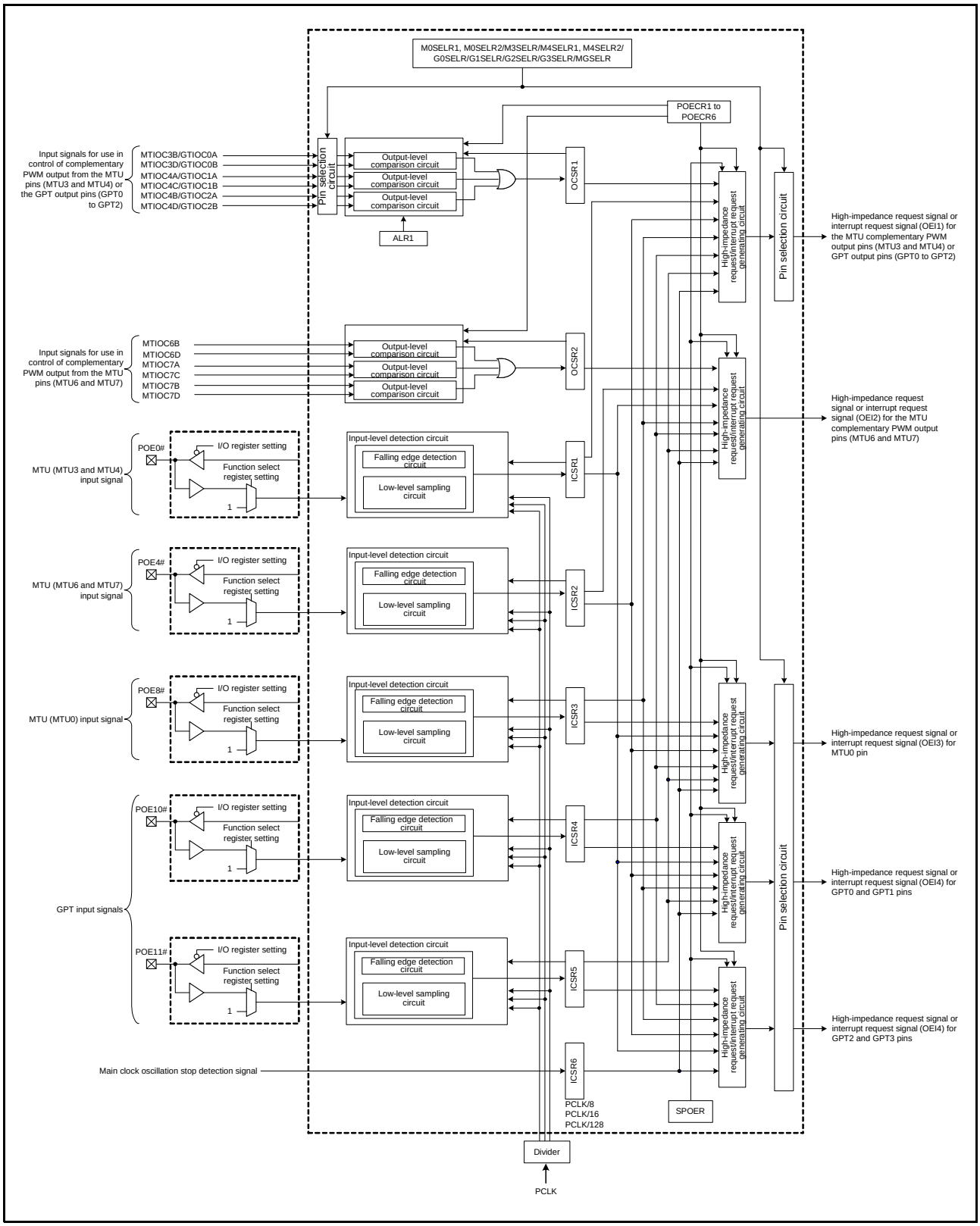


Figure 25.1 POE Block Diagram

Table 25.2 shows I/O pins to be used by the POE.

Table 25.2 POE I/O Pins

Pin Name	I/O	Description
POE0#	Input	Request signal to switch the MTU complementary PWM output pins (MTU3, MTU4 pins) or GPT output pins to the high-impedance state, and is also capable of switching the MTU0 pins, MTU complementary PWM output pins (MTU6, MTU7 pins), and other GPT pins by register settings.
POE4#	Input	Request signal to switch the MTU complementary PWM output pins (MTU6, MTU7 pins) to the high-impedance state, and is also capable of switching the MTU0 pins, MTU complementary PWM output pins (MTU3, MTU4 pins), and GPT pins by register settings.
POE8#	Input	Request signal to switch the MTU0 pins to the high-impedance state, and is also capable of switching the MTU complementary PWM output pins (MTU3, MTU4 pins or MTU6, MTU7 pins) and GPT pins by register settings.
POE10#	Input	Request signal to switch the GPT0 and GPT1 pins to the high-impedance state, and is also capable of switching the MTU0 pins, MTU complementary PWM output pins (MTU3, MTU4 pins or MTU6, MTU7 pins), GPT2, and GPT3 pins by register settings.
POE11#	Input	Request signal to switch the GPT2 and GPT3 pins to the high-impedance state, and is also capable of switching the MTU0 pins, MTU complementary PWM output pins (MTU3, MTU4 pins or MTU6, MTU7 pins), GPT0, and GPT1 pins by register settings.

Table 25.3 shows output-level comparisons with pin combinations.

Table 25.3 Pin Combinations

Pin Combination	I/O	Description
MTIOC3B and MTIOC3D	Output	The MTU complementary PWM output pins (MTU3 and MTU4 pins) set in the M3SELR, M4SELR1, and M4SELR2 registers are switched to the high-impedance state when two pins of the set simultaneously output the active level (low level when the MTUn.TOCR1A.OLSP bit is 0 or high level when the OLSP bit is 1 while the OLSEN bit in the ALR1 register is 0 and the MTUn.TOCR1A.TOCS bit is 0, low level when the OLS1P, OLS1N, OLS2P, OLS2N, OLS3P, and OLS3N bits in the MTUn.TOCR2A register are 0 or high level when these bits are 1 while the OLSEN bit in the ALR1 register is 0 and the MTUn.TOCR1A.TOCS bit is 1, or low level when the OLSG0A, OLSG0B, OLSG1A, OLSG1B, OLSG2A, and OLSG2B bits in the ALR1 register are 0 and high level when these bits are 1 while the OLSEN bit in the ALR1 register is 1) for at least 1 cycle of the peripheral module clock (PCLK). Pin combinations for output comparison can be selected by registers of POE.
MTIOC4A and MTIOC4C	Output	
MTIOC4B and MTIOC4D	Output	
MTIOC6B and MTIOC6D	Output	The MTU complementary PWM output pins (MTU6 and MTU7 pins) are switched to the high-impedance state when two pins of the set simultaneously output the active level (low level when the MTUn.TOCR1B.OLSP bit is 0 or high level when the OLSP bit is 1 while the MTUn.TOCR1B.TOCS bit is 0, or low level when the OLS1P, OLS1N, OLS2P, OLS2N, OLS3P, and OLS3N bits in the MTUn.TOCR2B register are 0 or high level when these bits are 1 while the MTUn.TOCR1B.TOCS bit is 1) for at least 1 cycle of the peripheral module clock (PCLK). Pin combinations for output comparison can be selected by registers of POE.
MTIOC7A and MTIOC7C	Output	
MTIOC7B and MTIOC7D	Output	
GTIOC0A and GTIOC0B	Output	The GPT output pins (GPT0 to GPT2 pins) set in the G0SELR, G1SELR, and G2SELR registers are switched to the high-impedance state when two pins of the set simultaneously output the active level (low level when the OLSG0A, OLSG0B, OLSG1A, OLSG1B, OLSG2A, and OLSG2B bits in the ALR1 register are 0 or high level when these bits are 1) for at least 1 cycle of the peripheral module clock (PCLK). Pin combinations for output comparison can be selected by registers of POE.
GTIOC1A and GTIOC1B	Output	
GTIOC2A and GTIOC2B	Output	

25.2 Register Descriptions

The POE registers are initialized by a reset.

25.2.1 Input Level Control/Status Register 1 (ICSR1)

Address(es): 0008 C4C0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	POE0F	—	—	—	PIE1	—	—	—	—	—	—	POE0M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	POE0M[1:0]	POE0 Mode Select	b1 b0 0 0: Accepts a request on the falling edge of POE0# pin input. 0 1: Accepts a request when POE0# pin input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE0# pin input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE0# pin input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PIE1	Port Interrupt Enable 1	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE0F	POE0 Flag	0: Indicates that a high-impedance request has not been input to the POE0# pin. 1: Indicates that a high-impedance request has been input to the POE0# pin.	R/(W)*2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR1 register selects the POE0# pin input modes, controls the enable/disable of interrupts, and indicates status.

POE0M[1:0] Bits (POE0 Mode Select)

These bits select the input mode of the POE0# pin.

PIE1 Bit (Port Interrupt Enable 1)

This bit enables or disables interrupt requests when the POE0F flag is set to 1.

POE0F Flag (POE0 Flag)

This flag indicates that a high-impedance request has been input to the POE0# pin.

[Setting condition]

- When the input set by the POE0M[1:0] bits occurs at the POE0# pin

[Clearing condition]

- By writing 0 to the POE0F flag after reading POE0F = 1
  When low-level sampling is set by the POE0M[1:0] bits, the high level needs to be input to the POE0# pin to write 0 to this flag.
  For details, refer to section 25.3.7, Recover from High-Impedance State.

25.2.2 Input Level Control/Status Register 2 (ICSR2)

Address(es): 0008 C4C4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	POE4F	—	—	—	PIE2	—	—	—	—	—	—	POE4M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	POE4M[1:0]	POE4 Mode Select	b1 b0 0 0: Accepts a request on the falling edge of POE4# pin input. 0 1: Accepts a request when POE4# pin input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE4# pin input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE4# pin input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PIE2	Port Interrupt Enable 2	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE4F	POE4 Flag	0: Indicates that a high-impedance request has not been input to the POE4# pin. 1: Indicates that a high-impedance request has been input to the POE4# pin.	R/(W) *2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR2 register selects the POE4# pin input mode, controls the enable/disable of interrupts, and indicates status.

POE4M[1:0] Bits (POE4 Mode Select)

These bits select the input mode of the POE4# pin.

PIE2 Bit (Port Interrupt Enable 2)

This bit enables or disables interrupt requests when the POE4F flag is set to 1.

POE4F Flag (POE4 Flag)

This flag indicates that a high-impedance request has been input to the POE4# pin.
[Setting condition]

- When the input set by POE4M[1:0] occurs at the POE4# pin

[Clearing condition]

- By writing 0 to POE4F after reading POE4F = 1
  When low-level sampling is set by the POE4M[1:0] bits, the high level needs to be input to the POE4# pin to write 0 to this flag.
  For details, refer to section 25.3.7, Recover from High-Impedance State.

25.2.3 Input Level Control/Status Register 3 (ICSR3)

Address(es): 0008 C4C8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	POE8F	—	—	POE8E	PIE3	—	—	—	—	—	—	POE8M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	POE8M[1:0]	POE8 Mode Select	b1 b0 0 0: Accepts a request on the falling edge of POE8# pin input. 0 1: Accepts a request when POE8# pin input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE8# pin input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE8# pin input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PIE3	Port Interrupt Enable 3	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	POE8E	POE8 High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE8F	POE8 Flag	0: Indicates that a high-impedance request has not been input to the POE8# pin. 1: Indicates that a high-impedance request has been input to the POE8# pin.	R/(W)*2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR3 register selects the POE8# pin input mode, controls the enable/disable of interrupts, and indicates status.

POE8M[1:0] Bits (POE8 Mode Select)

These bits select the input mode of the POE8# pin.

PIE3 Bit (Port Interrupt Enable 3)

This bit enables or disables interrupt requests when the POE8F flag is set to 1.

POE8E Bit (POE8 High-Impedance Enable)

This bit specifies whether to switch the corresponding pin to the high-impedance state when the POE8F flag is set to 1.

POE8F Flag (POE8 Flag)

This flag indicates that a high-impedance request has been input to the POE8# pin.

[Setting condition]

- When the input set by the POE8M[1:0] bits occurs at the POE8# pin

[Clearing condition]

- By writing 0 to the POE8F flag after reading POE8F = 1
  When low-level sampling is set by the POE8M[1:0] bits, the high level needs to be input to the POE8# pin to write 0 to this flag.
  For details, refer to section 25.3.7, Recover from High-Impedance State.

25.2.4 Input Level Control/Status Register 4 (ICSR4)

Address(es): 0008 C4D6h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	POE10F	—	—	POE10E	PIE4	—	—	—	—	—	—	POE10M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	POE10M[1:0]	POE10 Mode Select	b1 b0 0 0: Accepts a request on the falling edge of POE10# pin input. 0 1: Accepts a request when POE10# pin input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE10# pin input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE10# pin input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PIE4	Port Interrupt Enable 4	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	POE10E	POE10 High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE10F	POE10 Flag	0: Indicates that a high-impedance request has not been input to the POE10# pin. 1: Indicates that a high-impedance request has been input to the POE10# pin.	R/(W)*2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR4 register selects the POE10# pin input mode, controls the enable/disable of interrupts, and indicates status.

POE10M[1:0] Bits (POE10 Mode Select)

These bits select the input mode of the POE10# pin.

PIE4 Bit (Port Interrupt Enable 4)

This bit enables or disables interrupt requests when the POE10F flag is set to 1.

POE10E Bit (POE10 High-Impedance Enable)

This bit specifies whether to switch the corresponding pin to the high-impedance state when the POE10F flag is set to 1.

POE10F Flag (POE10 Flag)

This flag indicates that a high-impedance request has been input to the POE10# pin.
[Setting condition]

- When the input set by the POE10M[1:0] bits occurs at the POE10# pin

[Clearing condition]

- By writing 0 to the POE10F flag after reading POE10F = 1
  When low-level sampling is set by the POE10M[1:0] bits, the high level needs to be input to the POE10# pin to write 0 to this flag.
  For details, refer to section 25.3.7, Recover from High-Impedance State.

25.2.5 Input Level Control/Status Register 5 (ICSR5)

Address(es): 0008 C4D8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	POE11F	—	—	POE11E	PIE5	—	—	—	—	—	—	POE11M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	
b1, b0	POE11M[1:0]	POE11 Mode Select	b1 b0 0 0: Accepts a request on the falling edge of POE11# pin input. 0 1: Accepts a request when POE11# pin input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE11# pin input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE11# pin input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PIE5	Port Interrupt Enable 5	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	POE11E	POE11 High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE11F	POE11 Flag	0: Indicates that a high-impedance request has not been input to the POE11# pin. 1: Indicates that a high-impedance request has been input to the POE11# pin.	R/(W)*2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR5 register selects the POE11# pin input mode, controls the enable/disable of interrupts, and indicates status.

POE11M[1:0] Bits (POE11 Mode Select)

These bits select the input mode of the POE11# pin.

PIE5 Bit (Port Interrupt Enable 5)

This bit enables or disables interrupt requests when the POE11F flag is set to 1.

POE11E Bit (POE11 High-Impedance Enable)

This bit specifies whether to switch the corresponding pin to the high-impedance state when the POE11F flag is set to 1.

POE11F Flag (POE11 Flag)

This flag indicates that a high-impedance request has been input to the POE11# pin.

[Setting condition]

- When the input set by the POE11M[1:0] bits occurs at the POE11# pin

[Clearing condition]

- By writing 0 to the POE11F flag after reading POE11F = 1
  When low-level sampling is set by the POE11M[1:0] bits, the high level needs to be input to the POE11# pin to write 0 to this flag.
  For details, refer to section 25.3.7, Recover from High-Impedance State.

25.2.6 Input Level Control/Status Register 6 (ICSR6)

Address(es): 0008 C4DCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OSTSTF	—	—	OSTSTE	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	OSTSTE	OSTST High-Impedance Enable	0: Does not switch the MTU complementary PWM output pins, MTU0 pins, or GPT pins to high-impedance state. 1: Switch the MTU complementary PWM output pins, MTU0 pins, and GPT pins to high-impedance state.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	OSTSTF	OSTST High-Impedance Flag	0: Indicates that an oscillation stop high-impedance request has not been generated. 1: Indicates that an oscillation stop high-impedance request has been generated.	R/W*2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR6 register controls the oscillation stop high-impedance and indicates status.

OSTSTE Bit (OSTST High-Impedance Enable)

This bit enables/disables the MTU complementary PWM output pins, MTU0 pins, and GPT pins to switch to the high-impedance state when oscillation stop is detected.

OSTSTF Flag (OSTST High-Impedance Flag)

This flag indicates that an oscillation stop high-impedance request has been generated.
When oscillation stop is detected, this flag is set to 1. To clear this flag, wait for at least 10 cycles of PCLKB after this flag becomes 1 and write 0 to this flag while the OSTDSR.OSTDF flag is 0. Writing 0 to this flag while the OSTDSR.OSTDF flag is 1 cannot clear this flag. After clearing this flag, confirm that the flag has actually been modified to 0.
[Setting condition]

- When oscillation stop is detected

[Clearing condition]

- By writing 0 to the OSTSTF flag after reading OSTSTF = 1

25.2.7 Output Level Control/Status Register 1 (OCSR1)

Address(es): 0008 C4C2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OSF1	—	—	—	—	—	OCE1	OIE1	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	OIE1	Output Short Interrupt Enable 1	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	OCE1	Output Short High-Impedance Enable 1	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	OSF1	Output Short Flag 1	0: Indicates that outputs have not simultaneously become an active level. 1: Indicates that outputs have simultaneously become an active level.	R/(W)*2

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The OCSR1 register controls the enable/disable of output-level comparison and interrupts, and indicates status.

OIE1 Bit (Output Short Interrupt Enable 1)

This bit enables or disables interrupt requests when the OSF1 flag is set to 1.

OCE1 Bit (Output Short High-Impedance Enable 1)

This bit specifies whether to switch the pins to the high-impedance state when the OSF1 flag is set to 1.

OSF1 Flag (Output Short Flag 1)

This flag indicates that any one of the three pairs of two-phase output pins for MTU complementary PWM output (MTU3 and MTU4) or GPT output (GPT0 to GPT2) has simultaneously become at the active level. For setting the active level, refer to section 25.2.9, Active Level Setting Register 1 (ALR1).

[Setting condition]

- When any one of the three pairs of two-phase outputs has simultaneously become an active level*1

[Clearing condition]

- By writing 0 to the OSF1 flag after reading OSF1 = 1
  To write 0 to this flag, the inactive level needs to be output from MTU complementary PWM output pins or GPT output pins.
  For details, refer to section 25.3.7, Recover from High-Impedance State.

Note 1. The setting condition is judged only by the level of the pin regardless the setting of the MPC.PmnPFS register.

25.2.8 Output Level Control/Status Register 2 (OCSR2)

Address(es): 0008 C4C6h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OSF2	—	—	—	—	—	OCE2	OIE2	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	OIE2	Output Short Interrupt Enable 2	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	OCE2	Output Short High-Impedance Enable 2	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	OSF2	Output Short Flag 2	0: Indicates that outputs have not simultaneously become an active level. 1: Indicates that outputs have simultaneously become an active level.	R/(W)*2

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The OCSR2 register controls the enable/disable of output-level comparison and interrupts, and indicates status.

OIE2 Bit (Output Short Interrupt Enable 2)

This bit enables or disables interrupt requests when the OSF2 flag is set to 1.

OCE2 Bit (Output Short High-Impedance Enable 2)

This bit specifies whether to switch the pins to the high-impedance state when the OSF2 flag is set to 1.

OSF2 Flag (Output Short Flag 2)

This flag indicates that any one of the three pairs of two-phase output pins for MTU complementary PWM output (MTU6 and MTU7) has simultaneously become an active level. For setting the active level, refer to section 24, Multi-Function Timer Pulse Unit (MTU3a).

[Setting condition]

- When any one of the three pairs of two-phase outputs has simultaneously become an active level*1

[Clearing condition]

- By writing 0 to the OSF2 flag after reading OSF2 = 1
  To write 0 to this flag, the inactive level needs to be output from MTU complementary PWM output pins.
  For details, refer to section 25.3.7, Recover from High-Impedance State.

Note 1. The setting condition is judged only by the level of the pin regardless the setting of the MPC.PmnPFS register.

25.2.9 Active Level Setting Register 1 (ALR1)

Address(es): 0008 C4DAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	OLSEN	—	OLSG2B	OLSG2A	OLSG1B	OLSG1A	OLSG0B	OLSG0A
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLSG0A	MTIOC3B/GTIOC0A Active Level Setting	0: Active low 1: Active high	R/W*1
b1	OLSG0B	MTIOC3D/GTIOC0B Active Level Setting	0: Active low 1: Active high	R/W*1
b2	OLSG1A	MTIOC4A/GTIOC1A Active Level Setting	0: Active low 1: Active high	R/W*1
b3	OLSG1B	MTIOC4C/GTIOC1B Active Level Setting	0: Active low 1: Active high	R/W*1
b4	OLSG2A	MTIOC4B/GTIOC2A Active Level Setting	0: Active low 1: Active high	R/W*1
b5	OLSG2B	MTIOC4D/GTIOC2B Active Level Setting	0: Active low 1: Active high	R/W*1
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	OLSEN	Active Level Setting Enable	0: Disabled 1: Enabled	R/W*1
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The ALR1 register specifies the active levels of the MTU and GPT outputs selected in the MGSELR register for detection of short circuits of those outputs as reflected in the OCSR1 register.

OLSG0A Bit (MTIOC3B/GTIOC0A Active Level Setting)

This bit sets the active level of the MTIOC3B and GTIOC0A outputs. Specifically, setting the OLSG0A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG0B Bit (MTIOC3D/GTIOC0B Active Level Setting)

This bit sets the active level of the MTIOC3D and GTIOC0B outputs. Specifically, setting the OLSG0B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG1A Bit (MTIOC4A/GTIOC1A Active Level Setting)

This bit sets the active level of the MTIOC4A and GTIOC1A outputs. Specifically, setting the OLSG1A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG1B Bit (MTIOC4C/GTIOC1B Active Level Setting)

This bit sets the active level of the MTIOC4C and GTIOC1B outputs. Specifically, setting the OLSG1B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG2A Bit (MTIOC4B/GTIOC2A Active Level Setting)

This bit sets the active level of the MTIOC4B and GTIOC2A outputs. Specifically, setting the OLSG2A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG2B Bit (MTIOC4D/GTIOC2B Active Level Setting)

This bit sets the active level of the MTIOC4D and GTIOC2B outputs. Specifically, setting the OLSG2B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSEN Bit (Active Level Setting Enable)

This bit enables or disables of the active-level settings in the OLSGnm bits (n = 0 to 2; m = A, B). Clearing the OLSEN bit to 0 disables the OLSGnm bits, in which case the active levels of the MTU output are determined by the MTU.TOCR1j and MTU.TOCR2j registers (j = A, B). Setting the OLSEN bit to 1 enables the OLSGnm bits, in which case the active levels of the MTU output are as selected by the OLSGnm bits in this register.

Active levels for the GPT output can only be set when the OLSEN bit is 1. When output short-circuit detection is to be used on the GPT outputs, set the OLSEN bit to 1 and then use the OLSGnm bits to set the active levels for the GPT outputs.

25.2.10 Software Port Output Enable Register (SPOER)

Address(es): 0008 C4CAh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	GPT23 HIZ	GPT01 HIZ	MTUC H0HIZ	MTUC H67HIZ	MTUC H34HIZ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MTUCH34HIZ	MTU3 and MTU4 or GPT0 to GPT2 Output High-Impedance Enable	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W
b1	MTUCH67HIZ	MTU6 and MTU7 Output High-Impedance Enable	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W
b2	MTUCH0HIZ	MTU0 Output High-Impedance Enable	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W
b3	GPT01HIZ	GPT0 and GPT1 Output High-Impedance Enable	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W
b4	GPT23HIZ	GPT2 and GPT3 Output High-Impedance Enable	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPOER register is used to switch the pins to the high-impedance state.

MTUCH34HIZ Bit (MTU3 and MTU4 or GPT0 to GPT2 Output High-Impedance Enable)

This bit specifies whether to switch the MTU complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D) or the GPT output pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B, GTIOC2A, and GTIOC2B) to the high-impedance state.

[Setting condition]

- By writing 1 to the MTUCH34HIZ bit

[Clearing conditions]

- Reset
- By writing 0 to the MTUCH34HIZ bit after reading MTUCH34HIZ = 1

MTUCH67HIZ Bit (MTU6 and MTU7 Output High-Impedance Enable)

This bit specifies whether to switch the MTU complementary PWM output pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC7C, and MTIOC7D) to the high-impedance state.

[Setting condition]

- By writing 1 to the MTUCH67HIZ bit

[Clearing conditions]

- Reset
- By writing 0 to the MTUCH67HIZ bit after reading MTUCH67HIZ = 1

MTUCH0HIZ Bit (MTU0 Output High-Impedance Enable)

This bit specifies whether to switch the MTU0 pins to the high-impedance state.

[Setting condition]

- By writing 1 to the MTUCH0HIZ bit

[Clearing conditions]

- Reset

- By writing 0 to the MTUCH0HIZ bit after reading MTUCH0HIZ = 1

GPT01HIZ Bit (GPT0 and GPT1 Output High-Impedance Enable)

This bit specifies whether to switch the GPT0 and GPT1 pins (GTIOC0A, GTIOC0B, GTIOC1A, and GTIOC1B) to the high-impedance state.
[Setting condition]

- By writing 1 to the GPT01HIZ bit

[Clearing conditions]

- Reset
- By writing 0 to the GPT01HIZ bit after reading GPT01HIZ = 1

GPT23HIZ Bit (GPT2 and GPT3 Output High-Impedance Enable)

This bit specifies whether to switch the GPT2 and GPT3 pins (GTIOC2A, GTIOC2B, GTIOC3A, and GTIOC3B) to the high-impedance state.
[Setting condition]

- By writing 1 to the GPT23HIZ bit

[Clearing conditions]

- Reset
- By writing 0 to the GPT23HIZ bit after reading GPT23HIZ = 1

25.2.11 Port Output Enable Control Register 1 (POECR1)

Address(es): 0008 C4CBh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	MTU0DZE	MTU0CZE	MTU0BZE	MTU0AZE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MTU0AZE	MTIOC0A High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b1	MTU0BZE	MTIOC0B High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b2	MTU0CZE	MTIOC0C High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b3	MTU0DZE	MTIOC0D High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR1 register controls high-impedance state of the MTU0 pins.

MTU0AZE Bit (MTIOC0A High-Impedance Enable)

This bit specifies whether to switch the MTIOC0A output to the high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR5 register, the ICSRn.POEmF flag (n = 1, 2, 4, 5; m = 0, 4, 10, 11), is set to 1.

MTU0BZE Bit (MTIOC0B High-Impedance Enable)

This bit specifies whether to switch the MTIOC0B output to the high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR5 register, the ICSRn.POEmF flag (n = 1, 2, 4, 5; m = 0, 4, 10, 11), is set to 1.

MTU0CZE Bit (MTIOC0C High-Impedance Enable)

This bit specifies whether to switch the MTIOC0C output to the high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR5 register, the ICSRn.POEmF flag (n = 1, 2, 4, 5; m = 0, 4, 10, 11), is set to 1.

MTU0DZE Bit (MTIOC0D High-Impedance Enable)

This bit specifies whether to switch the MTIOC0D output to the high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR5 register, the ICSRn.POEmF flag (n = 1, 2, 4, 5; m = 0, 4, 10, 11), is set to 1.

25.2.12 Port Output Enable Control Register 2 (POECR2)

Address(es): 0008 C4CCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MTU3BDZE	MTU4ACZE	MTU4BDZE	—	—	—	—	—	MTU6BDZE	MTU7ACZE	MTU7BDZE
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	MTU7BDZE	MTIOC7B/7D High-Impedance Enable *2	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b1	MTU7ACZE	MTIOC7A/7C High-Impedance Enable *2	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b2	MTU6BDZE	MTIOC6B/6D High-Impedance Enable *2	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	MTU4BDZE	MTIOC4B/4D High-Impedance Enable	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b9	MTU4ACZE	MTIOC4A/4C High-Impedance Enable	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b10	MTU3BDZE	MTIOC3B/3D High-Impedance Enable	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Set this bit to 0 when MTU6 or MTU7 is not used.

The POECR2 register controls high-impedance state of the MTU complementary PWM output pins (MTU3, MTU4, MTU6, and MTU7 pins) or the GPT output pins (GPT0 to GPT2 pins).

MTU7BDZE Bit (MTIOC7B/7D High-Impedance Enable)

This bit specifies whether to switch the MTIOC7B output and MTIOC7D output to the high-impedance state when any one of the OCSR2.OSF2 flag, ICSR2.POE4F flag, SPOER.MTUCH67HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 1, 3 to 5; m = 0, 8, 10, 11), is set to 1.

MTU7ACZE Bit (MTIOC7A/7C High-Impedance Enable)

This bit specifies whether to switch the MTIOC7A output and MTIOC7C output to the high-impedance state when any one of the OCSR2.OSF2 flag, ICSR2.POE4F flag, SPOER.MTUCH67HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 1, 3 to 5; m = 0, 8, 10, 11), is set to 1.

MTU6BDZE Bit (MTIOC6B/6D High-Impedance Enable)

This bit specifies whether to switch the MTIOC6B output and MTIOC6D output to the high-impedance state when any one of the OCSR2.OSF2 flag, ICSR2.POE4F flag, SPOER.MTUCH67HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 1, 3 to 5; m = 0, 8, 10, 11), is set to 1.

MTU4BDZE Bit (MTIOC4B/4D High-Impedance Enable)

This bit specifies whether to switch the MTIOC4B/GTIOC2A output and MTIOC4D/GTIOC2B output to the high-impedance state when any one of the OCSR1.OSF1 flag, ICSR1.POE0F flag, SPOER.MTUCH34HIZ bit,

ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 2 to 5; m = 4, 8, 10, 11), is set to 1.

MTU4ACZE Bit (MTIOC4A/4C High-Impedance Enable)

This bit specifies whether to switch the MTIOC4A/GTIOC1A output and MTIOC4C/GTIOC1B output to the high-impedance state when any one of the OCSR1.OSF1 flag, ICSR1.POE0F flag, SPOER.MTUCH34HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 2 to 5; m = 4, 8, 10, 11), is set to 1.

MTU3BDZE Bit (MTIOC3B/3D High-Impedance Enable)

This bit specifies whether to switch the MTIOC3B/GTIOC0A output and MTIOC3D/GTIOC0B output to the high-impedance state when any one of the OCSR1.OSF1 flag, ICSR1.POE0F flag, SPOER.MTUCH34HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 2 to 5; m = 4, 8, 10, 11), is set to 1.

25.2.13 Port Output Enable Control Register 3 (POECR3)

Address(es): 0008 C4CEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	GPT3ABZE	GPT2ABZE	—	—	—	—	—	—	GPT1ABZE	GPT0ABZE
Value after reset:	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	GPT0ABZE	GTIOC0A/0B High-Impedance Enable	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b1	GPT1ABZE	GTIOC1A/1B High-Impedance Enable	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	GPT2ABZE	GTIOC2A/2B High-Impedance Enable	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b9	GPT3ABZE	GTIOC3A/3B High-Impedance Enable	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR3 register controls high-impedance state of the GPT pins (GPT0 to GPT3).

GPT0ABZE Bit (GTIOC0A/0B High-Impedance Enable)

This bit specifies whether to switch the GTIOC0A and GTIOC0B outputs to the high-impedance state when any one of the ICSR4.POE10F flag, SPOER.GPT01HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR6 register, the ICSRn.POEmF flag (n = 1 to 3, 5; m = 0, 4, 8, 11), is set to 1.

GPT1ABZE Bit (GTIOC1A/1B High-Impedance Enable)

This bit specifies whether to switch the GTIOC1A and GTIOC1B outputs to the high-impedance state when any one of the ICSR4.POE10F flag, SPOER.GPT01HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR6 register, the ICSRn.POEmF flag (n = 1 to 3, 5; m = 0, 4, 8, 11), is set to 1.

GPT2ABZE Bit (GTIOC2A/2B High-Impedance Enable)

This bit specifies whether to switch the GTIOC2A and GTIOC2B outputs to the high-impedance state when any one of the ICSR5.POE11F flag, SPOER.GPT23HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR6 register, the ICSRn.POEmF flag (n = 1 to 4; m = 0, 4, 8, 10), is set to 1.

GPT3ABZE Bit (GTIOC3A/3B High-Impedance Enable)

This bit specifies whether to switch the GTIOC3A and GTIOC3B outputs to the high-impedance state when any one of the ICSR5.POE11F flag, SPOER.GPT23HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR6 register, the ICSRn.POEmF flag (n = 1 to 4; m = 0, 4, 8, 10), is set to 1.

25.2.14 Port Output Enable Control Register 4 (POECR4)

Address(es): 0008 C4D0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	IC5ADDMT67ZE	IC4ADDMT67ZE	IC3ADDMT67ZE	—	IC1ADDMT67ZE	—	—	—	IC5ADDMT34ZE	IC4ADDMT34ZE	IC3ADDMT34ZE	IC2ADDMT34ZE	—	—
Value after reset:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b2	IC2ADDMT34ZE	MTU3 and MTU4 High-Impedance POE4F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W*1
b3	IC3ADDMT34ZE	MTU3 and MTU4 High-Impedance POE8F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W*1
b4	IC4ADDMT34ZE	MTU3 and MTU4 High-Impedance POE10F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W*1
b5	IC5ADDMT34ZE	MTU3 and MTU4 High-Impedance POE11F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W*1
b8 to b6	—	Reserved	These bits are read as 0. The write value should be 0	R/W
b9	IC1ADDMT67ZE	MTU6 and MTU7 High-Impedance POE0F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W*1
b10	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b11	IC3ADDMT67ZE	MTU6 and MTU7 High-Impedance POE8F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W*1
b12	IC4ADDMT67ZE	MTU6 and MTU7 High-Impedance POE10F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W*1
b13	IC5ADDMT67ZE	MTU6 and MTU7 High-Impedance POE11F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W*1
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR4 register is used to extend the conditions of the high-impedance control for the MTU complementary PWM output pins (MTU3, MTU4, MTU6, and MTU7) and the GPT output pins (GPT0 to GPT2).

IC2ADDMT34ZE Bit (MTU3 and MTU4 High-Impedance POE4F Add)

Adds the ICSR2.POE4F flag to the high-impedance control conditions for the MTU3, MTU4, and GPT0 to GPT2 pins (MTIOC3B/MTIOC3D/GTIOC0A/GTIOC0B/MTIOC4A/MTIOC4C/GTIOC1A/GTIOC1B/MTIOC4B/MTIOC4D/GTIOC2A/GTIOC2B).

IC3ADDMT34ZE Bit (MTU3 and MTU4 High-Impedance POE8F Add)

Adds the ICSR3.POE8F flag to the high-impedance control conditions for the MTU3, MTU4, and GPT0 to GPT2 pins (MTIOC3B/MTIOC3D/GTIOC0A/GTIOC0B/MTIOC4A/MTIOC4C/GTIOC1A/GTIOC1B/MTIOC4B/MTIOC4D/GTIOC2A/GTIOC2B).

IC4ADDMT34ZE Bit (MTU3 and MTU4 High-Impedance POE10F Add)

Adds the ICSR4.POE10F flag to the high-impedance control conditions for the MTU3,MTU4, and GPT0 to GPT2 pins (MTIOC3B/MTIOC3D/GTIOC0A/GTIOC0B/MTIOC4A/MTIOC4C/GTIOC1A/GTIOC1B/MTIOC4B/MTIOC4D/GTIOC2A/GTIOC2B).

IC5ADDMT34ZE Bit (MTU3 and MTU4 High-Impedance POE11F Add)

Adds the ICSR5.POE11F flag to the high-impedance control conditions for the MTU3, MTU4, and GPT0 to GPT2 pins (MTIOC3B/MTIOC3D/GTIOC0A/GTIOC0B/MTIOC4A/MTIOC4C/GTIOC1A/GTIOC1B/MTIOC4B/MTIOC4D/GTIOC2A/GTIOC2B).

IC1ADDMT67ZE Bit (MTU6 and MTU7 High-Impedance POE0F Add)

Adds the ICSR1.POE0F flag to the high-impedance control conditions for the MTU6 and MTU7 pins (MTIOC6B/MTIOC6D/MTIOC7A/MTIOC7C/MTIOC7B/MTIOC7D).

IC3ADDMT67ZE Bit (MTU6 and MTU7 High-Impedance POE8F Add)

Adds the ICSR3.POE8F flag to the high-impedance control conditions for the MTU6 and MTU7 pins (MTIOC6B/MTIOC6D/MTIOC7A/MTIOC7C/MTIOC7B/MTIOC7D).

IC4ADDMT67ZE Bit (MTU6 and MTU7 High-Impedance POE10F Add)

Adds the ICSR4.POE10F flag to the high-impedance control conditions for the MTU6 and MTU7 pins (MTIOC6B/MTIOC6D/MTIOC7A/MTIOC7C/MTIOC7B/MTIOC7D).

IC5ADDMT67ZE Bit (MTU6 and MTU7 High-Impedance POE11F Add)

Adds the ICSR5.POE11F flag to the high-impedance control conditions for the MTU6 and MTU7 pins (MTIOC6B/MTIOC6D/MTIOC7A/MTIOC7C/MTIOC7B/MTIOC7D).

25.2.15 Port Output Enable Control Register 5 (POECR5)

Address(es): 0008 C4D2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	IC5ADDMT0ZE	IC4ADDMT0ZE	—	IC2ADDMT0ZE	IC1ADDMT0ZE	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	IC1ADDMT0ZE	MTU0 High-Impedance POE0F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W*1
b2	IC2ADDMT0ZE	MTU0 High-Impedance POE4F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W*1
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W*1
b4	IC4ADDMT0ZE	MTU0 High-Impedance POE10F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W*1
b5	IC5ADDMT0ZE	MTU0 High-Impedance POE11F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W*1
b15 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR5 register is used to extend the control conditions of the high-impedance for the MTU0 pins.

IC1ADDMT0ZE Bit (MTU0 High-Impedance POE0F Add)

Adds the ICSR1.POE0F flag to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC2ADDMT0ZE Bit (MTU0 High-Impedance POE4F Add)

Adds the ICSR2.POE4F flag to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC4ADDMT0ZE Bit (MTU0 High-Impedance POE10F Add)

Adds the ICSR4.POE10F flag to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC5ADDMT0ZE Bit (MTU0 High-Impedance POE11F Add)

Adds the ICSR5.POE11F flag to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

25.2.16 Port Output Enable Control Register 6 (POECR6)

Address(es): 0008 C4D4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	IC4ADDGPT23ZE	IC3ADDGPT23ZE	IC2ADDGPT23ZE	IC1ADDGPT23ZE	—	—	—	IC5ADDGPT01ZE	—	IC3ADDGPT01ZE	IC2ADDGPT01ZE	IC1ADDGPT01ZE	—
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	IC1ADDGPT01ZE	GPT0 and GPT1 High-Impedance POE0F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W *1
b2	IC2ADDGPT01ZE	GPT0 and GPT1 High-Impedance POE4F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W *1
b3	IC3ADDGPT01ZE	GPT0 and GPT1 High-Impedance POE8F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W *1
b4	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b5	IC5ADDGPT01ZE	GPT0 and GPT1 High-Impedance POE11F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W *1
b8 to b6	—	Reserved	These bits are read as 0. The write value should be 0	R/W
b9	IC1ADDGPT23ZE	GPT2 and GPT3 High-Impedance POE0F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W *1
b10	IC2ADDGPT23ZE	GPT2 and GPT3 High-Impedance POE4F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W *1
b11	IC3ADDGPT23ZE	GPT2 and GPT3 High-Impedance POE8F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W *1
b12	IC4ADDGPT23ZE	GPT2 and GPT3 High-Impedance POE10F Add	0: Does not add the pins to the high-impedance control conditions. 1: Adds the pins to the high-impedance control conditions.	R/W *1
b13	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR6 register is used to extend the control conditions of the high-impedance for the GPT0 to GPT3 pins.

IC1ADDGPT01ZE Bit (GPT0 and GPT1 High-Impedance POE0F Add)

Adds the ICSR1.POE0F flag to the high-impedance control conditions for the GPT0 and GPT1 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B).

IC2ADDGPT01ZE Bit (GPT0 and GPT1 High-Impedance POE4F Add)

Adds the ICSR2.POE4F flag to the high-impedance control conditions for the GPT0 and GPT1 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B).

IC3ADDGPT01ZE Bit (GPT0 and GPT1 High-Impedance POE8F Add)

Adds the ICSR3.POE8F flag to the high-impedance control conditions for the GPT0 and GPT1 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B).

IC5ADDGPT01ZE Bit (GPT0 and GPT1 High-Impedance POE11F Add)

Adds the ICSR5.POE11F flag to the high-impedance control conditions for the GPT0 and GPT1 pins (GTIOC0A, GTIOC0B, GTIOC1A, GTIOC1B).

IC1ADDGPT23ZE Bit (GPT2 and GPT3 High-Impedance POE0F Add)

Adds the ICSR1.POE0F flag to the high-impedance control conditions for the GPT2 and GPT3 pins (GTIOC2A, GTIOC2B, GTIOC2B, GTIOC3A, GTIOC3B).

IC2ADDGPT23ZE Bit (GPT2 and GPT3 High-Impedance POE4F Add)

Adds the ICSR2.POE4F flag to the high-impedance control conditions for the GPT2 and GPT3 pins (GTIOC2A, GTIOC2B, GTIOC2B, GTIOC3A, GTIOC3B).

IC3ADDGPT23ZE Bit (GPT2 and GPT3 High-Impedance POE8F Add)

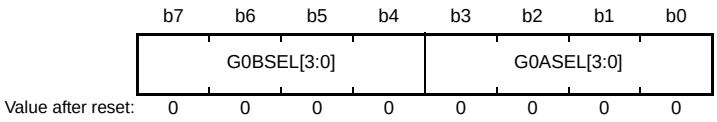
Adds the ICSR3.POE8F flag to the high-impedance control conditions for the GPT2 and GPT3 pins (GTIOC2A, GTIOC2B, GTIOC2B, GTIOC3A, GTIOC3B).

IC4ADDGPT23ZE Bit (GPT2 and GPT3 High-Impedance POE10F Add)

Adds the ICSR4.POE10F flag to the high-impedance control conditions for the GPT2 and GPT3 pins (GTIOC2A, GTIOC2B, GTIOC2B, GTIOC3A, GTIOC3B).

25.2.17 GPT0 Pin Select Register (G0SELR)

Address(es): 0008 C4E0h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	G0ASEL[3:0]	GPT0-A (GTIOC0A) Pin Select	b3 b0 0000: Controls the high-impedance state of PE5 when it is in use as the GTIOC0A pin. 0001: Controls the high-impedance state of P23 when it is in use as the GTIOC0A pin. 0010: Controls the high-impedance state of PA5 when it is in use as the GTIOC0A pin. 0011: Controls the high-impedance state of P83 when it is in use as the GTIOC0A pin.*2 0100: Controls the high-impedance state of PD3 when it is in use as the GTIOC0A pin. Settings other than above are prohibited.	R/W*1
b7 to b4	G0BSEL[3:0]	GPT0-B (GTIOC0B) Pin Select	b7 b4 0000: Controls the high-impedance state of PE2 when it is in use as the GTIOC0B pin. 0001: Controls the high-impedance state of P17 when it is in use as the GTIOC0B pin. 0010: Controls the high-impedance state of PA0 when it is in use as the GTIOC0B pin. 0011: Controls the high-impedance state of P81 when it is in use as the GTIOC0B pin.*2 0100: Controls the high-impedance state of PD2 when it is in use as the GTIOC0B pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.
Note 2. This is only selectable in 144-pin and 176-pin products.

The G0SELR register is an 8-bit readable/writable register that selects the target pins for high-impedance control on GPT0.

G0ASEL[3:0] Bits (GPT0-A (GTIOC0A) Pin Select)

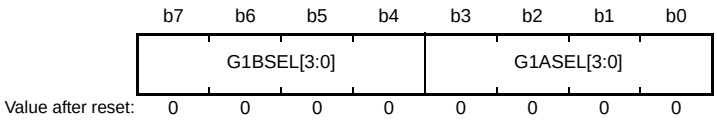
These bits select the target GTIOC0A pin for high-impedance control.

G0BSEL[3:0] Bits (GPT0-B (GTIOC0B) Pin Select)

These bits select the target GTIOC0B pin for high-impedance control.

25.2.18 GPT1 Pin Select Register (G1SELR)

Address(es): 0008 C4E1h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	G1ASEL[3:0]	GPT1-A (GTIOC1A) Pin Select	b3 b0 0000: Controls the high-impedance state of PE4 when it is in use as the GTIOC1A pin. 0001: Controls the high-impedance state of P22 when it is in use as the GTIOC1A pin. 0010: Controls the high-impedance state of PA2 when it is in use as the GTIOC1A pin. 0011: Controls the high-impedance state of PC5 when it is in use as the GTIOC1A pin. 0100: Controls the high-impedance state of PD1 when it is in use as the GTIOC1A pin. Settings other than above are prohibited.	R/W*1
b7 to b4	G1BSEL[3:0]	GPT1-B (GTIOC1B) Pin Select	b7 b4 0000: Controls the high-impedance state of PE1 when it is in use as the GTIOC1B pin. 0001: Controls the high-impedance state of P87 when it is in use as the GTIOC1B pin.*2 0010: Controls the high-impedance state of P67 when it is in use as the GTIOC1B pin.*2 0011: Controls the high-impedance state of PC3 when it is in use as the GTIOC1B pin. 0100: Controls the high-impedance state of PD0 when it is in use as the GTIOC1B pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.
Note 2. This is only selectable in 144-pin and 176-pin products.

The G1SELR register is an 8-bit readable/writable register that selects the target pins for high-impedance control on GPT1.

G1ASEL[3:0] Bits (GPT1-A (GTIOC1A) Pin Select)

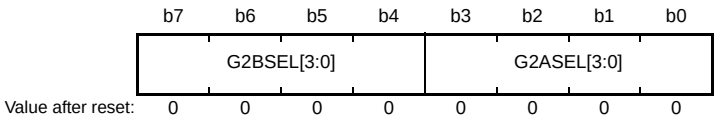
These bits select the target GTIOC1A pin for high-impedance control.

G1BSEL[3:0] Bits (GPT1-B (GTIOC1B) Pin Select)

These bits select the target GTIOC1B pin for high-impedance control.

25.2.19 GPT2 Pin Select Register (G2SELR)

Address(es): 0008 C4E2h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	G2ASEL[3:0]	GPT2-A (GTIOC2A) Pin Select	b3 b0 0000: Controls the high-impedance state of PE3 when it is in use as the GTIOC2A pin. 0001: Controls the high-impedance state of P21 when it is in use as the GTIOC2A pin. 0010: Controls the high-impedance state of PA1 when it is in use as the GTIOC2A pin. 0011: Controls the high-impedance state of P82 when it is in use as the GTIOC2A pin.*2 Settings other than above are prohibited.	R/W*1
b7 to b4	G2BSEL[3:0]	GPT2-B (GTIOC2B) Pin Select	b7 b4 0000: Controls the high-impedance state of PE0 when it is in use as the GTIOC2B pin. 0001: Controls the high-impedance state of P86 when it is in use as the GTIOC2B pin.*2 0010: Controls the high-impedance state of P66 when it is in use as the GTIOC2B pin.*2 0011: Controls the high-impedance state of PC2 when it is in use as the GTIOC2B pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.
Note 2. This is only selectable in 144-pin and 176-pin products.

The G2SELR register is an 8-bit readable/writable register that selects the target pins for high-impedance control on GPT2.

G2ASEL[3:0] Bits (GPT2-A (GTIOC2A) Pin Select)

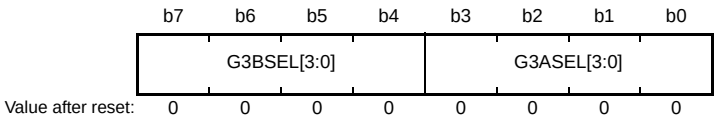
These bits select the target GTIOC2A pin for high-impedance control.

G2BSEL[3:0] Bits (GPT2-B (GTIOC2B) Pin Select)

These bits select the target GTIOC2B pin for high-impedance control.

25.2.20 GPT3 Pin Select Register (G3SELR)

Address(es): 0008 C4E3h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	G3ASEL[3:0]	GPT3-A (GTIOC3A) Pin Select	b3 b0 0000: Controls the high-impedance state of PC7 when it is in use as the GTIOC3A pin. 0100: Controls the high-impedance state of PE7 when it is in use as the GTIOC3A pin. Settings other than above are prohibited.	R/W*1
b7 to b4	G3BSEL[3:0]	GPT3-B (GTIOC3B) Pin Select	b7 b4 0000: Controls the high-impedance state of PC6 when it is in use as the GTIOC3B pin. 0100: Controls the high-impedance state of PE6 when it is in use as the GTIOC3B pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The G3SELR register is an 8-bit readable/writable register that selects the target pins for high-impedance control on GPT3.

G3ASEL[3:0] Bits (GPT3-A (GTIOC3A) Pin Select)

These bits select the target GTIOC3A pin for high-impedance control.

G3BSEL[3:0] Bits (GPT3-B (GTIOC3B) Pin Select)

These bits select the target GTIOC3B pin for high-impedance control.

25.2.21 MTU0 Pin Select Register 1 (M0SELR1)

Address(es): 0008 C4E4h

	b7	b6	b5	b4	b3	b2	b1	b0
	M0BSEL[3:0]				M0ASEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M0ASEL[3:0]	MTU0-A (MTIOC0A) Pin Select	b3 b0 0000: Controls the high-impedance state of P34 when it is in use as the MTIOC0A pin. 0010: Controls the high-impedance state of PB3 when it is in use as the MTIOC0A pin. Settings other than above are prohibited.	R/W*1
b7 to b4	M0BSEL[3:0]	MTU0-B (MTIOC0B) Pin Select	b7 b4 0000: Controls the high-impedance state of P13 when it is in use as the MTIOC0B pin. 0001: Controls the high-impedance state of P15 when it is in use as the MTIOC0B pin. 0010: Controls the high-impedance state of PA1 when it is in use as the MTIOC0B pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The M0SELR1 register is an 8-bit readable/writable register that selects the MTU0-A/B pins as targets for high-impedance control.

M0ASEL[3:0] Bits (MTU0-A (MTIOC0A) Pin Select)

These bits select the target MTIOC0A pin for high-impedance control.

M0BSEL[3:0] Bits (MTU0-B (MTIOC0B) Pin Select)

These bits select the target MTIOC0B pin for high-impedance control.

25.2.22 MTU0 Pin Select Register 2 (M0SELR2)

Address(es): 0008 C4E5h

	b7	b6	b5	b4	b3	b2	b1	b0
	M0DSEL[3:0]				M0CSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M0CSEL[3:0]	MTU0-C (MTIOC0C) Pin Select	b3 b0 0000: Controls the high-impedance state of P32 when it is in use as the MTIOC0C pin. 0010: Controls the high-impedance state of PB1 when it is in use as the MTIOC0C pin. Settings other than above are prohibited.	R/W*1
b7 to b4	M0DSEL[3:0]	MTU0-D (MTIOC0D) Pin Select	b7 b4 0000: Controls the high-impedance state of P33 when it is in use as the MTIOC0D pin. 0010: Controls the high-impedance state of PA3 when it is in use as the MTIOC0D pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.

The M0SELR2 register is an 8-bit readable/writable register that selects the MTU0-C/D pins as targets for high-impedance control.

M0CSEL[3:0] Bits (MTU0-C (MTIOC0C) Pin Select)

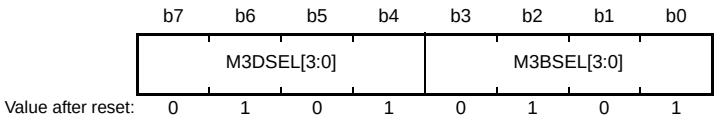
These bits select the target MTIOC0C pin for high-impedance control.

M0DSEL[3:0] Bits (MTU0-D (MTIOC0D) Pin Select)

These bits select the target MTIOC0D pin for high-impedance control.

25.2.23 MTU3 Pin Select Register (M3SELR)

Address(es): 0008 C4E6h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M3BSEL[3:0]	MTU3-B (MTIOC3B) Pin Select	b3 b0 0000: Controls the high-impedance state of PE1 when it is in use as the MTIOC3B pin. 0001: Controls the high-impedance state of P22 when it is in use as the MTIOC3B pin. 0010: Controls the high-impedance state of P80 when it is in use as the MTIOC3B pin.*2 0011: Controls the high-impedance state of PC5 when it is in use as the MTIOC3B pin. 0100: Controls the high-impedance state of PB7 when it is in use as the MTIOC3B pin. 0101: Controls the high-impedance state of P17 when it is in use as the MTIOC3B pin. Settings other than above are prohibited.	R/W*1
b7 to b4	M3DSEL[3:0]	MTU3-D (MTIOC3D) Pin Select	b7 b4 0000: Controls the high-impedance state of PE0 when it is in use as the MTIOC3D pin. 0001: Controls the high-impedance state of P23 when it is in use as the MTIOC3D pin. 0010: Controls the high-impedance state of PC4 when it is in use as the MTIOC3D pin.*2 0011: Controls the high-impedance state of P81 when it is in use as the MTIOC3D pin. 0100: Controls the high-impedance state of PB6 when it is in use as the MTIOC3D pin. 0101: Controls the high-impedance state of P16 when it is in use as the MTIOC3D pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.
Note 2. This is only selectable in 144-pin and 176-pin products.

The M3SELR register is an 8-bit readable/writable register that selects the MTU3-B/D pins as targets for high-impedance control.

M3BSEL[3:0] Bits (MTU3-B (MTIOC3B) Pin Select)

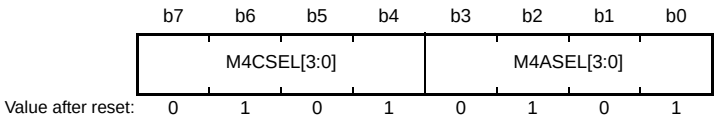
These bits select the target MTIOC3B pin for high-impedance control.

M3DSEL[3:0] Bits (MTU3-D (MTIOC3D) Pin Select)

These bits select the target MTIOC3D pin for high-impedance control.

25.2.24 MTU4 Pin Select Register 1 (M4SELR1)

Address(es): 0008 C4E7h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M4ASEL[3:0]	MTU4-A (MTIOC4A) Pin Select	b3 b0 0000: Controls the high-impedance state of PE2 when it is in use as the MTIOC4A pin. 0001: Controls the high-impedance state of P21 when it is in use as the MTIOC4A pin. 0010: Controls the high-impedance state of PB3 when it is in use as the MTIOC4A pin. 0011: Controls the high-impedance state of P82 when it is in use as the MTIOC4A pin.*2 0100: Controls the high-impedance state of PA0 when it is in use as the MTIOC4A pin. 0101: Controls the high-impedance state of P24 when it is in use as the MTIOC4A pin. Settings other than above are prohibited.	R/W*1
b7 to b4	M4CSEL[3:0]	MTU4-C (MTIOC4C) Pin Select	b7 b4 0000: Controls the high-impedance state of PE5 when it is in use as the MTIOC4C pin. 0001: Controls the high-impedance state of P87 when it is in use as the MTIOC4C pin.*2 0010: Controls the high-impedance state of PB1 when it is in use as the MTIOC4C pin. 0011: Controls the high-impedance state of P83 when it is in use as the MTIOC4C pin.*2 0100: Controls the high-impedance state of PE1 when it is in use as the MTIOC4C pin. 0101: Controls the high-impedance state of P25 when it is in use as the MTIOC4C pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.
Note 2. This is only selectable in 144-pin and 176-pin products.

The M4SELR1 register is an 8-bit readable/writable register that selects the MTU4-A/C pins as targets for high-impedance control.

M4ASEL[3:0] Bits (MTU4-A (MTIOC4A) Pin Select)

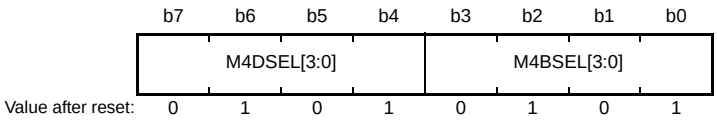
These bits select the target MTIOC4A pin for high-impedance control.

M4CSEL[3:0] Bits (MTU4-C (MTIOC4C) Pin Select)

These bits select the target MTIOC4C pin for high-impedance control.

25.2.25 MTU4 Pin Select Register 2 (M4SELR2)

Address(es): 0008 C4E8h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M4BSEL[3:0]	MTU4-B (MTIOC4B) Pin Select	b3 b0 0000: Controls the high-impedance state of PE3 when it is in use as the MTIOC4B pin. 0001: Controls the high-impedance state of P17 when it is in use as the MTIOC4B pin. 0010: Controls the high-impedance state of P54 when it is in use as the MTIOC4B pin.*3 0011: Controls the high-impedance state of PC2 when it is in use as the MTIOC4B pin. 0100: Controls the high-impedance state of PD1 when it is in use as the MTIOC4B pin. 0101: Controls the high-impedance state of P30 when it is in use as the MTIOC4B pin. Settings other than above are prohibited.	R/W*1
b7 to b4	M4DSEL[3:0]	MTU4-D (MTIOC4D) Pin Select	b7 b4 0000: Controls the high-impedance state of PE4 when it is in use as the MTIOC4D pin. 0001: Controls the high-impedance state of P86 when it is in use as the MTIOC4D pin.*2 0010: Controls the high-impedance state of P55 when it is in use as the MTIOC4D pin.*3 0011: Controls the high-impedance state of PC3 when it is in use as the MTIOC4D pin. 0100: Controls the high-impedance state of PD2 when it is in use as the MTIOC4D pin. 0101: Controls the high-impedance state of P31 when it is in use as the MTIOC4D pin. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.
Note 2. This is only selectable in 144-pin and 176-pin products.
Note 3. This is only selectable in 100-pin and 144-pin products.

The M4SELR2 register is an 8-bit readable/writable register that selects the MTU4-B/D pins as targets for high-impedance control.

M4BSEL[3:0] Bits (MTU4-B (MTIOC4B) Pin Select)

These bits select the target MTIOC4A pin for high-impedance control.

M4DSEL[3:0] Bits (MTU4-D (MTIOC4D) Pin Select)

These bits select the target MTIOC4C pin for high-impedance control.

25.2.26 MTU/GPT Pin Function Select Register (MGSELR)

Address(es): 0008 C4E9h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	M4G2SEL	M4G1SEL	M3G0SEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	M3G0SEL	MTU3/GPT0 Pin Select	0: Selects the B and D pins of the MTU3 (MTIOC3B and MTIOC3D) pin. 1: Selects the A and B pins of the GPT0 (GTIOC0A and GTIOC0B) pin.	R/W*1
b1	M4G1SEL	MTU4/GPT1 Pin Select	0: Selects the A and C pins of the MTU4 (MTIOC4A and MTIOC4C) pin. 1: Selects the A and B pins of the GPT1 (GTIOC1A and GTIOC1B) pin.	R/W*1
b2	M4G2SEL	MTU4/GPT2 Pin Select	0: Selects the B and D pins of the MTU3 (MTIOC4B and MTIOC4D) pin. 1: Selects the A and B pins of the GPT2 (GTIOC2A and GTIOC2B) pin.	R/W*1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The MGSELR register is an 8-bit readable/writable register that selects MTU pins or GPT pins as targets for high-impedance control.

M3G0SEL Bit (MTU3/GPT0 Pin Select)

This bit selects the B and D pins of MTU3 pin or the A and B outputs of GTP0 pin as targets for high-impedance control in response to output-level comparison.

M4G1SEL Bit (MTU4/GPT1 Pin Select)

This bit selects the A and C pins of MTU4 pin or the A and B outputs of GTP1 pin as targets for high-impedance control in response to output-level comparison.

M4G2SEL Bit (MTU4/GPT2 Pin Select)

This bit selects the B and D pins of MTU4 pin or the A and B outputs of GTP2 pin as targets for high-impedance control in response to output-level comparison.

25.3 Operation

The following shows the target pins and conditions for high-impedance control.

(1) MTU3 pins or GPT0 pins*1 (MTIOC3B/GTIOC0A, MTIOC3D/GTIOC0B)

When one of the following conditions is satisfied while the POECR2.MTU3BDZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE0# input level
  When the ICSR1.POE0F flag becomes 1.
- Operation for comparison of the output levels on the MTIOC3B and MTIOC3D or GTIOC0A and GTIOC0B pins
  When the OCSR1.OSF1 flag becomes 1 while the OCSR1.OCE1 bit is 1.
- SPOER setting
  When the SPOER.MTUCH34HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR2.POE4F flag becomes 1 while the POECR4.IC2ADDMT34ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT34ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT34ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR4.IC5ADDMT34ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

Note 1. The pins selected in the MGSELR.M3G0SEL bit are for the control.

(2) MTU4 pins or GPT1 pins*1 (MTIOC4A/GTIOC1A, MTIOC4C/GTIOC1B)

When one of the following conditions is satisfied while the POECR2.MTU4ACZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE0# input level
  When the ICSR1.POE0F flag becomes 1.
- Operation for comparison of the output levels on the MTIOC4A and MTIOC4C or GTIOC1A and GTIOC1B pins
  When the OCSR1.OSF1 flag becomes 1 while the OCSR1.OCE1 bit is 1.
- SPOER setting
  When the SPOER.MTUCH34HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR2.POE4F flag becomes 1 while the POECR4.IC2ADDMT34ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT34ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT34ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR4.IC5ADDMT34ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

Note 1. The pins selected in the MGSELR.M4G1SEL bit are for the control.

(3) MTU4 pins or GPT2 pins*1 (MTIOC4B/GTIOC2A, MTIOC4D/GTIOC2B)

When one of the following conditions is satisfied while the POECR2.MTU4BDZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE0# input level
  When the ICSR1.POE0F flag becomes 1.
- Operation for comparison of the output levels on the MTIOC4B and MTIOC4D or GTIOC2A and GTIOC2B pins
  When the OCSR1.OSF1 flag becomes 1 while the OCSR1.OCE1 bit is 1.
- SPOER setting
  When the SPOER.MTUCH34HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR2.POE4F flag becomes 1 while the POECR4.IC2ADDMT34ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT34ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT34ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR4.IC5ADDMT34ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

Note 1. The pins selected in the MGSELR.M4G2SEL bit are for the control.

(4) MTU6 pins (MTIOC6B, MTIOC6D)

When one of the following conditions is satisfied while the POECR2.MTU6BDZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE4# input level
  When the ICSR2.POE4F flag becomes 1.
- Operation for comparison of the output levels on the MTIOC6B and MTIOC6D pins
  When the OCSR2.OSF2 flag becomes 1 while the OCSR2.OCE2 bit is 1.
- SPOER setting
  When the SPOER.MTUCH67HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR1.POE0F flag becomes 1 while the POECR4.IC1ADDMT67ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT67ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT67ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR4.IC5ADDMT67ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(5) MTU7 pins (MTIOC7A, MTIOC7C)

When one of the following conditions is satisfied while the POECR2.MTU7ACZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE4# input level
  When the ICSR2.POE4F flag becomes 1.
- Operation for comparison of the output levels on the MTIOC7A and MTIOC7C pins
  When the OCSR2.OSF2 flag becomes 1 while the OCSR2.OCE2 bit is 1.
- SPOER setting
  When the SPOER.MTUCH67HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR1.POE0F flag becomes 1 while the POECR4.IC1ADDMT67ZE bit is 1.

When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT67ZE bit and the ICSR3.POE8E bit are 1.
When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT67ZE bit and the ICSR4.POE10E bit are 1.
When the ICSR5.POE11F flag becomes 1 while the POECR4.IC5ADDMT67ZE bit and the ICSR5.POE11E bit are 1.

- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(6) MTU7 pins (MTIOC7B, MTIOC7D)

When one of the following conditions is satisfied while the POECR2.MTU7BDZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE4# input level
  When the ICSR2.POE4F flag becomes 1.
- Operation for comparison of the output levels on the MTIOC7B and MTIOC7D pins
  When the OCSR2.OSF2 flag becomes 1 while the OCSR2.OCE2 bit is 1.
- SPOER setting
  When the SPOER.MTUCH67HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR1.POE0F flag becomes 1 while the POECR4.IC1ADDMT67ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT67ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT67ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR4.IC5ADDMT67ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(7) MTU0 pin (MTIOC0A)

When one of the following conditions is satisfied while the POECR1.MTU0AZE bit is 1, the pin becomes high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR5.IC2ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR5.IC5ADDMT0ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(8) MTU0 pin (MTIOC0B)

When one of the following conditions is satisfied while the POECR1.MTU0BZE bit is 1, the pin becomes high-impedance.

- Operation for detection of the POE8# input level

When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.

- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR5.IC2ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR5.IC5ADDMT0ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(9) MTU0 pin (MTIOC0C)

When one of the following conditions is satisfied while the POECR1.MTU0CZE bit is 1, the pin becomes high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR5.IC2ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR5.IC5ADDMT0ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(10) MTU0 pin (MTIOC0D)

When one of the following conditions is satisfied while the POECR1.MTU0DZE bit is 1, the pin becomes high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR5.IC2ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR5.IC5ADDMT0ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(11) GPT0 pins (GTIOC0A, GTIOC0B)

When one of the following conditions is satisfied while the POECR3.GPT0ABZE bit is 1, the pins become high-

impedance.

- Operation for detection of the POE10# input level
  When the ICSR4.POE10F flag becomes 1 while the ICSR4.POE10E bit is 1.
- SPOER setting
  When the SPOER.GPT01HIZ bit is set to 1.
- Conditions added by POECR6
  When the ICSR1.POE0F flag becomes 1 while the POECR6.IC1ADDGPT01ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR6.IC2ADDGPT01ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR6.IC3ADDGPT01ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR6.IC5ADDGPT01ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(12) GPT1 pins (GTIOC1A, GTIOC1B)

When one of the following conditions is satisfied while the POECR3.GPT1ABZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE10# input level
  When the ICSR4.POE10F flag becomes 1 while the ICSR4.POE10E bit is 1.
- SPOER setting
  When the SPOER.GPT01HIZ bit is set to 1.
- Conditions added by POECR6
  When the ICSR1.POE0F flag becomes 1 while the POECR6.IC1ADDGPT01ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR6.IC2ADDGPT01ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR6.IC3ADDGPT01ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR6.IC5ADDGPT01ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(13) GPT2 pins (GTIOC2A, GTIOC2B)

When one of the following conditions is satisfied while the POECR3.GPT2ABZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE11# input level
  When the ICSR5.POE11F flag becomes 1 while the ICSR5.POE11E bit is 1.
- SPOER setting
  When the SPOER.GPT23HIZ bit is set to 1.
- Conditions added by POECR6
  When the ICSR1.POE0F flag becomes 1 while the POECR6.IC1ADDGPT23ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR6.IC2ADDGPT23ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR6.IC3ADDGPT23ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR6.IC4ADDGPT23ZE bit and the ICSR4.POE10E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(14) GPT3 pins (GTIOC3A, GTIOC3B)

When one of the following conditions is satisfied while the POECR3.GPT3ABZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE11# input level
  When the ICSR5.POE11F flag becomes 1 while the ICSR5.POE11E bit is 1.
- SPOER setting
  When the SPOER.GPT23HIZ bit is set to 1.
- Conditions added by POECR6
  When the ICSR1.POE0F flag becomes 1 while the POECR6.IC1ADDGPT23ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR6.IC2ADDGPT23ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR6.IC3ADDGPT23ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR6.IC4ADDGPT23ZE bit and the ICSR4.POE10E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

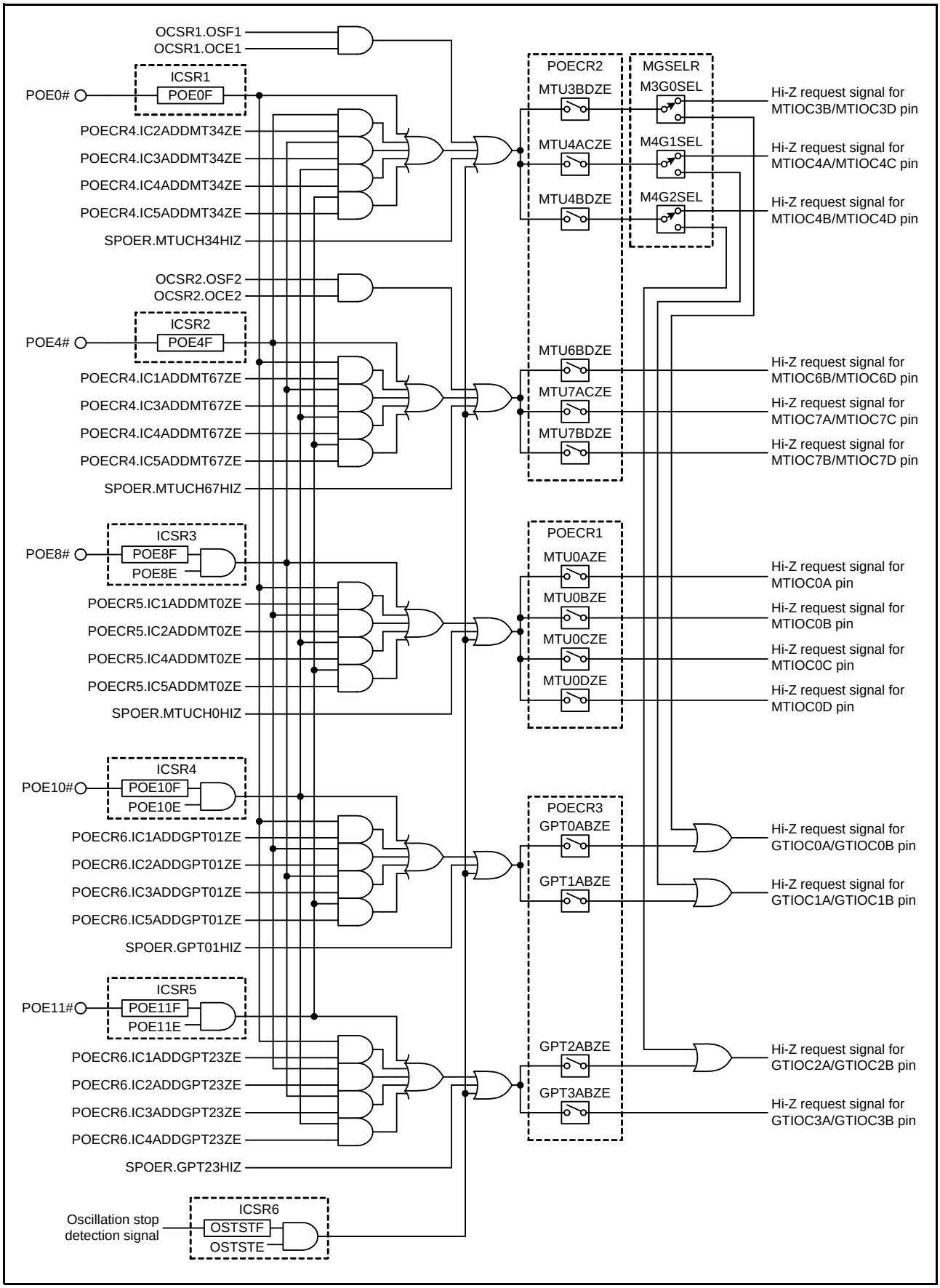


Figure 25.2 Target Pins and Conditions for High-Impedance Control

25.3.1 MTU/GPT Pin Selection

In this MCU, the pin functions for MTU and GPT are respectively multiplexed with multiple sets of port pins. The target pins for high-impedance control can be selected by the pin select register in POE (M0SELR1, M0SELR2, M3SELR, M4SELR1, M4SELR2, G0SELR, G1SELR, G2SELR, G3SELR, or MGSELR register). Table 25.4 shows the correspondence between MTU and GPT pins and select registers.
Note that settings for pins to be used as MTU or GPT must be separately made in the registers of the multi-function pin controller (MPC). Take care so that there are no differences between the pins selected in the POE registers and the pins selected in the MPC registers.

Table 25.4 Correspondence between MTU and GPT Pins

MTU/GPT Pin Functions	Corresponding Ports	Select Registers
MTIOC0A	P34	M0SELR1
	PB3	
MTIOC0B	P15	
	P13	
	PA1	
MTIOC0C	P32	M0SELR2
	PB1	
MTIOC0D	P33	
	PA3	
MTIOC3B	P22	M3SELR + MGSELR
	P17	
	PC5	
	P80*1	
	PB7	
	PE1	
MTIOC3D	P23	
	P16	
	P81*1	
	PC4	
	PB6	
	PE0	
MTIOC4A	P24	M4SELR1 + MGSELR
	P21	
	P82*1	
	PB3	
	PA0	
	PE2	
MTIOC4C	P25	
	P87*1	
	P83*1	
	PB1	
	PE5	
	PE1	
MTIOC4B	P30	M4SELR2 + MGSELR
	P17	
	PC2	
	PE3	
	PD1	
	P54*2	
MTIOC4D	P31	
	P86*1	
	PC3	
	PE4	
	PD2	
	P55*2	
MTIOC6B	PA5	—
MTIOC6D	PA0	
MTIOC7A	PA2	
MTIOC7C	P67*1	
MTIOC7B	PA1	
MTIOC7D	P66*1	
GTIOC0A	P23	G0SELR + MGSELR
	P83*1	
	PA5	
	PE5	
	PD3	
GTIOC0B	P17	
	P81*1	
	PA0	
	PE2	
	PD2	
GTIOC1A	P22	G1SELR + MGSELR
	PC5	
	PA2	
	PE4	
	PD1	
GTIOC1B	P87*1	
	PC3	
	P67*1	
	PE1	
	PD0	
GTIOC2A	P21	G2SELR + MGSELR
	P82*1	
	PA1	
	PE3	
GTIOC2B	P86*1	
	PC2	
	P66*1	
	PE0	
GTIOC3A	PC7	G3SELR
	PE7	
GTIOC3B	PC6	
	PE6	

Note 1. This is only selectable in 144-pin and 176-pin products.
Note 2. This is only selectable in 100-pin and 144-pin products.

25.3.2 Input-Level Detection Operation

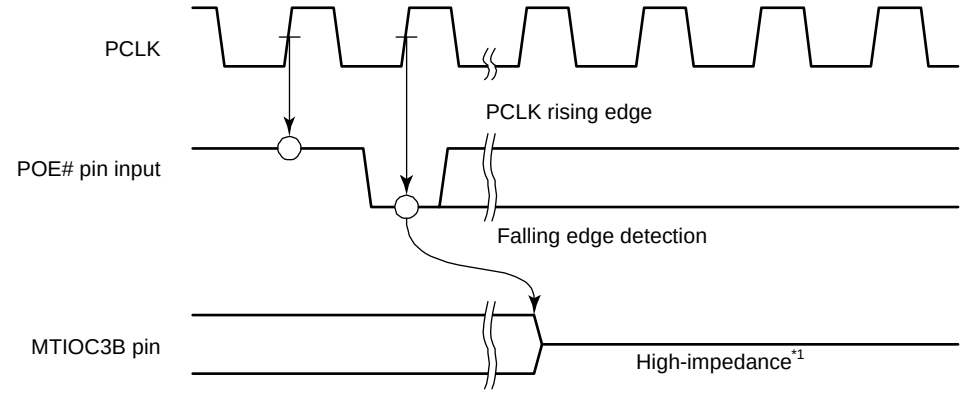
If the input conditions set by ICSR1 to ICSR5 occur on the POE0#, POE4#, POE8#, POE10#, and POE11# pins, the MTU complementary PWM output pins (MTU3 and MTU4 or MTU6 and MTU7), MTU0 pins, and GPT pins are switched to the high-impedance state. Note however, that these pins are still switched to the high-impedance state even when the MTU and GPT functions are not selected for the pins.

(1) Falling Edge Detection

When a change from a high to low level is input to the POE0#, POE4#, POE8#, POE10#, and POE11# pins, the pins multiplexed with MTU complementary PWM output pins, MTU0 pins, and GPT pins are switched to the high-impedance state.

The falling edge is detected after the level is sampled with PCLK. Input a low level for at least one PCLK clock to the POE0#, POE4#, POE8#, POE10#, and POE11# pins.

Figure 25.3 shows a sample timing after the level changes in input to the POE0#, POE4#, POE8#, POE10#, and POE11# pins until the respective pins become high-impedance.



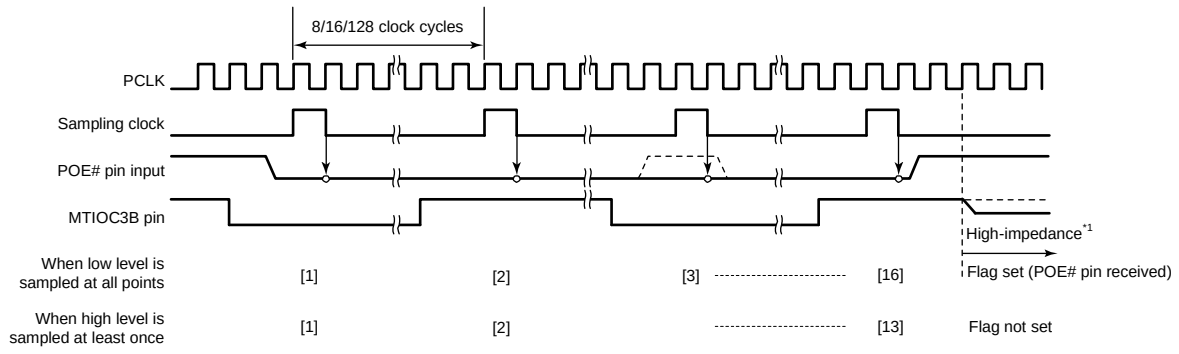
Note 1. Other pins also become high-impedance at the same timing.

Figure 25.3 Operation when A Falling Edge Detection is Selected

(2) Low-Level Detection

Figure 25.4 shows an example of operation when a pin is placed in the high-impedance state in response to low-level detection. When 16 continuous low levels are sampled with the sampling clock selected by the ICSR1 to ICSR5 registers, the low level is recognized and the MTU complementary PWM output pins, MTU0 pins, and GPT pins are switched to the high-impedance state. If even one high level is detected during this interval, the low level is not recognized.

The timing when pins for the MTU complementary PWM output, MTU0 pins, and GPT pins are switched to the high-impedance state after the sampling clock is input is the same in both falling-edge detection and in low-level detection.

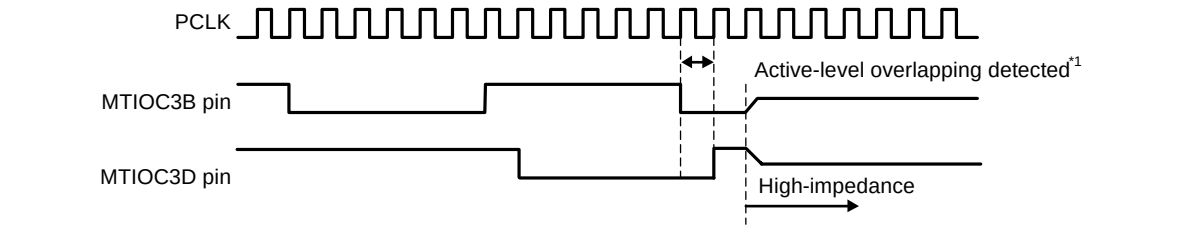


Note 1. Other pins also become high-impedance at the same timing.

Figure 25.4 Operation when A Low-Level Detection is Selected

25.3.3 Output-Level Compare Operation

Figure 25.5 shows an example of the output-level compare operation for the combination of MTIOC3B and MTIOC3D. The operation is the same for the other pin combinations.



Note 1. When the active level of both MTIOC3B and MTIOC3D pins is set to low.

Figure 25.5 Output-Level Compare Operation

25.3.4 High-Impedance Control Using Registers

The high-impedance state of the MTU pins (MTU0, MTU3, MTU4, MTU6, and MTU7) and the GPT pins can be directly controlled by using the SPOER register.
For instance, setting the SPOER.MTUCH34HIZ bit to 1 switches the MTU3 and MTU4 pins specified by the POECR2 register to the high-impedance state.
The high-impedance state of other pins can also be controlled by setting the appropriate bits in SPOER.

25.3.5 High-Impedance Control through Detection of Oscillation Stop

When oscillation stop is detected by the oscillation stop detection function of the clock generator while the ICSR6.OSTSTE bit is 1, the MTU complementary PWM output pins specified by the POECR2 register, the MTU0 pins specified by the POECR1 register, and the GPT pins specified by the POECR3 register are switched to the high-impedance state.

25.3.6 Additional Functions for High-Impedance Control

High-impedance control conditions for the MTU complementary PWM output pins, MTU0 pins, and GPT pins can be added by setting the POECR4 to POECR6 registers.
For instance, the settings listed below can be added as high-impedance control conditions for the MTU3 and MTU4 pins.

- Setting the POECR4.IC2ADDMT34ZE bit to 1 adds the input-level detection by the POE4# pin
- Setting the POECR4.IC3ADDMT34ZE bit to 1 and adds the input-level detection by the POE8# pin
- Setting the POECR4.IC4ADDMT34ZE bit to 1 and adds the input-level detection by the POE10# pin
- Setting the POECR4.IC5ADDMT34ZE bit to 1 and adds the input-level detection by the POE11# pin

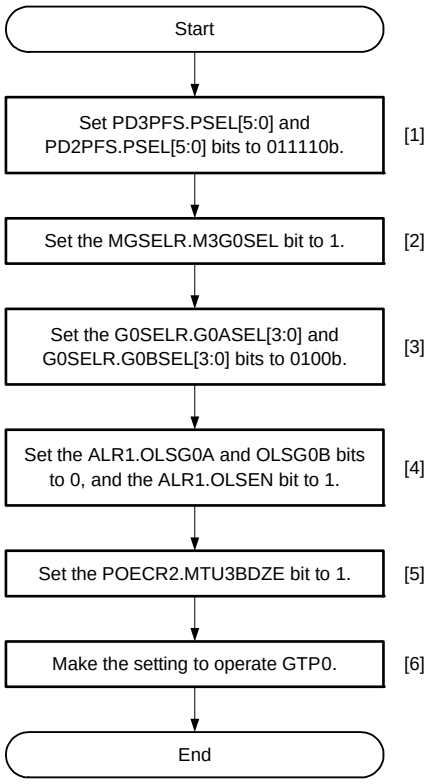
The high-impedance control of other pins can also be controlled by setting the appropriate bits in the POECR4 to POECR6 registers.

25.3.7 Recover from High-Impedance State

MTU pins which have switched to the high-impedance state due to input-level detection can be recovered from the state either by returning them to their initial state with a reset, or by clearing all of the ICSR1.POE0F, ICSR2.POE4F, ICSR3.POE8F, ICSR4.POE10F, and ICSR5.POE11F flags. However, note that when low-level sampling is selected with the ICSR1.POE0M[1:0], ICSR2.POE4M[1:0], ICSR3.POE8M[1:0], ICSR4.POE10M[1:0], and ICSR5.POE11M[1:0] bits, just writing 0 to a flag is ignored (the flag is not set to 0); flags can be cleared by writing 0 to it only after a high level is input to the POE0#, POE4#, POE8#, POE10#, and POE11# pins and is detected.
MTU pins which have switched to the high-impedance state due to output-level detection can be recovered from the state either by returning them to their initial state with a reset, or by setting the OCSR1.OSF1 flag or the OCSR2.OSF2 flag to 0. However, note that just writing 0 to a flag is ignored (the flag is not set to 0); the flags can be cleared by writing 0 to it only after setting the inactive level to be output from the pin. The inactive level is output by setting the registers in MTU and GPT, and the ALR1 register.
MTU pins which have switched to the high-impedance state due to oscillation stop detection can be recovered from the state either by returning them to their initial state with a reset or by setting the SYSTEM.OSTDSR.OSTDF flag to 0 to set the ICSR6.OSTSTF flag to 0.

25.4 POE Setting Procedure

Figure 25.6 shows the procedure for setting the POE. It illustrates an example of high-impedance control in response to comparison of the output levels on the GPT0 pins (GTIOC0A/GTIOC0B). In the figure, PD3 is selected as the GTIOC0A pin and PD2 is selected as the GTIOC0B pin.



Set the GPT0 pins (PD3/PD2) for the peripheral function by the PMR register of the I/O port before making the following settings.

[1] Set the PD3PFS and PD2PFS registers of the MPC to select PD3 as GTIOC0A pin and PD2 as GTIOC0B pin.

[2] Set the MGSELR.M3G0SEL bit to 1 to select the GPT pins (GTIOC0A and GTIOC0B).

[3] Set the G0SELR register to control the high-impedance state of PD3 for GTIOC0A pin and PD2 for GTIOC0B pin.

[4] Set the ALR1.OLSG0A and OLSG0B bits to 0 and the ALR1.OLSEN bit to 1 to set GTIOC0A and GTIOC0B pins as active low.

[5] Set the POECR2.MTU3BDZE bit to 1 to enable high-impedance control on the GTIOC0A and GTIOC0B pins.

[6] Make the setting to operate GTP0 pin.

After the above settings, if GTIOC0A (PD3) and GTIOC0B (PD2) pins are simultaneously at the low level, then the pins become high-impedance and the OCSR1.OSF1 flag is set to 1.
For the method for releasing high-impedance, refer to section 25.3.7, Recover from High-Impedance State.

Figure 25.6 Procedure for Setting the POE

25.5 Interrupts

The POE issues a request to generate an interrupt when the specified condition is satisfied during input-level detection or output-level comparison. Table 25.5 shows the interrupt sources and their conditions.

Table 25.5 Interrupt Sources and Conditions

Name	Interrupt Source	Interrupt Flag	Condition
OEI1	Output enable interrupt 1	POE0F OSF1	When the ICSR1.POE0F flag becomes 1 while the ICSR1.PIE1 bit is 1 or when the OCSR1.OSF1 flag becomes 1 while the OCSR1.OIE1 bit is 1
OEI2	Output enable interrupt 2	POE4F OSF2	When the ICSR2.POE4F flag becomes 1 while the ICSR2.PIE2 bit is 1 or when the OCSR2.OSF2 flag becomes 1 while the OCSR2.OIE2 bit is 1
OEI3	Output enable interrupt 3	POE8F	When the ICSR3.POE8F flag becomes 1 while the ICSR3.PIE3 bit is 1
OEI4	Output enable interrupt 4	POE10F POE11F	When the ICSR4.POE10F flag is set to 1 while the ICSR4.PIE4 bit is 1 or when the ICSR5.POE11F flag becomes 1 while the ICSR5.PIE5 bit is 1

25.6 Usage Notes

25.6.1 Transition to Low Power Consumption Mode

When the POE is used, do not make a transition to software standby mode or deep software standby mode. In these modes, the POE stops and thus the high-impedance control of pins cannot operate.

25.6.2 High-Impedance Control When the MTU and GPT Pins are Not Selected

If high-impedance control for a pin having a multiplexed MTU or GPT pin function is enabled by setting the POECR1 to POECR3 registers and the high-impedance control condition is satisfied, the pin is switched to the high-impedance state even if the MTU/GPT function is not selected for the pin on which it is multiplexed.
To avoid unintended high-impedance states, ensure that there are no differences between the settings for MTU and GPT pin selection in the PmnPFS registers of the MPC and for MTU and GPT pin selection in the pin select register of the POE.

25.6.3 When the POE is Not Used

The high-impedance control of some pins can be enabled using the POE after a reset. When the POE is not used, write 0 to the target bits in the POECR1 to POECR3 registers.

26. General PWM Timer (GPTA)

This MCU has a general PWM timer (GPTA) consisting of a four-channel 16-bit timer.
The GPT operates at a maximum of 120 MHz.

26.1 Overview

Table 26.1 lists the specifications for the GPT, and Table 26.2 shows the functions of the GPT. Figure 26.1 shows a block diagram of the GPT.

Table 26.1 GPT Specifications

Item	Description
Functions	• 16 bits × 4 channels • Up-counting or down-counting (saw waves) or up/down-counting (triangle waves) for each counter. • Operating mode Saw-wave PWM mode Saw-wave one-shot pulse mode Triangle-wave PWM mode 1 Triangle-wave PWM mode 2 Triangle-wave PWM mode 3 • Clock sources independently selectable for each channel • Two I/O pins per channel • Noise filter can be set on each input path. • Two output compare/input capture registers per channel • For the two output compare/input capture registers of each channel, four registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use. • In output compare operation, buffer switching can be at crests or troughs, enabling the generation of laterally asymmetric PWM waveforms. • Registers for setting up frame cycles in each channel (with capability for generating interrupts at overflow or underflow) • Synchronous operation of the several counters • Synchronous operation modes: simultaneous start or phase shifting start by desired times • Generation of dead times in PWM operation • Through combination of three counters, generation of three-phase PWM waveforms incorporating dead times • Starting, clearing, and stopping counters in response to external or internal triggers (hardware sources) • Internal trigger sources: software and compare match

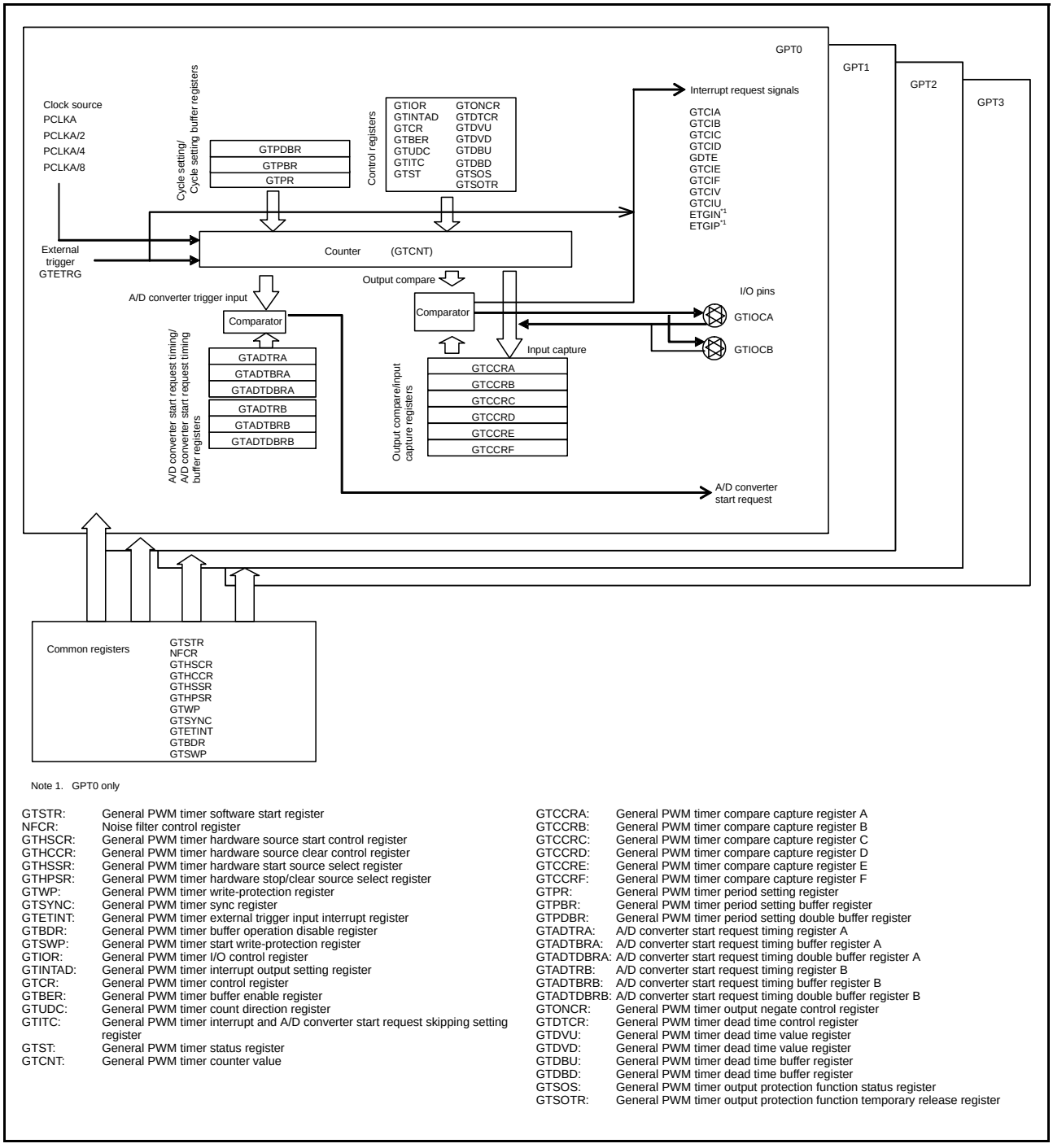
Table 26.2 GPT Functions (1/2)

Item		GPT0	GPT1	GPT2	GPT3
Count clocks		PCLKA PCLKA/2 PCLKA/4 PCLKA/8	PCLKA PCLKA/2 PCLKA/4 PCLKA/8	PCLKA PCLKA/2 PCLKA/4 PCLKA/8	PCLKA PCLKA/2 PCLKA/4 PCLKA/8
Output compare/input capture registers (GTCCR)		GTCCRA GTCCRB	GTCCRA GTCCRB	GTCCRA GTCCRB	GTCCRA GTCCRB
Compare/buffer registers		GTCCRC GTCCRD GTCCRE GTCCRF	GTCCRC GTCCRD GTCCRE GTCCRF	GTCCRC GTCCRD GTCCRE GTCCRF	GTCCRC GTCCRD GTCCRE GTCCRF
PWM period setting register		GTPR	GTPR	GTPR	GTPR
PWM period setting buffer registers		GTPBR GTPDBR	GTPBR GTPDBR	GTPBR GTPDBR	GTPBR GTPDBR
I/O pins		GTIOC0A GTIOC0B	GTIOC1A GTIOC1B	GTIOC2A GTIOC2B	GTIOC3A GTIOC3B
External trigger input pin		GTETRG			
Count clear sources		GTPR register compare match, input capture, GTETRG pin input, GTIOC3A/GTIOC3B pin input, and GTIOC3A/GTIOC3B internal output (output compare)			
Compare match output	Low output	✓	✓	✓	✓
	High output	✓	✓	✓	✓
	Toggle output	✓	✓	✓	✓
Input capture function		✓	✓	✓	✓
Synchronous operation		✓	✓	✓	✓
Phase shift start		✓	✓	✓	✓
Automatic addition of dead time		✓	✓	✓	✓
PWM mode		✓	✓	✓	✓
Buffer operation		✓	✓	✓	✓
One-shot operation		✓	✓	✓	✓
DMAC/DTC activation		All interrupt sources			
A/D converter start trigger		Compare match of GTADTRA or GTADTRB	Compare match of GTADTRA or GTADTRB	Compare match of GTADTRA or GTADTRB	Compare match of GTADTRA or GTADTRB

Table 26.2 GPT Functions (2/2)

Item	GPT0	GPT1	GPT2	GPT3
Interrupt sources	Nine sources • GTCCRA compare match/input capture (GTCIA0) • GTCCRB compare match/input capture (GTCIB0) • GTCCRC compare match (GTCIC0) • GTCCRD compare match (GTCID0) • Dead time error (GDTE0) • GTCCRE compare match (GTCIE0) • GTCCRF compare match (GTCIF0) • GTCNT overflow (GTPR compare match) (GTCIV0) • GTCNT underflow (GTCIU0)	Nine sources • GTCCRA compare match/input capture (GTCIA1) • GTCCRB compare match/input capture (GTCIB1) • GTCCRC compare match (GTCIC1) • GTCCRD compare match (GTCID1) • Dead time error (GDTE1) • GTCCRE compare match (GTCIE1) • GTCCRF compare match (GTCIF1) • GTCNT overflow (GTPR compare match) (GTCIV1) • GTCNT underflow (GTCIU1)	Nine sources • GTCCRA compare match/input capture (GTCIA2) • GTCCRB compare match/input capture (GTCIB2) • GTCCRC compare match (GTCIC2) • GTCCRD compare match (GTCID2) • Dead time error (GDTE2) • GTCCRE compare match (GTCIE2) • GTCCRF compare match (GTCIF2) • GTCNT overflow (GTPR compare match) (GTCIV2) • GTCNT underflow (GTCIU2)	Nine sources • GTCCRA compare match/input capture (GTCIA3) • GTCCRB compare match/input capture (GTCIB3) • GTCCRC compare match (GTCIC3) • GTCCRD compare match (GTCID3) • Dead time error (GDTE3) • GTCCRE compare match (GTCIE3) • GTCCRF compare match (GTCIF3) • GTCNT overflow (GTPR compare match) (GTCIV3) • GTCNT underflow (GTCIU3)
Common interrupt source	External trigger			
Interrupt skipping function	Skips GTCNT overflows (GTPR compare match) (GTCIV0)/ GTCNT underflow (GTCIU0) interrupts (with interlocking function for other interrupts or A/D conversion requests).	Skips GTCNT overflows (GTPR compare match) (GTCIV1)/ GTCNT underflow (GTCIU1) interrupts (with interlocking function for other interrupts or A/D conversion requests).	Skips GTCNT overflows (GTPR compare match) (GTCIV2)/ GTCNT underflow (GTCIU2) interrupts (with interlocking function for other interrupts or A/D conversion requests).	Skips GTCNT overflows (GTPR compare match) (GTCIV3)/ GTCNT underflow (GTCIU3) interrupts (with interlocking function for other interrupts or A/D conversion requests).
Event link function (output)	GTCCRA compare match (compare match A) GTCCRB compare match (compare match B) GTCCRC compare match (compare match C) GTCCRD compare match (compare match D) GTCNT overflow (overflow) GTCNT underflow (underflow)			
Event link function (input)	Count start Count restart Count stop Input capture			

✓: Possible



Note 1. GPT0 only

GTSTR: General PWM timer software start register
NFCR: Noise filter control register
GTHSCR: General PWM timer hardware source start control register
GTHCCR: General PWM timer hardware source clear control register
GTHSSR: General PWM timer hardware start source select register
GTHPSR: General PWM timer hardware stop/clear source select register
GTWP: General PWM timer write-protection register
GTSYNC: General PWM timer sync register
GTETINT: General PWM timer external trigger input interrupt register
GTBDR: General PWM timer buffer operation disable register
GTSWP: General PWM timer start write-protection register
GTIOR: General PWM timer I/O control register
GTINTAD: General PWM timer interrupt output setting register
GTCR: General PWM timer control register
GTBER: General PWM timer buffer enable register
GTUDC: General PWM timer count direction register
GTITC: General PWM timer interrupt and A/D converter start request skipping setting register
GTST: General PWM timer status register
GTCNT: General PWM timer counter value

GTCCRA: General PWM timer compare capture register A
GTCCRB: General PWM timer compare capture register B
GTCCRC: General PWM timer compare capture register C
GTCCRD: General PWM timer compare capture register D
GTCCRE: General PWM timer compare capture register E
GTCCRF: General PWM timer compare capture register F
GTPR: General PWM timer period setting register
GTPBR: General PWM timer period setting buffer register
GTPDBR: General PWM timer period setting double buffer register
GTADTRA: A/D converter start request timing register A
GTADTBRA: A/D converter start request timing buffer register A
GTADTDBRA: A/D converter start request timing double buffer register A
GTADTRB: A/D converter start request timing register B
GTADTBRB: A/D converter start request timing buffer register B
GTADTDBRB: A/D converter start request timing double buffer register B
GTONCR: General PWM timer output negate control register
GTDTCR: General PWM timer dead time control register
GTDVU: General PWM timer dead time value register
GTDVD: General PWM timer dead time value register
GTDBU: General PWM timer dead time buffer register
GTDBD: General PWM timer dead time buffer register
GTSOS: General PWM timer output protection function status register
GTSOTR: General PWM timer output protection function temporary release register

Figure 26.1 GPT Block Diagram

Table 26.3 lists the I/O pins used in the GPT.

Table 26.3 GPT I/O Pins

Channel	Pin Name	I/O	Function
GPT	GTETRG	Input	External trigger input pin
GPT0	GTIOC0A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC0B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT1	GTIOC1A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC1B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT2	GTIOC2A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC2B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT3	GTIOC3A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC3B	I/O	GTCCRB register input capture input/output compare output/PWM output pin

26.2 Register Descriptions

26.2.1 General PWM Timer Software Start Register (GTSTR)

Address(es): GPT.GTSTR 000C 2000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	CST3	CST2	CST1	CST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CST0	GPT0.GTCNT Count Start	0: Count operation is stopped 1: Count operation is started	R/W
b1	CST1	GPT1.GTCNT Count Start		R/W
b2	CST2	GPT2.GTCNT Count Start		R/W
b3	CST3	GPT3.GTCNT Count Start		R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTSTR register starts or stops the GPTn.GTCNT counter (n = 0 to 3).

CSTn Bit (GPTn.GTCNT Count Start) (n = 0 to 3)

This bit starts or stops the GPTn.GTCNT counter.

If the GTSWP.SWPn bit is set to disable writing to the CSTn bit, writing to the CSTn bit is ignored.

The counter can also be started or stopped by a hardware source by setting the GTHSCR register. When count operation is started by a hardware source, this bit is automatically set to 1, and when count operation is stopped by a hardware source, this bit is automatically set to 0.

26.2.2 Noise Filter Control Register (NFCR)

Address(es): GPT.NFCR 000C 2002h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	NFCS3[1:0]		NFCS2[1:0]		NFCS1[1:0]		NFCS0[1:0]		NFB3EN	NFA3EN	NFB2EN	NFA2EN	NFB1EN	NFA1EN	NFB0EN	NFA0EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFA0EN	Noise Filter 0A Enable	0: The noise filter for the GTIOC0A pin is disabled. 1: The noise filter for the GTIOC0A pin is enabled.	R/W
b1	NFB0EN	Noise Filter 0B Enable	0: The noise filter for the GTIOC0B pin is disabled. 1: The noise filter for the GTIOC0B pin is enabled.	R/W
b2	NFA1EN	Noise Filter 1A Enable	0: The noise filter for the GTIOC1A pin is disabled. 1: The noise filter for the GTIOC1A pin is enabled.	R/W
b3	NFB1EN	Noise Filter 1B Enable	0: The noise filter for the GTIOC1B pin is disabled. 1: The noise filter for the GTIOC1B pin is enabled.	R/W
b4	NFA2EN	Noise Filter 2A Enable	0: The noise filter for the GTIOC2A pin is disabled. 1: The noise filter for the GTIOC2A pin is enabled.	R/W
b5	NFB2EN	Noise Filter 2B Enable	0: The noise filter for the GTIOC2B pin is disabled. 1: The noise filter for the GTIOC2B pin is enabled.	R/W
b6	NFA3EN	Noise Filter 3A Enable	0: The noise filter for the GTIOC3A pin is disabled. 1: The noise filter for the GTIOC3A pin is enabled.	R/W
b7	NFB2EN	Noise Filter 3B Enable	0: The noise filter for the GTIOC3B pin is disabled. 1: The noise filter for the GTIOC3B pin is enabled.	R/W
b9, b8	NFCS0[1:0]	GPT0 Noise Filter Sampling Clock Select	b9 b8 0 0: PCLKA/1 0 1: PCLKA/4 1 0: PCLKA/32 1 1: Clock source for counting	R/W
b11, b10	NFCS1[1:0]	GPT1 Noise Filter Sampling Clock Select	b11 b10 0 0: PCLKA/1 0 1: PCLKA/4 1 0: PCLKA/32 1 1: Clock source for counting	R/W
b13, b12	NFCS2[1:0]	GPT2 Noise Filter Sampling Clock Select	b13 b12 0 0: PCLKA/1 0 1: PCLKA/4 1 0: PCLKA/32 1 1: Clock source for counting	R/W
b15, b14	NFCS3[1:0]	GPT3 Noise Filter Sampling Clock Select	b15 b14 0 0: PCLKA/1 0 1: PCLKA/4 1 0: PCLKA/32 1 1: Clock source for counting	R/W

The NFCR register controls enabling and disabling of the noise filters and selects the sampling clocks for the noise filters.

NFAnEN Bit (Noise Filter nA Enable) (n = 0 to 3)

This bit disables or enables the noise filter for input from the GTIOCnA pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the GTIOR register before doing so.

NFBnEN Bit (Noise Filter nB Enable) (n = 0 to 3)

This bit disables or enables the noise filter for input from the GTIOCnB pin. Since changing the value of this bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the GTIOR register before changing the value of the bit.

NFCSn[1:0] Bits (GPTn Noise Filter Sampling Clock Select) (n = 0 to 3)

These bits set the sampling clock for the noise filter. When changing the setting of these bits, set the corresponding pin function of the GTIOR register to output compare. After these bits are set, wait for two cycles of the selected sampling interval before setting the input capture function of the GTIOR register.

26.2.3 General PWM Timer Hardware Source Start/Stop Control Register (GTHSCR)

Address(es): GPT.GTHSCR 000C 2004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CPHW3[1:0]		CPHW2[1:0]		CPHW1[1:0]		CPHW0[1:0]		CSHW3[1:0]		CSHW2[1:0]		CSHW1[1:0]		CSHW0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CSHW0[1:0]	GPT0.GTCNT Hardware Source Count Start	b1 b0 0 0: Count operation is not started by a hardware source. 0 1: Count operation is started at the rising edge of a hardware source. 1 0: Count operation is started at the falling edge of a hardware source. 1 1: Count operation is started at both rising and falling edges of a hardware source.	R/W
b3, b2	CSHW1[1:0]	GPT1.GTCNT Hardware Source Count Start	b3 b2 0 0: Count operation is not started by a hardware source. 0 1: Count operation is started at the rising edge of a hardware source. 1 0: Count operation is started at the falling edge of a hardware source. 1 1: Count operation is started at both rising and falling edges of a hardware source.	R/W
b5, b4	CSHW2[1:0]	GPT2.GTCNT Hardware Source Count Start	b5 b4 0 0: Count operation is not started by a hardware source. 0 1: Count operation is started at the rising edge of a hardware source. 1 0: Count operation is started at the falling edge of a hardware source. 1 1: Count operation is started at both rising and falling edges of a hardware source.	R/W
b7, b6	CSHW3[1:0]	GPT3.GTCNT Hardware Source Count Start	b7 b6 0 0: Count operation is not started by a hardware source. 0 1: Count operation is started at the rising edge of a hardware source. 1 0: Count operation is started at the falling edge of a hardware source. 1 1: Count operation is started at both rising and falling edges of a hardware source.	R/W
b9, b8	CPHW0[1:0]	GPT0.GTCNT Hardware Source Count Stop	b9 b8 0 0: Count operation is not stopped by a hardware source. 0 1: Count operation is stopped at the rising edge of a hardware source. 1 0: Count operation is stopped at the falling edge of a hardware source. 1 1: Count operation is stopped at both rising and falling edges of a hardware source.	R/W
b11, b10	CPHW1[1:0]	GPT1.GTCNT Hardware Source Count Stop	b11 b10 0 0: Count operation is not stopped by a hardware source. 0 1: Count operation is stopped at the rising edge of a hardware source. 1 0: Count operation is stopped at the falling edge of a hardware source. 1 1: Count operation is stopped at both rising and falling edges of a hardware source.	R/W

Bit	Symbol	Bit Name	Description	R/W
b13, b12	CPHW2[1:0]	GPT2.GTCNT Hardware Source Count Stop	b13 b12 0 0: Count operation is not stopped by a hardware source. 0 1: Count operation is stopped at the rising edge of a hardware source. 1 0: Count operation is stopped at the falling edge of a hardware source. 1 1: Count operation is stopped at both rising and falling edges of a hardware source.	R/W
b15, b14	CPHW3[1:0]	GPT3.GTCNT Hardware Source Count Stop	b15 b14 0 0: Count operation is not stopped by a hardware source. 0 1: Count operation is stopped at the rising edge of a hardware source. 1 0: Count operation is stopped at the falling edge of a hardware source. 1 1: Count operation is stopped at both rising and falling edges of a hardware source.	R/W

The GTHSCR register sets a hardware source to start or stop the GPTn.GTCNT counter (n = 0 to 3).
When starting and stopping the GPTn.GTCNT counter by a hardware source occur simultaneously, counter start is given priority.

CSHWn[1:0] Bits (GPTn.GTCNT Hardware Source Count Start) (n = 0 to 3)

The GPTn.GTCNT counter is started by a hardware source.
When the count operation is started by a hardware source, the corresponding bit in GTSTR automatically becomes 1.
The hardware source can be selected by GTHSSR.

CPHWn[1:0] Bits (GPTn.GTCNT Hardware Source Count Stop) (n = 0 to 3)

The GPTn.GTCNT counter is stopped by a hardware source.
When the count operation is stopped by a hardware source, the corresponding bit in GTSTR automatically becomes 0.
The hardware source can be selected by GTHPSR.

26.2.4 General PWM Timer Hardware Source Clear Control Register (GTHCCR)

Address(es): GPT.GTHCCR 000C 2006h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CCSW3	CCSW2	CCSW1	CCSW0	CCHW3[1:0]		CCHW2[1:0]		CCHW1[1:0]		CCHW0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CCHW0[1:0]	GPT0.GTCNT Hardware Source Count Clear	b1 b0 0 0: Count is not cleared by a hardware source. 0 1: Count is cleared at the rising edge of a hardware source. 1 0: Count is cleared at the falling edge of a hardware source. 1 1: Count is cleared at both rising and falling edges of a hardware source.	R/W
b3, b2	CCHW1[1:0]	GPT1.GTCNT Hardware Source Count Clear	b3 b2 0 0: Count is not cleared by a hardware source. 0 1: Count is cleared at the rising edge of a hardware source. 1 0: Count is cleared at the falling edge of a hardware source. 1 1: Count is cleared at both rising and falling edges of a hardware source.	R/W
b5, b4	CCHW2[1:0]	GPT2.GTCNT Hardware Source Count Clear	b5 b4 0 0: Count is not cleared by a hardware source. 0 1: Count is cleared at the rising edge of a hardware source. 1 0: Count is cleared at the falling edge of a hardware source. 1 1: Count is cleared at both rising and falling edges of a hardware source.	R/W
b7, b6	CCHW3[1:0]	GPT3.GTCNT Hardware Source Count Clear	b7 b6 0 0: Count is not cleared by a hardware source. 0 1: Count is cleared at the rising edge of a hardware source. 1 0: Count is cleared at the falling edge of a hardware source. 1 1: Count is cleared at both rising and falling edges of a hardware source.	R/W
b8	CCSW0	GPT0.GTCNT Counter Clear	When 1 is written to this bit, the count is cleared. This bit automatically returns to 0 after the writing of 1. These bits are read as 0.	R/W
b9	CCSW1	GPT1.GTCNT Counter Clear	When 1 is written to this bit, the count is cleared. This bit automatically returns to 0 after the writing of 1. These bits are read as 0.	R/W
b10	CCSW2	GPT2.GTCNT Counter Clear	When 1 is written to this bit, the count is cleared. This bit automatically returns to 0 after the writing of 1. These bits are read as 0.	R/W
b11	CCSW3	GPT3.GTCNT Counter Clear	When 1 is written to this bit, the count is cleared. This bit automatically returns to 0 after the writing of 1. These bits are read as 0.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTHCCR register sets a hardware source to clear the GPTn.GTCNT counter (n = 0 to 3).
Once the clearing of GPTn.GTCNT counter by a hardware source is set, count clearing by the hardware source is executed whether the GPTn.GTCNT counter is started (GTSTR.CSTn = 1; n = 0 to 3) or stopped (GTSTR.CSTn = 0; n = 0 to 3).

When the count direction is down-counting (GTST.TUCF flag is 0) in saw-wave mode, the GPTn.GTCNT counter is set to the value set in the GTPR register (the GTPBR register in buffer operation). In other cases, the GPTn.GTCNT counter is set to 0000h when count clearing is executed.

CCHWn[1:0] Bits (GPTn.GTCNT Hardware Source Count Clear) (n = 0 to 3)

The GPTn.GTCNT is cleared by a hardware source.
The hardware source can be selected by GTHPSR. The hardware source is accepted repeatedly when the CCHWn[1:0] bits are set to 01b, 10b, or 11b.

CCSWn Bit (GPTn.GTCNT Counter Clear) (n = 0 to 3)

When 1 is written to this bit, the GPTn.GTCNT counter is cleared. This bit automatically returns to 0 after the writing of 1. This bit is read as 0.

26.2.5 General PWM Timer Hardware Start Source Select Register (GTHSSR)

Address(es): GPT.GTHSSR 000C 2008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CSHSL3[3:0]				CSHSL2[3:0]				CSHSL1[3:0]				CSHSL0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CSHSL0[3:0]	GPT0.GTCNT Hardware Count Start Source Select	b3 b0 1 0 0 0: GTIOC3A pin input 1 0 0 1: GTIOC3B pin input 1 0 1 0: GTIOC3A internal output (output compare) 1 0 1 1: GTIOC3B internal output (output compare) 1 1 0 0: GTETRG pin input Settings other than the above are prohibited when count operation is started by a hardware source.	R/W
b7 to b4	CSHSL1[3:0]	GPT1.GTCNT Hardware Count Start Source Select	b7 b4 1 0 0 0: GTIOC3A pin input 1 0 0 1: GTIOC3B pin input 1 0 1 0: GTIOC3A internal output (output compare) 1 0 1 1: GTIOC3B internal output (output compare) 1 1 0 0: GTETRG pin input Settings other than the above are prohibited when count operation is started by a hardware source.	R/W
b11 to b8	CSHSL2[3:0]	GPT2.GTCNT Hardware Count Start Source Select	b11 b8 1 0 0 0: GTIOC3A pin input 1 0 0 1: GTIOC3B pin input 1 0 1 0: GTIOC3A internal output (output compare) 1 0 1 1: GTIOC3B internal output (output compare) 1 1 0 0: GTETRG pin input Settings other than the above are prohibited when count operation is started by a hardware source.	R/W
b15 to b12	CSHSL3[3:0]	GPT3.GTCNT Hardware Count Start Source Select	b15 b12 1 0 0 0: GTIOC3A pin input 1 0 0 1: GTIOC3B pin input 1 1 0 0: GTETRG pin input Settings other than the above are prohibited when count operation is started by a hardware source.	R/W

The GTHSSR register sets the hardware source to start the GPTn.GTCNT counter (n = 0 to 3).
Use the GTHSCR register to set the edge polarity of the hardware source.
To change the source, set the GTHSCR.CSHWn[1:0] bits to 00b before changing the source.

CSHSLn[3:0] Bits (GPTn.GTCNT Hardware Count Start Source Select) (n = 0 to 3)

These bits select the hardware source to start the GPTn.GTCNT.
When 1000b is selected as the hardware source, set the GPT3.GTIOR.GTIOA[5] bit and the GPT3.GTONCR.OAE bit to 0. When 1001b is selected as the hardware source, set the GPT3.GTIOR.GTIOB[5] bit and the GPT3.GTONCR.OBE bit to 0.
If a hardware source to start count operation is not selected in the GTHSCR register, it is not necessary to change the value of these bits from their initial value.

26.2.6 General PWM Timer Hardware Stop/Clear Source Select Register (GTHPSR)

Address(es): GPT.GTHPSR 000C 200Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CSHPL3[3:0]				CSHPL2[3:0]				CSHPL1[3:0]				CSHPL0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CSHPL0[3:0]	GPT0.GTCNT Hardware Count Stop/ Clear Source Select	b3 b0 1 0 0 0: GTIOC3A pin input 1 0 0 1: GTIOC3B pin input 1 0 1 0: GTIOC3A internal output (output compare) 1 0 1 1: GTIOC3B internal output (output compare) 1 1 0 0: GTETRG pin input Settings other than the above are prohibited when count operation is stopped or cleared by a hardware source.	R/W
b7 to b4	CSHPL1[3:0]	GPT1.GTCNT Hardware Count Stop/ Clear Source Select	b7 b4 1 0 0 0: GTIOC3A pin input 1 0 0 1: GTIOC3B pin input 1 0 1 0: GTIOC3A internal output (output compare) 1 0 1 1: GTIOC3B internal output (output compare) 1 1 0 0: GTETRG pin input Settings other than the above are prohibited when count operation is stopped or cleared by a hardware source.	R/W
b11 to b8	CSHPL2[3:0]	GPT2.GTCNT Hardware Count Stop/ Clear Source Select	b11 b8 1 0 0 0: GTIOC3A pin input 1 0 0 1: GTIOC3B pin input 1 0 1 0: GTIOC3A internal output (output compare) 1 0 1 1: GTIOC3B internal output (output compare) 1 1 0 0: GTETRG pin input Settings other than the above are prohibited when count operation is stopped or cleared by a hardware source.	R/W
b15 to b12	CSHPL3[3:0]	GPT3.GTCNT Hardware Count Stop/ Clear Source Select	b15 b12 1 0 0 0: GTIOC3A pin input 1 0 0 1: GTIOC3B pin input 1 0 1 0: GTIOC3A internal output (output compare) 1 0 1 1: GTIOC3B internal output (output compare) 1 1 0 0: GTETRG pin input Settings other than the above are prohibited when count operation is stopped or cleared by a hardware source.	R/W

The GTHPSR register sets the hardware source to stop or clear the GPTn.GTCNT counter (n = 0 to 3).
Use the GTHSCR register to set the edge polarity of the hardware source to stop count operation. Use the GTHCCR register to set the edge polarity of the hardware source to clear the count.
To change the source, set the GTHSCR.CPHWn[1:0] bits and GTHCCR.CCHWn[1:0] bits to 00b before changing the source.

CSHPLn[3:0] Bits (GPTn.GTCNT Hardware Count Stop/Clear Source Select) (n = 0 to 3)

This bit selects the hardware source to stop or clear the GPTn.GTCNT counter.
When 1000b is selected as the hardware source, set the GPT3.GTIOR.GTIOA[5] bit and the GPT3.GTONCR.OAE bit to 0. When 1001b is selected as the hardware source, set the GPT3.GTIOR.GTIOB[5] bit and the GPT3.GTONCR.OBE bit to 0.
If a hardware source to stop or clear count operation is not selected in the GTHSCR or GTHCCR register, it is not necessary to change the value of these bits from their initial value.

26.2.7 General PWM Timer Write-Protection Register (GTWP)

Address(es): GPT.GTWP 000C 200Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	WP3	WP2	WP1	WP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WP0	GPT0 Register Write Disable	0: Write to the register is enabled 1: Write to the register is disabled	R/W
b1	WP1	GPT1 Register Write Disable	0: Write to the register is enabled 1: Write to the register is disabled	R/W
b2	WP2	GPT2 Register Write Disable	0: Write to the register is enabled 1: Write to the register is disabled	R/W
b3	WP3	GPT3 Register Write Disable	0: Write to the register is enabled 1: Write to the register is disabled	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

GTWP enables or disables writing to registers to prevent accidental modification.

For registers that are write enabled or disabled depending on the setting of the GPWP register, see section 26.7.1, Write-Protection for Registers.

26.2.8 General PWM Timer Sync Register (GTSYNC)

Address(es): GPT.GTSYNC 000C 200Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	SYNC3[1:0]		—	—	SYNC2[1:0]		—	—	SYNC1[1:0]		—	—	SYNC0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SYNC0[1:0]	GPT0.GTCNT Counter Synchronous Clear Source Select	b1 b0 0 0: Synchronous clear is not performed. 0 1: GPT0.GTCNT is synchronously cleared by a GPT1 clearing source. 1 0: GPT0.GTCNT is synchronously cleared by a GPT2 clearing source. 1 1: GPT0.GTCNT is synchronously cleared by a GPT3 clearing source.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	SYNC1[1:0]	GPT1.GTCNT Counter Synchronous Clear Source Select	b5 b4 0 0: GPT1.GTCNT is synchronously cleared by a GPT0 clearing source. 0 1: Synchronous clear is not performed. 1 0: GPT1.GTCNT is synchronously cleared by a GPT2 clearing source. 1 1: GPT1.GTCNT is synchronously cleared by a GPT3 clearing source.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	SYNC2[1:0]	GPT2.GTCNT Counter Synchronous Clear Source Select	b9 b8 0 0: GPT2.GTCNT is synchronously cleared by a GPT0 clearing source. 0 1: GPT2.GTCNT is synchronously cleared by a GPT1 clearing source. 1 0: Synchronous clear is not performed. 1 1: GPT2.GTCNT is synchronously cleared by a GPT3 clearing source.	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13, b12	SYNC3[1:0]	GPT3.GTCNT Counter Synchronous Clear Source Select	b13 b12 0 0: GPT3.GTCNT is synchronously cleared by a GPT0 clearing source. 0 1: GPT3.GTCNT is synchronously cleared by a GPT1 clearing source. 1 0: GPT3.GTCNT is synchronously cleared by a GPT2 clearing source. 1 1: Synchronous clear is not performed	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

GTSYNC sets the clearing source of the GPTn.GTCNT counter for synchronous clearing/synchronous operation. This register should be modified while the GPTn.GTCNT counter is stopped (n = 0 to 3).

SYNCn[1:0] Bits (GPTn.GTCNT Counter Synchronous Clear Source Select) (n = 0 to 3)

These bits select which channel's counter clearing source is used to clear the GPTn.GTCNT counter. When setting the SYNCn[1:0] bits, first set the GPTn.GTCR.CCLR[1:0] bits to 11b (cleared by counter clearing in another channel performing synchronous clearing/synchronous operation).

26.2.9 General PWM Timer External Trigger Input Interrupt Register (GTETINT)

Address(es): GPT.GTETINT 000C 2010h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GTETRGEN	GTENFCS[1:0]		—	—	—	—	—	—	—	—	—	—	—	ETINEN	ETIPEN
Value after reset:	0	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	ETIPEN	External Trigger Rising Input Interrupt Request Enable	0: Interrupt request is disabled 1: Interrupt request is enabled	R/W
b1	ETINEN	External Trigger Falling Input Interrupt Request Enable	0: Interrupt request is disabled 1: Interrupt request is enabled	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	—	Reserved	The read value is undefined. The write value should be 1.	R/W
b12 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14, b13	GTENFCS[1:0]	GTETRG Noise Filter Sampling Clock Select	b14 b13 0 0: PCLKA/1 0 1: PCLKA/2 1 0: PCLKA/4 1 1: PCLKA/32	R/W
b15	GTETRGEN	GTETRG Noise Filter Enable	0: The noise filter for the GTETRG pin is disabled. 1: The noise filter for the GTETRG pin is enabled.	R/W

GTETINT enables or disables interrupts through the external trigger input pin (GTETRG).

ETIPEN Bit (External Trigger Rising Input Interrupt Request Enable)

This bit enables or disables an interrupt request generated at the rising edge of an external trigger input.

ETINEN Bit (External Trigger Falling Input Interrupt Request Enable)

This bit enables or disables an interrupt request generated at the falling edge of an external trigger input.

GTENFCS[1:0] Bits (GTETRG Noise Filter Sampling Clock Select)

These bits set the sampling interval for the noise filters. After setting these bits, wait for two selected sampling periods, set the GTHSCR, GTHCCR, and GTHSSR registers to enable the counter operation by a hardware source of GTETRG.

GTETRGEN Bit (GTETRG Noise Filter Enable)

This bit disables or enables the noise filter for input from the GTETRG pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, set the GTHSCR, GTHCCR, and GTHSSR registers to disable the counter operation by a hardware source of GTETRG before changing the value.

26.2.10 General PWM Timer Buffer Operation Disable Register (GTBDR)

Address(es): GPT.GTBDR 000C 2014h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BD33	BD32	BD31	BD30	BD23	BD22	BD21	BD20	BD13	BD12	BD11	BD10	BD03	BD02	BD01	BD00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BD00	GPT0.GTCCR Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
b1	BD01	GPT0.GTPR Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
b2	BD02	GPT0.GTADTR Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
b3	BD03	GPT0.GTDV Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
b4	BD10	GPT1.GTCCR Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
b5	BD11	GPT1.GTPR Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
b6	BD12	GPT1.GTADTR Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
b7	BD13	GPT1.GTDV Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
b8	BD20	GPT2.GTCCR Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
b9	BD21	GPT2.GTPR Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
b10	BD22	GPT2.GTADTR Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
b11	BD23	GPT2.GTDV Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
b12	BD30	GPT3.GTCCR Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
b13	BD31	GPT3.GTPR Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
b14	BD32	GPT3.GTADTR Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
b15	BD33	GPT3.GTDV Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled	R/W

The GTBDR register collectively enables or disables buffer operation of each channel. Even though a bit in the GTBDR register is set to 0 (buffer operation is enabled), buffer operation is not performed unless buffer operation is enabled by the GTBER register.

BDn0 Bit (GPTn.GTCCR Buffer Operation Disable) (n = 0 to 3)

This bit disables buffer operation using the GTCCRA, GTCCRC, and GTCCRD registers of GPTn together and buffer operation using the GTCCRB, GTCCRE, and GTCCRF registers of GPTn together.

BDn1 Bit (GPTn.GTPR Buffer Operation Disable) (n = 0 to 3)

This bit disables buffer operation using the GTPR, GTPBR, and GTPDBR registers of GPTn together.

BDn2 Bit (GPTn.GTADTR Buffer Operation Disable) (n = 0 to 3)

This bit disables buffer operation using the GTADTRA, GTADTBRA, and GTADTDBRA registers of GPTn together and buffer operation using the GTADTRB, GTADTBRB, and GTADTDBRB registers of GPTn together.

BDn3 Bit (GPTn.GTDV Buffer Operation Disable) (n = 0 to 3)

This bit disables buffer operation using the GTDVU and GTDBU registers of GPTn together and buffer operation using the GTDVD and GTDBD registers of GPTn together.

26.2.11 General PWM Timer Start Write-Protection Register (GTSWP)

Address(es): GPT.GTSWP 000C 2018h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	SWP3	SWP2	SWP1	SWP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SWP0	GTSTR.CST0 Bit Write Disable	0: Writing to the register bit is enabled 1: Writing to the register bit is disabled	R/W
b1	SWP1	GTSTR.CST1 Bit Write Disable	0: Writing to the register bit is enabled 1: Writing to the register bit is disabled	R/W
b2	SWP2	GTSTR.CST2 Bit Write Disable	0: Writing to the register bit is enabled 1: Writing to the register bit is disabled	R/W
b3	SWP3	GTSTR.CST3 Bit Write Disable	0: Writing to the register bit is enabled 1: Writing to the register bit is disabled	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTSWP register enables or disables writing to the GTSTR register to prevent accidental modification.

SWPn Bit (GTSTR.CSTn Bit Write Disable) (n = 0 to 3)

This bit enables or disables writing to the GTSTR.CSTn bit.

When this bit is set to disable writing, writing to the GTSTR.CSTn bit is ignored.

However, if the GTHSCR register is set to start or stop count operation by a hardware source, the status of count operation (started or stopped by a hardware source) is written to the GTSTR.CSTn bit even if the SWPn bit is set to disable writing to GTSTR.CSTn bit.

26.2.12 General PWM Timer I/O Control Register (GTIOR)

Address(es): GPT0.GTIOR 000C 2100h, GPT1.GTIOR 000C 2180h, GPT2.GTIOR 000C 2200h, GPT3.GTIOR 000C 2280h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OBHLD	OBDFLT	GTIOB[5:0]						OAHLD	OADFLT	GTIOA[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	GTIOA[5:0]	GTIOCnA Pin Function Select	See Table 26.4.	R/W
b6	OADFLT	GTIOCnA Pin Output Value Setting at the Count Stop	0: The GTIOCnA pin outputs low when counting is stopped. 1: The GTIOCnA pin outputs high when counting is stopped.	R/W
b7	OAHLD	GTIOCnA Pin Output Setting at the Start/Stop Count	0: The GTIOCnA pin output level at start/stop of counting depends on the register setting. 1: The GTIOCnA pin output level is retained at start/stop of counting.	R/W
b13 to b8	GTIOB[5:0]	GTIOCnB Pin Function Select	See Table 26.4.	R/W
b14	OBDFLT	GTIOCnB Pin Output Value Setting at the Count Stop	0: The GTIOCnB pin outputs low when counting is stopped. 1: The GTIOCnB pin outputs high when counting is stopped.	R/W
b15	OBHLD	GTIOCnB Pin Output Setting at the Start/Stop Count	0: The GTIOCnB pin output level at start/stop of counting depends on the register setting. 1: The GTIOCnB pin output level is retained at start/stop of counting.	R/W

n = 0 to 3

GPTn.GTIOR sets the functions of the GTIOCnA and GTIOCnB pins (n = 0 to 3).
Each channel has one GTIOCnA pin and one GTIOCnB pin. Values written to the GTIOR register of the relevant channel in which write access is disabled by the GTWP.WPn bit are ignored.

GTIOA[5:0] Bits (GTIOCnA Pin Function Select)

These bits select the GTIOCnA pin function. For details, see Table 26.4.

OADFLT Bit (GTIOCnA Pin Output Value Setting at the Count Stop)

This bit sets whether the GTIOCnA pin outputs high or low when counting is stopped.

OAHLD Bit (GTIOCnA Pin Output Setting at the Start/Stop Count)

This bit specifies whether the GTIOCnA pin output level is retained or the level depends on the register setting when counting is started or stopped.
[When the OAHLD bit is set to 0]

- The value specified by the GTIOA[4] bit is output when counting starts.
- The value specified by the OADFLT bit is output when counting stops.
- If the OADFLT bit is modified while counting is stopped, it is immediately reflected in the output.

[When the OAHLD bit is set to 1]

- The output is retained when counting starts or stops.

GTIOB[5:0] Bits (GTIOCnB Pin Function Select)

These bits select the GTIOCnB pin function. For details, see Table 26.4.

OBDFLT Bit (GTIOCnB Pin Output Value Setting at the Count Stop)

This bit sets whether the GTIOCnB pin outputs high or low when counting is stopped.

OBHLD Bit (GTIOCnB Pin Output Setting at the Start/Stop Count)

This bit specifies whether the GTIOCnB pin output level is retained or the level depends on the register setting when counting is started or stopped.

[When the OBHLD bit is set to 0]

- The value specified by the GTIOB[4] bit is output when counting starts.
- The value specified by the OBDFLT bit is output when counting stops.
- If the OBDFLT bit is modified while counting is stopped, it is immediately reflected in the output.

[When the OBHLD bit is set to 1]

- The output is retained when counting starts or stops.

Table 26.4 Settings of GTIOA[5:0] Bits (GTIOB[5:0] Bits) (1/2)

GTIOA[5:0]/GTIOB[5:0] Bits						Function			
b5	**b4**	**b3**	**b2**	**b1**	**b0**	**b5**	**b4**	**b3, b2**	**b1, b0**
0	0	0	0	0	0	Compare match	Initial output is Low.	Output retained at the end of a cycle	Output retained at GPTn.GTCCRA/ GTCCRB compare match
0	0	0	0	0	1				Low output at GPTn.GTCCRA/GTCCRB compare match
0	0	0	0	1	0				High output at GPTn.GTCCRA/GTCCRB compare match
0	0	0	0	1	1				Output toggled at GPTn.GTCCRA/ GTCCRB compare match
0	0	0	1	0	0			Low output at the end of a cycle	Output retained at GPTn.GTCCRA/ GTCCRB compare match
0	0	0	1	0	1				Low output at GPTn.GTCCRA/GTCCRB compare match
0	0	0	1	1	0				High output at GPTn.GTCCRA/GTCCRB compare match
0	0	0	1	1	1				Output toggled at GPTn.GTCCRA/ GTCCRB compare match
0	0	1	0	0	0			High output at the end of a cycle	Output retained at GPTn.GTCCRA/ GTCCRB compare match
0	0	1	0	0	1				Low output at GPTn.GTCCRA/GTCCRB compare match
0	0	1	0	1	0				High output at GPTn.GTCCRA/GTCCRB compare match
0	0	1	0	1	1				Output toggled at GPTn.GTCCRA/ GTCCRB compare match
0	0	1	1	0	0			Output toggled at the end of a cycle	Output retained at GPTn.GTCCRA/ GTCCRB compare match
0	0	1	1	0	1				Low output at GPTn.GTCCRA/GTCCRB compare match
0	0	1	1	1	0				High output at GPTn.GTCCRA/GTCCRB compare match
0	0	1	1	1	1				Output toggled at GPTn.GTCCRA/ GTCCRB compare match

Table 26.4 Settings of GTIOA[5:0] Bits (GTIOB[5:0] Bits) (2/2)

GTIOA[5:0]/GTIOB[5:0] Bits						Function			
b5	b4	b3	b2	b1	b0	b5	b4	b3, b2	b1, b0
0	1	0	0	0	0	Compare match	Initial output is High.	Output retained at the end of a cycle	Output retained at GPTn.GTCCRA/ GTCCRB compare match
0	1	0	0	0	1				Low output at GPTn.GTCCRA/GTCCRB compare match
0	1	0	0	1	0				High output at GPTn.GTCCRA/GTCCRB compare match
0	1	0	0	1	1				Output toggled at GPTn.GTCCRA/ GTCCRB compare match
0	1	0	1	0	0			Low output at the end of a cycle	Output retained at GTPn.GTCCRA/ GTCCRB compare match
0	1	0	1	0	1				Low output at GTPn.GTCCRA/GTCCRB compare match
0	1	0	1	1	0				High output at GPTn.GTCCRA/GTCCRB compare match
0	1	0	1	1	1				Output toggled at GPTn.GTCCRA/ GTCCRB compare match
0	1	1	0	0	0			High output at the end of a cycle	Output retained at GTPn.GTCCRA/ GTCCRB compare match
0	1	1	0	0	1				Low output at GPTn.GTCCRA/GTCCRB compare match
0	1	1	0	1	0				High output at GPTn.GTCCRA/GTCCRB compare match
0	1	1	0	1	1				Output toggled at GPTn.GTCCRA/ GTCCRB compare match
0	1	1	1	0	0			Output toggled at the end of a cycle	Output retained at GTPn.GTCCRA/ GTCCRB compare match
0	1	1	1	0	1				Low output at GPTn.GTCCRA/GTCCRB compare match
0	1	1	1	1	0				High output at GPTn.GTCCRA/GTCCRB compare match
0	1	1	1	1	1				Output toggled at GPTn.GTCCRA/ GTCCRB compare match
1	x	x	x	0	0	Input capture	—	—	Input capture at rising edge
1	x	x	x	0	1				Input capture at falling edge
1	x	x	x	1	0				Input capture at both edges
1	x	x	x	1	1				

x: Don't care

Note: In saw-wave mode, “end of a cycle” refers to an overflow (the value of the GTCNT counter changing from that of the GTPR register to 0000h in up-counting), an underflow (the value of the GTCNT counter changing from 0000h to that of the GTPR register in down-counting), or counter clearing by a hardware source, software, event input, or synchronous clearing. It refers to a trough in triangle-wave mode (the value of the GTCNT counter changing from 0000h to 0001h).

Note: When the timing of an end of a cycle and the timing of a GTCCRA/GTCCRB compare match are the same in a compare-match operation, the b3 and b2 settings are given priority in saw-wave PWM mode, and the b1 and b0 settings are given priority in any other mode.

Note: Even though a compare match is set in GTIOR, output will not be made to the pins. GTONCR needs to be set separately.

26.2.13 General PWM Timer Interrupt Output Setting Register (GTINTAD)

Address(es): GPT0.GTINTAD 000C 2102h, GPT1.GTINTAD 000C 2182h, GPT2.GTINTAD 000C 2202h, GPT3.GTINTAD 000C 2282h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADTRBDEN	ADTRBUEN	ADTRADEN	ADTRAUEN	EINT	—	—	—	GTINTPR[1:0]		GTINTF	GTINTE	GTINTD	GTINTC	GTINTB	GTINTA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	GTINTA	GTCCRA Compare Match/Input Capture Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b1	GTINTB	GTCCRB Compare Match/Input Capture Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b2	GTINTC	GTCCRC Compare Match Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b3	GTINTD	GTCCRD Compare Match Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b4	GTINTE	GTCCRE Compare Match Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b5	GTINTF	GTCCRF Compare Match Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b7, b6	GTINTPR[1:0]	GTPR Compare Match Interrupt Enable	b7 b6 0 0: Interrupt request is disabled. 0 1: In saw-wave mode, interrupt requests are enabled at overflows. In triangle-wave mode, interrupt requests are enabled at crests. 1 0: In saw-wave mode, interrupt requests are enabled at underflows. In triangle-wave mode, interrupt requests are enabled at troughs. 1 1: In saw-wave mode, interrupt requests are enabled at both overflows and underflows. In triangle-wave mode, interrupt requests are enabled at both crests and troughs.	R/W
b10 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	EINT	Dead Time Error Interrupt Enable	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b12	ADTRAUEN	GTADTRA Compare Match (Up-Counting) A/D Converter Start Request Enable	0: A/D converter start request is disabled. 1: A/D converter start request is enabled.	R/W
b13	ADTRADEN	GTADTRA Compare Match (Down-Counting) A/D Converter Start Request Enable	0: A/D converter start request is disabled. 1: A/D converter start request is enabled.	R/W
b14	ADTRBUEN	GTADTRB Compare Match (Up-Counting) A/D Converter Start Request Enable	0: A/D converter start request is disabled. 1: A/D converter start request is enabled.	R/W
b15	ADTRBDEN	GTADTRB Compare Match (Down-Counting) A/D Converter Start Request Enable	0: A/D converter start request is disabled. 1: A/D converter start request is enabled.	R/W

The GTINTAD register enables or disables interrupt requests and A/D converter start requests. Values written to the GTINTAD register of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.

GTINTA Bit (GTCCRA Compare Match/Input Capture Interrupt Enable)

This bit enables or disables interrupt requests by GTCCRA compare match/input capture (GTCIA).

GTINTB Bit (GTCCRB Compare Match/Input Capture Interrupt Enable)

This bit enables or disables interrupt requests by GTCCRB compare match/input capture (GTCIB).

GTINTC Bit (GTCCRC Compare Match Interrupt Enable)

This bit enables or disables interrupt requests by GTCCRC compare match (GTCIC).

GTINTD Bit (GTCCRD Compare Match Interrupt Enable)

This bit enables or disables interrupt requests by GTCCRD compare match (GTCIC).

GTINTE Bit (GTCCRE Compare Match Interrupt Enable)

This bit enables or disables interrupt requests by GTCCRE compare match (GTCIE).

GTINTF Bit (GTCCRF Compare Match Interrupt Enable)

This bit enables or disables interrupt requests by GTCCRF compare match (GTCIE).

GTINTPR[1:0] Bits (GTPR Compare Match Interrupt Enable)

These bits enable or disable interrupt requests by a GTPR compare match (GTCNT counter overflow) and those by a GTCNT counter underflow (GTCIV/GTCIU).

EINT Bit (Dead Time Error Interrupt Enable)

This bit enables or disables interrupt requests by a dead time error (GDTE).

ADTRAUEN Bit (GTADTRA Compare Match (Up-Counting) A/D Converter Start Request Enable)

This bit enables or disables A/D converter start requests generated by GTADTRA compare matches during GTCNT up-counting.

ADTRADEN Bit (GTADTRA Compare Match (Down-Counting) A/D Converter Start Request Enable)

This bit enables or disables A/D converter start requests generated by GTADTRA compare matches during GTCNT down-counting.

ADTRBUEN Bit (GTADTRB Compare Match (Up-Counting) A/D Converter Start Request Enable)

This bit enables or disables A/D converter start requests generated by GTADTRB compare matches during GTCNT up-counting.

ADTRBDEN Bit (GTADTRB Compare Match (Down-Counting) A/D Converter Start Request Enable)

This bit enables or disables A/D converter start requests generated by GTADTRB compare matches during GTCNT down-counting.

26.2.14 General PWM Timer Control Register (GTCR)

Address(es): GPT0.GTCR 000C 2104h, GPT1.GTCR 000C 2184h, GPT2.GTCR 000C 2204h, GPT3.GTCR 000C 2284h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CCLR[1:0]		—	—	TPCS[1:0]		—	—	—	—	—	MD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	MD[2:0]	Mode Select	b2 b0 0 0 0: Saw-wave PWM mode (single buffer or double buffer possible) 0 0 1: Saw-wave one-shot pulse mode (fixed buffer operation) 0 1 0: Setting prohibited 0 1 1: Setting prohibited 1 0 0: Triangle-wave PWM mode 1 (16-bit transfer at crest) (single buffer or double buffer possible) 1 0 1: Triangle-wave PWM mode 2 (16-bit transfer at crest and trough) (single buffer or double buffer possible) 1 1 0: Triangle-wave PWM mode 3 (32-bit transfer at trough) fixed buffer operation) 1 1 1: Setting prohibited	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	TPCS[1:0]	Timer Prescaler Select	b9 b8 0 0: PCLKA 0 1: PCLKA/2 1 0: PCLKA/4 1 1: PCLKA/8	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13, b12	CCLR[1:0]	Count Clear Source Select	b13 b12 0 0: None of the following clearing sources is specified. 0 1: Cleared by GTCCRA input capture 1 0: Cleared by GTCCRB input capture 1 1: Cleared by count clearing in another channel performing synchronous clearing/synchronous operation	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTCR register controls the GTCNT counter. The GTCR register should be set while the GTCNT counter is stopped. Values written to the GTCR register of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.

MD[2:0] Bits (Mode Select)

These bits select the GPT operating mode.

TPCS[1:0] Bits (Timer Prescaler Select)

These bits select a clock for the GTCNT counter. The clock source can be selected independently for each channel.

CCLR[1:0] Bits (Count Clear Source Select)

These bits select a clearing source for the GTCNT counter. The source selected by these bits is added besides counter clearing by a hardware source, software, and an event input.

When clearing in response to input capture is selected, transfer from the buffer registers to the GTCCRA and GTCCRB registers proceeds in response to by the counter clearing and input capture are performed, and other buffer transfer does not proceed. If clearing by a hardware source, software, or an event input is occurs at the same time as clearing by input capture, other buffer transfer proceeds.

When synchronous clearing is selected in saw-wave mode, this is handled as clearing in response to the given counter

overflow or underflow and the pin output and buffer transfers proceed. Even if the GTINTAD.GTINTPR[1:0] bits are set to 01b, 10b, or 11b at this time, the GTCIV or GTCIU interrupt is not requested. In triangle-wave mode, count clearing is only performed and pin output and buffer transfer are not performed. The counter value becomes 0000h, but it is not treated as a trough.

When 01b, 10b, or 11b is selected as a count clear source, count clearing by the source is executed whether the GPTn.GTCNT counter is running (the GTSTR.CSTn bit is 1 (n = 0 to 3)) or stopped (the GTSTR.CSTn bit is 0 (n = 0 to 3)).

When the count direction is down-counting (the GTST.TUCF flag is 0) in saw-wave mode, the GTCNT counter is set to the value set in the GTPR register by executing count clearing. In other cases, the GTCNT counter is set to 0000h.

Do not set the CCLR[1:0] bits for the operating channel to 01b or 10b during synchronous clearing.

26.2.15 General PWM Timer Buffer Enable Register (GTBER)

Address(es): GPT0.GTBER 000C 2106h, GPT1.GTBER 000C 2186h, GPT2.GTBER 000C 2206h, GPT3.GTBER 000C 2286h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	ADTDB	ADTTB[1:0]		—	ADTDA	ADTTA[1:0]		—	CCRSWT	PR[1:0]		CCRB[1:0]		CCRA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CCRA[1:0]	GTCCRA Buffer Operation	b1 b0 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTCCRA ⇔ GTCCRC) 1 x: Double buffer operation (GTCCRA ⇔ GTCCRC ⇔ GTCCRD)	R/W
b3, b2	CCRB[1:0]	GTCCRB Buffer Operation	b3 b2 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTCCRB ⇔ GTCCRE) 1 x: Double buffer operation (GTCCRB ⇔ GTCCRE ⇔ GTCCRF)	R/W
b5, b4	PR[1:0]	GTPR Buffer Operation	b5 b4 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTPBR ⇒ GTPR) 1 x: Double buffer operation (GTPDBR ⇒ GTPBR ⇒ GTPR)	R/W
b6	CCRSWT	GTCCRA and GTCCRB Forcible Buffer Operation	Writing 1 to this bit forcibly performs buffer transfer of GTCCRA and GTCCRB. This bit automatically returns to 0 after the writing of 1. This bit is read as 0.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9, b8	ADTTA[1:0]	GTADTRA Buffer Transfer Timing Select	• Triangle waves b9 b8 0 0: No transfer 0 1: Transfer at crest 1 0: Transfer at trough 1 1: Transfer at both crest and trough • Saw waves b9 b8 0 0: No transfer Values other than 0 0: Transfer at underflow (during down-counting), overflow (during up-counting), or count clearing	R/W
b10	ADTDA	GTADTRA Double Buffer Operation	0: Single buffer operation (GTADTBRA ⇒ GTADTRA) 1: Double buffer operation (GTADTDBRA ⇒ GTADTBRA ⇒ GTADTRA)	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13, b12	ADTTB[1:0]	GTADTRB Buffer Transfer Timing Select	• Triangle waves b13 b12 0 0: No transfer 0 1: Transfer at crest 1 0: Transfer at trough 1 1: Transfer at both crest and trough • Saw waves b13 b12 0 0: No transfer Values other than 0 0: Transfer at underflow (during down-counting), overflow (during up-counting), or count clearing	R/W
b14	ADTDB	GTADTRB Double Buffer Operation	0: Single buffer operation (GTADTBRB ⇒ GTADTRB) 1: Double buffer operation (GTADTDBRB ⇒ GTADTBRB ⇒ GTADTRB)	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The GTBER register makes settings for buffer operation.

The GTBER register should be set while the GTCNT counter is stopped. Values written to the GTBER register of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.

CCRA[1:0] Bits (GTCCRA Buffer Operation)

These bits set buffer operation with the GTCCRA, GTCCRC, and GTCCRD registers combined. When buffer operation is restricted by the operating mode set in the GTCR register, the GTCR setting is given priority.*1

CCRB[1:0] Bits (GTCCRB Buffer Operation)

These bits set buffer operation with the GTCCRB, GTCCRE, and GTCCRF registers combined. When buffer operation is restricted by the operating mode set in the GTCR register, the GTCR setting is given priority.*1

PR[1:0] Bits (GTPR Buffer Operation)

These bits set buffer operation with the GTPR, GTPBR, and GTPDBR registers combined.
Set these bits to 00b when down-counting in saw-wave mode.

CCRSWT Bit (GTCCRA and GTCCRB Forcible Buffer Operation)

Writing 1 to the CCRSWT bit forcibly performs buffer transfer of the GTCCRA and GTCCRB registers. This bit automatically returns to 0 after the writing of 1. This bit is read as 0.
This bit is valid only when counting is stopped with compare match operation specified.

ADTTA[1:0] Bits (GTADTRA Buffer Transfer Timing Select)

These bits set the transfer timing for buffer operation of the GTADTRA, GTADTBRA, and GTADTDBRA registers. When a buffer transfer is performed in saw-wave mode, the source of count clearing is a hardware source, software, event input, or synchronous clearing. During clearing by input capture, buffer transfer does not proceed.

ADTDA Bit (GTADTRA Double Buffer Operation)

These bits set buffer operation with the GTADTRA, GTADTBRA, and GTADTDBRA registers combined.

ADTTB[1:0] Bits (GTADTRB Buffer Transfer Timing Select)

These bits set the transfer timing for buffer operation of the GTADTRB, GTADTBRB, and GTADTDBRB registers. When a buffer transfer is performed in saw-wave mode, the source of count clearing is a hardware source, software, event input, or synchronous clearing. During clearing by input capture, buffer transfer does not proceed.

ADTDB Bit (GTADTRB Double Buffer Operation)

These bits set buffer operation with the GTADTRB, GTADTBRB, and GTADTDBRB registers combined.

Note 1. The buffer operation mode is fixed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3 (32-bit transfer at trough).

26.2.16 General PWM Timer Count Direction Register (GTUDC)

Address(es): GPT0.GTUDC 000C 2108h, GPT1.GTUDC 000C 2188h, GPT2.GTUDC 000C 2208h, GPT3.GTUDC 000C 2288h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UDF	UD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	UD	Count Direction Setting	0: GTCNT counts down. 1: GTCNT counts up.	R/W
b1	UDF	Forcible Count Direction Setting	0: Not forcibly set 1: Forcibly set	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTUDC register sets the direction in which the GTCNT counter counts (up-counting or down-counting). Values written to the GTUDC register of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.

- In saw-wave mode

When the UD value is set to 0 during up-counting, the count direction is changed at an overflow (count clock when GTCNT counter value is equal to the GTPR value). When the UD value is set to 1 during down-counting, the count direction is changed at an underflow (count clock when GTCNT counter value is equal to 0000h).
If the UD value is changed from 1 to 0 with the UDF bit being 0 and while counting is stopped, the counter starts up-counting and the count direction is changed at an overflow (count clock when GTCNT counter value is equal to GTPR value).
If the UD value is changed from 0 to 1 with the UDF bit being 0 and while counting is stopped, the counter starts down-counting and the count direction is changed at an underflow (count clock when GTCNT counter value is equal to 0000h).
When the UDF bit is set to 1 while counting is stopped, the UD bit value at that time is reflected in the count direction when counting starts.

- In triangle-wave mode

Even if the UD value is changed during counting, the change will not be reflected in the count direction.
If the UD value is modified while the UDF bit is 0 and counting is stopped, the change will not be reflected in the count direction when counting starts. If the UDF bit is set to 1 while counting is stopped, the UD bit value at that time is reflected in the count direction when counting starts.

UD Bit (Count Direction Setting)

This bit sets the count direction (up-counting or down-counting) for the GTCNT counter.

UDF Bit (Forcible Count Direction Setting)

This bit forcibly sets the count direction when the GTCNT counter starts operation as the UD value.
Only 0 should be written to this bit during count operation.
When 1 has been written to this bit while counting is stopped, this bit should be returned to 0 before counting starts.

26.2.17 General PWM Timer Interrupt and A/D Converter Start Request Skipping Setting Register (GTITC)

Address(es): GPT0.GTITC 000C 210Ah, GPT1.GTITC 000C 218Ah, GPT2.GTITC 000C 220Ah, GPT3.GTITC 000C 228Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	ADTBL	—	ADTAL	—	IVTT[2:0]			IVTC[1:0]		ITLF	ITLE	ITLD	ITLC	ITLB	ITLA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ITLA	GTCCRA Compare Match/ Input Capture Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b1	ITLB	GTCCRB Compare Match/ Input Capture Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b2	ITLC	GTCCRC Compare Match Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b3	ITLD	GTCCRD Compare Match Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b4	ITLE	GTCCRE Compare Match Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b5	ITLF	GTCCRF Compare Match Interrupt Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b7, b6	IVTC[1:0]	GTCIV/GTCIU Interrupt Skipping Function Select	b7 b6 0 0: Skipping is not performed 0 1: Both overflow and underflow for saw waves and crest for triangle waves are counted and skipped. 1 0: Both overflow and underflow for saw waves and trough for triangle waves are counted and skipped. 1 1: Both overflow and underflow for saw waves and both crest and trough for triangle waves are counted and skipped	R/W
b10 to b8	IVTT[2:0]	GTCIV/GTCIU Interrupt Skipping Count Select	b10 b8 0 0 0: Skipping is not performed 0 0 1: Skipping count of 1 0 1 0: Skipping count of 2 0 1 1: Skipping count of 3 1 0 0: Skipping count of 4 1 0 1: Skipping count of 5 1 1 0: Skipping count of 6 1 1 1: Skipping count of 7	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	ADTAL	GTADTRA A/D Converter Start Request Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	ADTBL	GTADTRB A/D Converter Start Request Link	0: Not linked with GTCIV/GTCIU interrupt skipping function. 1: Linked with GTCIV/GTCIU interrupt skipping function.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

GTITC sets the skipping function for the GTCNT counter overflow (GTPR compare match) interrupt (GTCIV) and underflow interrupt (GTCIU) and also sets whether to link the other interrupts and A/D converter start requests with the GTCIV/GTCIU interrupt skipping function. Note that dead time error interrupts cannot be linked with the GTCIV/GTCIU interrupt skipping function. Values written to the GTITC register of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.

ITLA Bit (GTCCRA Compare Match/Input Capture Interrupt Link)

This bit specifies whether to link the GTCCRA compare match/input capture interrupt (GTCIA) with the GTCIV/GTCIU interrupt skipping function.

ITLB Bit (GTCCRB Compare Match/Input Capture Interrupt Link)

This bit specifies whether to link the GTCCRB compare match/input capture interrupt (GTCIB) with the GTCIV/GTCIU interrupt skipping function.

ITLC Bit (GTCCRC Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRC compare match interrupt (GTCIC) with the GTCIV/GTCIU interrupt skipping function.

ITLD Bit (GTCCRD Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRD compare match interrupt (GTCID) with the GTCIV/GTCIU interrupt skipping function.

ITLE Bit (GTCCRE Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRE compare match interrupt (GTCIE) with the GTCIV/GTCIU interrupt skipping function.

ITLF Bit (GTCCRF Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRF compare match interrupt (GTCIF) with the GTCIV/GTCIU interrupt skipping function.

IVTC[1:0] Bits (GTCIV/GTCIU Interrupt Skipping Function Select)

These bits set the skipping function for the GTPR compare match (GTCNT counter overflow) interrupt (GTCIV) and GTCNT counter underflow interrupt (GTCIU).

IVTT[2:0] Bits (GTCIV/GTCIU Interrupt Skipping Count Select)

These bits set the skipping count for the GTPR compare match (GTCNT counter overflow) interrupt (GTCIV) and GTCNT counter underflow interrupt (GTCIU).
When modifying the IVTT[2:0] bits, first set the IVTC[1:0] bits to 00b.

ADTAL Bit (GTADTRA A/D Converter Start Request Link)

This bit specifies whether to link the GTADTRA A/D converter start request with GTCIVn/GTCIUn interrupt skipping function (n = 0 to 3).

ADTBL Bit (GTADTRB A/D Converter Start Request Link)

This bit specifies whether to link the GTADTRB A/D converter start request with GTCIVn/GTCIUn interrupt skipping function (n = 0 to 3).

26.2.18 General PWM Timer Status Register (GTST)

Address(es): GPT0.GTST 000C 210Ch, GPT1.GTST 000C 218Ch, GPT2.GTST 000C 220Ch, GPT3.GTST 000C 228Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TUCF	—	—	—	DTEF	ITCNT[2:0]			—	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	The read value is undefined. The write value should be 1.	R/W
b10 to b8	ITCNT[2:0]	GTCIV/GTCIU Interrupt Skipping Count Counter	Counter for counting the number of times a timer interrupt has been skipped.	R
b11	DTEF	Dead Time Error Flag	0: No dead time error has occurred. 1: A dead time error has occurred.	R
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	TUCF	Count Direction Flag	0: The GTCNT counter counts downward. 1: The GTCNT counter counts upward.	R

GTST indicates the status of the GPT. Values written to the GTST register of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.

ITCNT[2:0] Bits (GTCIV/GTCIU Interrupt Skipping Count Counter)

When the GTCIV/GTCIU interrupt skipping function is used (the GTITC.IVTC[1:0] bits are set to a value other than 00b), the counter is incremented by 1 every time the GTCIV/GTCIU interrupt source is generated.
[Clearing conditions]

- The GTCIV/GTCIU interrupt skipping function is not used (the GTITC.IVTT[2:0] bits are 000b when the GTITC.IVTC[1:0] bits are 00b).
- The GTCIV/GTCIU interrupt skipping count matches the specified count (the ITCNT[2:0] bits match the skipping count specified by the GTITC.IVTT[2:0] bits).

DTEF Flag (Dead Time Error Flag)

This flag indicates that the timer output toggle point after the automatic addition of dead time has exceeded the count period.
This flag returns to 0 when the timer output toggle point after the automatic addition of dead time is back within the period. This flag can only be read from. (Writing 0 to clear the flag is not allowed.)
When an interrupt by the DTEF flag is enabled (the GTINTAD.EINT bit is 1), a GDTE interrupt is generated every time the DTEF flag changes from 0 to 1.
[Setting condition]

- When a change point of the waveform after the automatic setting of dead time has exceeded the count period

[Clearing condition]

- The timer output toggle point after the automatic addition of dead time is within the count period.

TUCF Flag (Count Direction Flag)

This flag indicates the count direction of the GTCNT counter.
When the GTUDC.UDF bit is set to 1 while the count operation is stopped, the value of the GTUDC.UD bit at that time is set to the flag.
In triangle-wave mode whether count operation is stopped or in progress, the flag is set to 1 by clearing (a hardware

source, software, event input, input capture, or synchronous clearing). In saw-wave mode, the TUCF flag is not updated by clearing.
In saw-wave mode during count operation, the value of the GTUDC.UD bit is set to the TUCF flag at an overflow in up-counting or an underflow in down-counting. In triangle-wave mode, the flag is set to 0 at a crest, to 1 at a trough, and to 0 when a value larger than the value of the GTCNT counter is set to the GTPR register.

26.2.19 General PWM Timer Counter (GTCNT)

Address(es): GPT0.GTCNT 000C 210Eh, GPT1.GTCNT 000C 218Eh, GPT2.GTCNT 000C 220Eh, GPT3.GTCNT 000C 228Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The GTCNT counter is a 16-bit readable/writable counter. There is one GTCNT counter for each channel. The GTCNT counter can be written only when count operation is stopped and cannot be written during count operation.
Access in 8-bit units to the GTCNT counter is prohibited, and it should be accessed in 16-bit units. Values written to the GTCNT counter of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.

26.2.20 General PWM Timer Compare Capture Register m (GTCCRm) (m = A to F)

Address(es): GPT0.GTCCRA 000C 2110h, GPT1.GTCCRA 000C 2190h, GPT2.GTCCRA 000C 2210h, GPT3.GTCCRA 000C 2290h,
GPT0.GTCCRB 000C 2112h, GPT1.GTCCRB 000C 2192h, GPT2.GTCCRB 000C 2212h, GPT3.GTCCRB 000C 2292h,
GPT0.GTCCRC 000C 2114h, GPT1.GTCCRC 000C 2194h, GPT2.GTCCRC 000C 2214h, GPT3.GTCCRC 000C 2294h,
GPT0.GTCCRD 000C 2116h, GPT1.GTCCRD 000C 2196h, GPT2.GTCCRD 000C 2216h, GPT3.GTCCRD 000C 2296h,
GPT0.GTCCRE 000C 2118h, GPT1.GTCCRE 000C 2198h, GPT2.GTCCRE 000C 2218h, GPT3.GTCCRE 000C 2298h,
GPT0.GTCCRF 000C 211Ah, GPT1.GTCCRF 000C 219Ah, GPT2.GTCCRF 000C 221Ah, GPT3.GTCCRF 000C 229Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTCCRm registers are 16-bit readable/writable registers. There are six GTCCRm registers for each channel.
The GTCCRA and GTCCRB registers are registers used for both output compare and input capture.
The GTCCRC and GTCCRE registers are compare match registers, and can also function as buffer registers for the GTCCRA and GTCCRB registers.
The GTCCRD and GTCCRF registers are compare match registers, and can also function as buffer registers for the GTCCRC and GTCCRE registers (double buffer registers for the GTCCRA and GTCCRB registers). Values written to the GTCCRm register of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.

26.2.21 General PWM Timer Period Setting Register (GTPR)

Address(es): GPT0.GTPR 000C 211Ch, GPT1.GTPR 000C 219Ch, GPT2.GTPR 000C 221Ch, GPT3.GTPR 000C 229Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTPR register is a 16-bit readable/writable register that sets the maximum count value of the GTCNT counter. There is one GTPR register for each channel.
In saw-wave mode, the value of (GTPR + 1) is the count period. In triangle-wave mode, the value of (GTPR value × 2) is the count period. Values written to the GTPR register of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.
Do not modify the GTPR register during down-counting in saw-wave mode.

26.2.22 General PWM Timer Period Setting Buffer Register (GTPBR)

Address(es): GPT0.GTPBR 000C 211Eh, GPT1.GTPBR 000C 219Eh, GPT2.GTPBR 000C 221Eh, GPT3.GTPBR 000C 229Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTPBR register is a 16-bit readable/writable register that functions as a buffer register for the GTPR register. There is one GTPBR register for each channel. Values written to the GTPBR register of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.

26.2.23 General PWM Timer Period Setting Double Buffer Register (GTPDBR)

Address(es): GPT0.GTPDBR 000C 2120h, GPT1.GTPDBR 000C 21A0h, GPT2.GTPDBR 000C 2220h, GPT3.GTPDBR 000C 22A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTPDBR register is a 16-bit readable/writable register that functions as a buffer register for the GTPBR register (a double buffer register for the GTPR register). There is one GTPDBR register for each channel. Values written to the GTPDBR register of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.

26.2.24 A/D Converter Start Request Timing Register m (GTADTRm) (m = A, B)

Address(es): GPT0.GTADTRA 000C 2124h, GPT1.GTADTRA 000C 21A4h, GPT2.GTADTRA 000C 2224h, GPT3.GTADTRA 000C 22A4h, GPT0.GTADTRB 000C 212Ch, GPT1.GTADTRB 000C 21Ach, GPT2.GTADTRB 000C 222Ch, GPT3.GTADTRB 000C 22ACh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTADTRm registers are 16-bit readable/writable registers that set the timing of A/D converter start request generation. When the GTADTRm value matches the GTCNT counter value, an A/D converter start request is generated. There are two GTADTRm registers for each channel.

Access in 8-bit units to the GTADTRm registers is prohibited, and they should be accessed in 16-bit units. Values written to the GTADTRm register of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.

26.2.25 A/D Converter Start Request Timing Buffer Register m (GTADTBRm) (m = A, B)

Address(es): GPT0.GTADTBRA 000C 2126h, GPT1.GTADTBRA 000C 21A6h, GPT2.GTADTBRA 000C 2226h, GPT3.GTADTBRA 000C 22A6h, GPT0.GTADTBRB 000C 212Eh, GPT1.GTADTBRB 000C 21AEh, GPT2.GTADTBRB 000C 222Eh, GPT3.GTADTBRB 000C 22AEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTADTBRm registers are 16-bit readable/writable registers that function as buffer registers for the GTADTRm register. There are two GTADTBRm registers for each channel.

Access in 8-bit units to the GTADTBRm registers is prohibited, and they should be accessed in 16-bit units. Values written to the GTADTBRm register of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.

26.2.26 A/D Converter Start Request Timing Double Buffer Register m (GTADTDBRm) (m = A, B)

Address(es): GPT0.GTADTDBRA 000C 2128h, GPT1.GTADTDBRA 000C 21A8h, GPT2.GTADTDBRA 000C 2228h, GPT3.GTADTDBRA 000C 22A8h, GPT0.GTADTDBRB 000C 2130h, GPT1.GTADTDBRB 000C 21B0h, GPT2.GTADTDBRB 000C 2230h, GPT3.GTADTDBRB 000C 22B0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTADTDBRm registers are 16-bit readable/writable registers that function as buffer registers for the GTADTBRm registers (double buffer registers for the GTADTRm registers). There are two GTADTDBRm registers for each channel.

Access in 8-bit units to the GTADTDBRm registers is prohibited, and they should be accessed in 16-bit units. Values written to the GTADTDBRm register of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.

26.2.27 General PWM Timer Output Negate Control Register (GTONCR)

Address(es): GPT0.GTONCR 000C 2134h, GPT1.GTONCR 000C 21B4h, GPT2.GTONCR 000C 2234h, GPT3.GTONCR 000C 22B4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OBE	OAE	—	SWN	—	—	—	NFV	NFS[3:0]				NVB	NVA	NEB	NEA
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NEA	GTIOCnA Pin Negate Control Enable	0: Negate is disabled 1: Negate is enabled	R/W
b1	NEB	GTIOCnB Pin Negate Control Enable	0: Negate is disabled 1: Negate is enabled	R/W
b2	NVA	GTIOCnA Pin Negate Value Setting	0: GTIOCnA pin is set to 0 when negate control is performed. 1: GTIOCnA pin is set to 1 when negate control is performed.	R/W
b3	NVB	GTIOCnB Pin Negate Value Setting	0: GTIOCnB pin is set to 0 when negate control is performed. 1: GTIOCnB pin is set to 1 when negate control is performed.	R/W
b7 to b4	NFS[3:0]	GTIOC Output Negate Source Select	b7 b4 0 1 1 1: GTETRG pin input 1 x x x: Software control (control through SWN bit) Settings other than the above are prohibited when negate control is enabled by the NEA or NEB bit.	R/W
b8	NFV	Negate Source Polarity Select	0: Negate control is provided when the negate source has become 0. 1: Negate control is provided when the negate source has become 1.	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	SWN	Software Negate Control	When NFV bit is 0: 0: Negate control is provided. 1: Negate control is not provided. When NFV bit is 1: 0: Negate control is not provided. 1: Negate control is provided.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	OAE	GTIOCnA Pin Output Enable	0: Output is disabled 1: Output is enabled	R/W
b15	OBE	GTIOCnB Pin Output Enable	0: Output is disabled 1: Output is enabled	R/W

n = 0 to 3

The GTONCR register controls negate of the GTIOCnA pin output and GTIOCnB pin output. Values written to the GTONCR register of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.

NEA Bit (GTIOCnA Pin Negate Control Enable)

This bit enables negate control of the GTIOCnA pin output.

NEB Bit (GTIOCnB Pin Negate Control Enable)

This bit enables negate control of the GTIOCnB pin output.

NVA Bit (GTIOCnA Pin Negate Value Setting)

This bit sets the output value of the GTIOCnA pin at negate control.

NVB Bit (GTIOCnB Pin Negate Value Setting)

This bit sets the output value of the GTIOCnB pin at negate control.

NFS[3:0] Bits (GTIOC Output Negate Source Select)

These bits select the negate source for the GTIOCnA pin output and GTIOCnB pin output.
If negate control is not enabled by the NEA or NEB bit, it is not necessary to change the value of these bits from their initial value.

NFV Bit (Negate Source Polarity Select)

This bit selects the negate source polarity for the GTIOCnA pin output and GTIOCnB pin output.

SWN Bit (Software Negate Control)

This bit specifies whether to provide negate control for the GTIOCnA pin output and GTIOCnB pin output.
This bit setting is valid when software control is selected as the negate source (NFS[3] bit is set to 1).

OAE Bit (GTIOCnA Pin Output Enable)

This bit selects whether to output the GTIOCnA pin output. This bit setting is valid only when compare match has been set (the GTIOR.GTIOA[5] bit is 0).

OBE Bit (GTIOCnB Pin Output Enable)

This bit selects whether to output the GTIOCnB pin output.This bit setting is valid only when compare match has been set (the GTIOR.GTIOB[5] bit is 0).

26.2.28 General PWM Timer Dead Time Control Register (GTDTCR)

Address(es): GPT0.GTDTCR 000C 2136h, GPT1.GTDTCR 000C 21B6h, GPT2.GTDTCR 000C 2236h, GPT3.GTDTCR 000C 22B6h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TDFER	—	—	TDBDE	TDBUE	—	—	—	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TDE	Negative-Phase Waveform Setting	0: GTCCRB is set without using GTDVU and GTDVD. 1: GTDVU and GTDVD are used to set the compare match value for negative-phase waveform with dead time automatically in GTCCRB.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	TDBUE	GTDVU Buffer Operation Enable	0: GTDVU buffer operation is disabled 1: GTDVU buffer operation is enabled	R/W
b5	TDBDE	GTDVD Buffer Operation Enable	0: GTDVD buffer operation is disabled 1: GTDVD buffer operation is enabled	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TDFER	GTDVD Setting	0: GTDVU and GTDVD are set separately. 1: The value written to GTDVU is automatically set to GTDVD.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTDTCR register enables automatic setting of a compare match value for negative-phase waveform with dead time. Values written to the GTDTCR register of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.

TDE Bit (Negative-Phase Waveform Setting)

This bit sets whether to use the GTDVU and GTDVD registers. When the GTDVU and GTDVD registers are used, the compare match value for a negative-phase waveform with dead time that was obtained by the compare match value of a positive-phase waveform (GTCCRA) and the dead time value (GTDVU and GTDVD) is automatically set in the GTCCRB register.

This bit is enabled in saw-wave one-shot pulse mode and all triangle-wave PWM modes. The TDE bit setting is ignored in saw-wave PWM mode, and automatic setting does not take place.

The automatically set GTCCRB value has the following upper and lower limit values.

- Saw-wave one-shot pulse mode
  Upper limit value: the value set in the GTPR register
  Lower limit value: 0000h
- Triangle wave PWM mode
  Upper limit value: the value set in the GTPR register - 1
  Lower limit value: 0001h in up-counting, 0000h in down-counting

If the obtained GTCCRB value is not within the range between the upper and lower limits, the upper or lower limit is set in the GTCCRB register, and the GTST.DTEF flag becomes 1. However, if the obtained GTCCRB value exceeds the upper limit in triangle-wave PWM mode, the GTST.DTEF flag becomes 0.

TDBUE Bit (GTDVU Buffer Operation Enable)

This bit enables buffer operation with the GTDVU and GTDBU registers combined.

The buffer transfer timing is at an overflow or underflow for saw waves, and the trough for triangle waves.

TDBDE Bit (GTDVD Buffer Operation Enable)

This bit enables buffer operation with the GTDVD and GTDBD registers combined.
The buffer transfer timing is at an overflow or underflow for saw waves, and the trough for triangle waves.
When this bit and the TDFER bit are set to 1 simultaneously, the TDFER bit setting is given priority.

TDFER Bit (GTDVD Setting)

This bit sets whether or not the value written to the GTDVU register is also set to the GTDVD register automatically.

26.2.29 General PWM Timer Dead Time Value Register m (GTDVm) (m = U, D)

Address(es): GPT0.GTDVU 000C 2138h, GPT1.GTDVU 000C 21B8h, GPT2.GTDVU 000C 2238h, GPT3.GTDVU 000C 22B8h, GPT0.GTDVD 000C 213Ah, GPT1.GTDVD 000C 21Bah, GPT2.GTDVD 000C 223Ah, GPT3.GTDVD 000C 22BAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTDVm registers are 16-bit readable/writable registers that set the dead time for generating PWM waveforms with dead time.
There are two GTDVm registers for each channel: the GTDVU register used for up-counting and the GTDVD register used for down-counting.
When using the dead-time automatic setting function, do not set a value that makes a change point of the waveform exceeding the count period. The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. When using the GTDVm register, writing to the GTCCRB register is prohibited. When setting this register to 0000h, waveforms without dead time are output.
Access in 8-bit units to the GTDVm registers is prohibited, and they should be accessed in 16-bit units. Values written to the GTDVm register of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.

26.2.30 General PWM Timer Dead Time Buffer Register m (GTDBm) (m = U, D)

Address(es): GPT0.GTDBU 000C 213Ch, GPT1.GTDBU 000C 21BCh, GPT2.GTDBU 000C 223Ch, GPT3.GTDBU 000C 22BCh, GPT0.GTDBD 000C 213Eh, GPT1.GTDBD 000C 21Beh, GPT2.GTDBD 000C 223Eh, GPT3.GTDBD 000C 22BEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTDBm registers are 16-bit readable/writable registers that function as buffer registers for the GTDVm registers.
There are two GTDBm registers for each channel.
Values written to the GTDBm register of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.

26.2.31 General PWM Timer Output Protection Function Status Register (GTSOS)

Address(es): GPT0.GTSOS 000C 2140h, GPT1.GTSOS 000C 21C0h, GPT2.GTSOS 000C 2240h, GPT3.GTSOS 000C 22C0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOS[1:0]	
Value after reset:	0	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SOS[1:0]	Output Protection Function Status	b1 b0 0 0: Normal operation 0 1: Protected state (GTCCRA = 0 is set during transfer at trough or crest) 1 0: Protected state (GTCCRA ≥ GTPR is set during transfer at trough) 1 1: Protected state (GTCCRA ≥ GTPR is set during transfer at crest)	R
b7 to b2	—	Reserved	These bits are read as 0 and cannot be modified.	R
b9, b8	—	Reserved	The read value is undefined. These bits cannot be modified.	R
b15 to b10	—	Reserved	These bits are read as 0 and cannot be modified.	R

The GTSOS register is a status register that indicates the status of the output protection function. The output protection function is enabled only when the dead time is automatically set (the GTDTCR.TDE bit is 1) in triangle-wave mode.

SOS[1:0] Bits (Output Protection Function Status)

This bit indicates the status of the output protection function in triangle-wave PWM mode. For details of the output protection function, see section 26.7.4, Output Protection Function for GTIOC Pin Output.

26.2.32 General PWM Timer Output Protection Function Temporary Release Register (GTSOTR)

Address(es): GPT0.GTSOTR 000C 2142h, GPT1.GTSOTR 000C 21C2h, GPT2.GTSOTR 000C 2242h, GPT3.GTSOTR 000C 22C2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOTR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SOTR	Output Protection Function Temporary Release	0: Protected state is not released 1: Protected state is released	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTSOTR register temporarily releases the protected state of GTIOCnB pin output when output protection has been set.

The protected state can be released only for the case of the GTSOS.SOS[1:0] bits are 10b (protected state in which GTCCRA ≥ GTPR has occurred during transfer at trough). The protected state cannot be released for any other case.

Values written to the GTSOTR register of the relevant channel in which write access is disabled by the GTWP.WPn bit (n = 0 to 3) are ignored.

SOTR Bit (Output Protection Function Temporary Release)

This bit sets whether to temporarily release the protected state of the GTIOCnB pin output in an output protected state. After the SOTR bit has been set to 1, the output protection function is canceled from the first trough. After the SOTR bit has been set to 0, output protection is resumed from the first trough.

26.3 Operation

26.3.1 Basic Operation

Each channel has a GTCNT counter, GTPR register, and GTCCRm register (m = A to F). The GTCNT counter performs up-counting, down-counting, or up-/down-counting, available for periodic count operation. The timer period is set by the GTPR register.
In this section, up-counting and down-counting operation is referred to as saw-wave (half-wave) operation. Up-/down-counting operation is referred to as triangle-wave (full-wave) operation.

26.3.1.1 Counter Operation

(1) Periodic Count Operation (in Up-Counting)

The GTCNT counter in each channel starts up-counting when the corresponding GTSTS.CSTn bit is set to 1. When the GTCNT counter value changes from the GTPR register value to 0000h (overflow), a GTCIV interrupt is requested if the GTINTAD.GTINTPR[0] bit is 1. After GTCNT overflows, up-counting is resumed from 0000h.
Figure 26.2 shows an example of periodic count operation in up-counting.

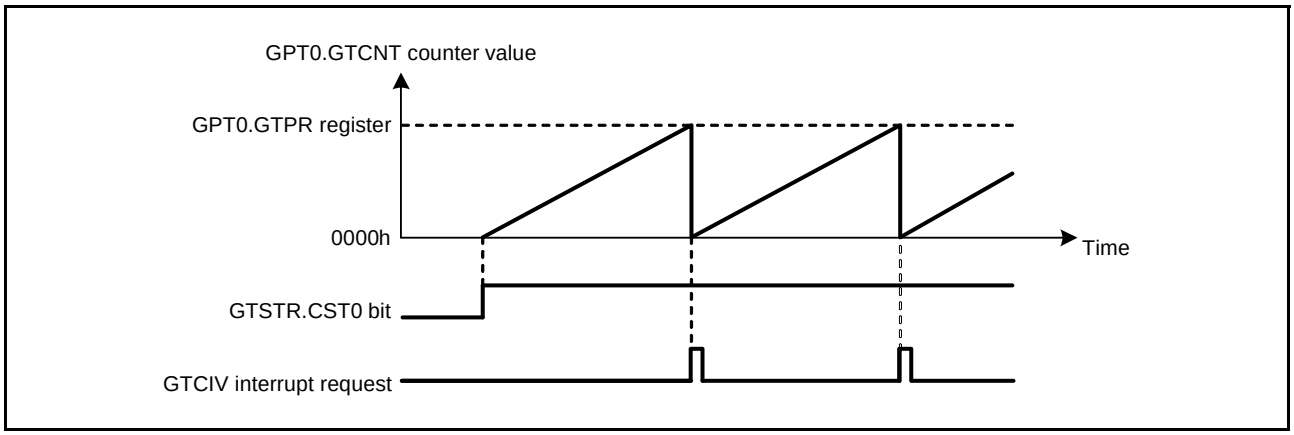


Figure 26.2 Example of Periodic Count Operation (in Up-Counting)

Figure 26.3 shows an example for setting periodic count operation in up-counting.

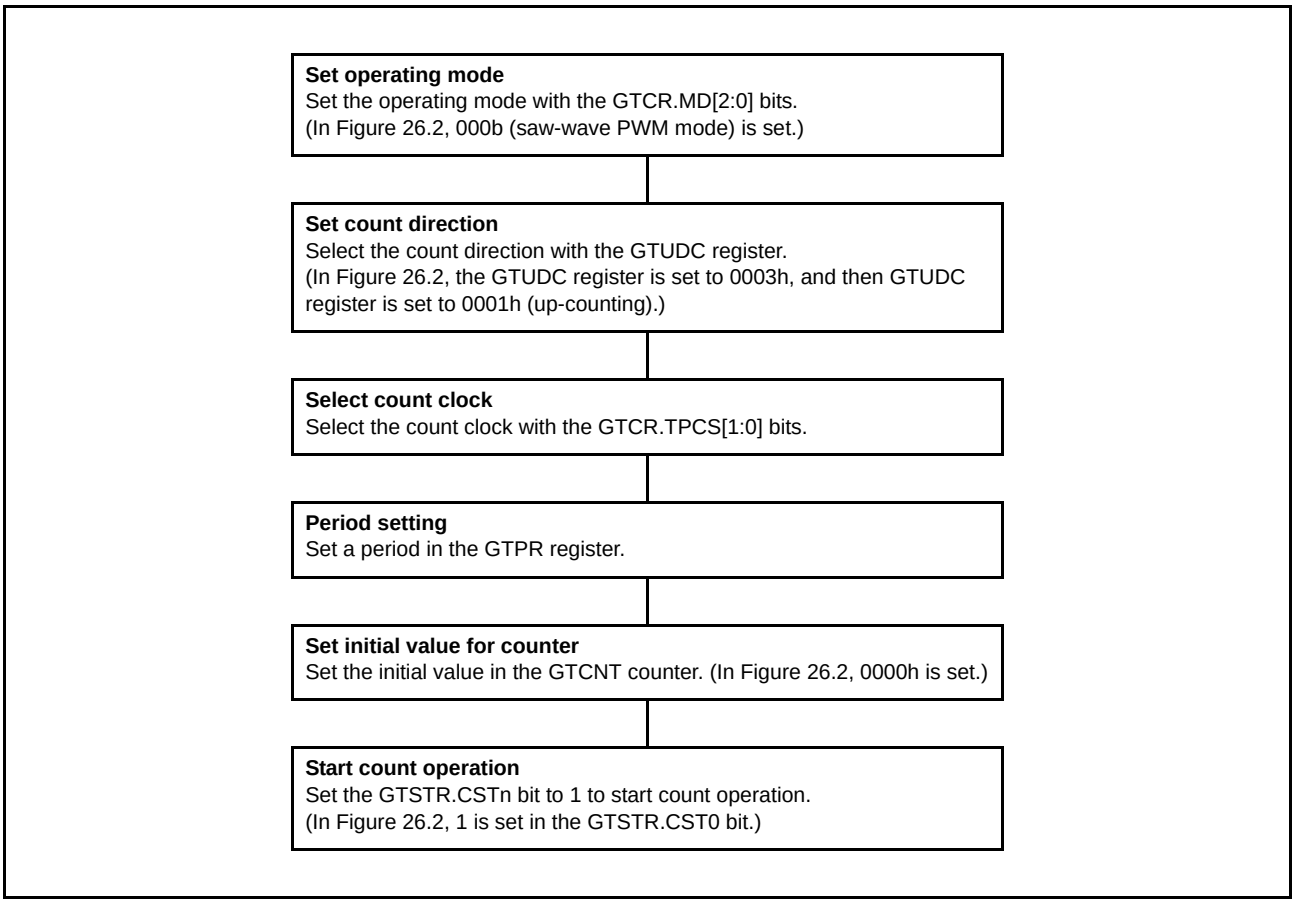


Figure 26.3 Example for Setting Periodic Count Operation (in Up-Counting)

(2) Periodic Count Operation (in Down-Counting)

The GTCNT counter in each channel can perform down-counting by setting GTUDC.
When the GTCNT counter value changes from 0000h to the GTPR register value (underflow), a GTCIU interrupt is requested if the GTINTAD.GTINTPR[1] bit is 1. After the GTCNT counter underflows, down-counting is resumed from the GTPR value.
Figure 26.4 shows an example of periodic count operation in down-counting.

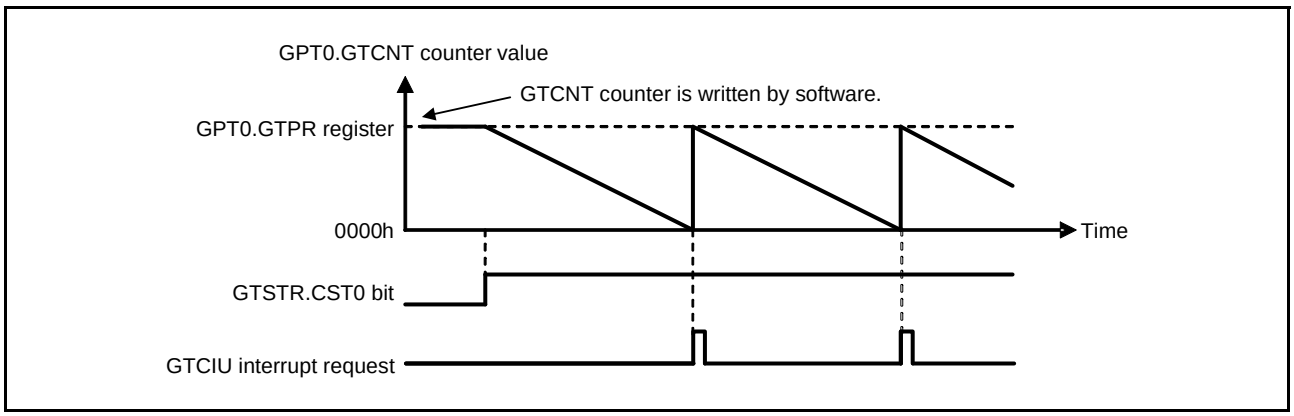


Figure 26.4 Example of Periodic Count Operation (in Down-Counting)

Figure 26.5 shows an example for setting periodic count operation in down-counting.

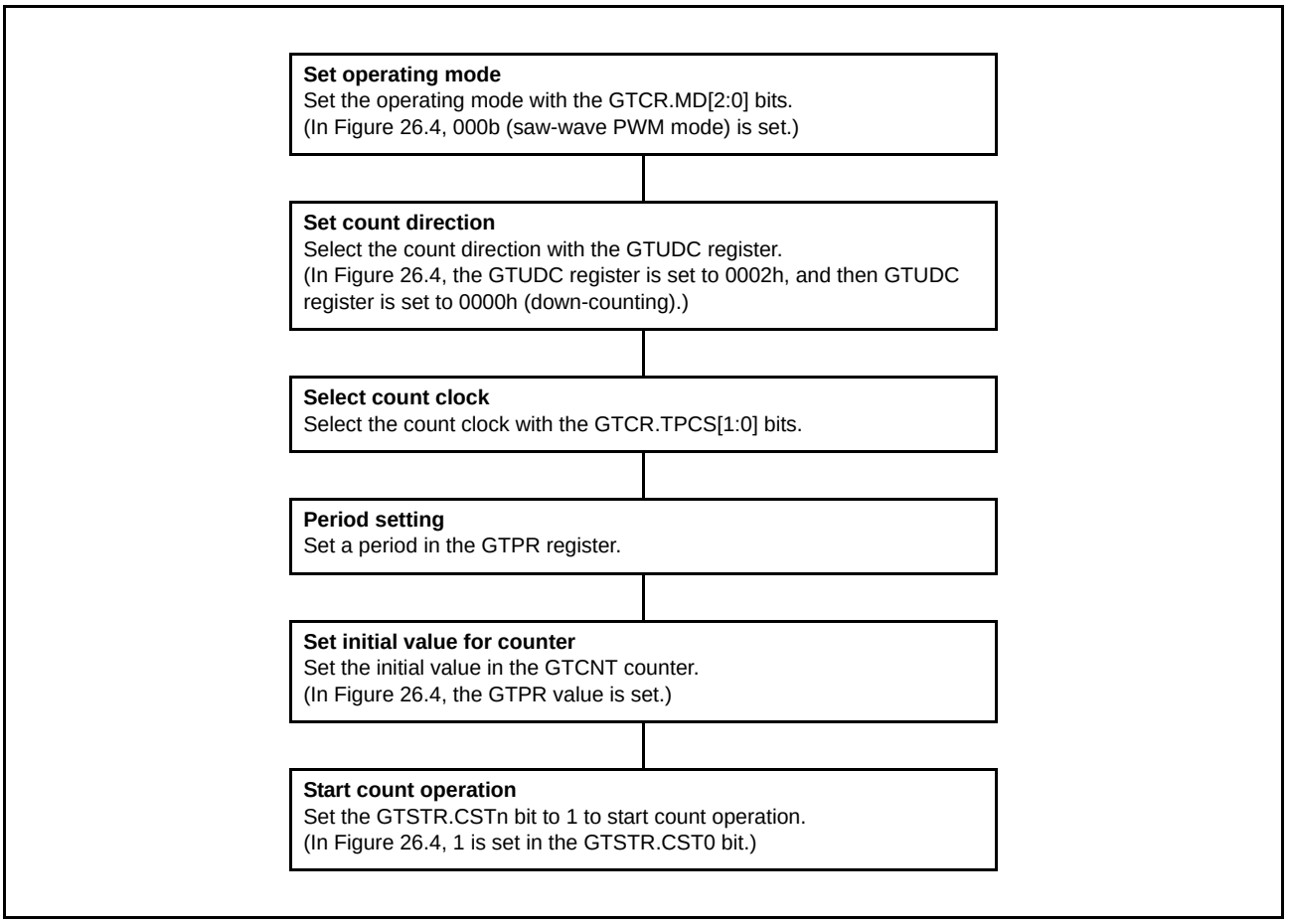


Figure 26.5 Example for Setting Periodic Count Operation (in Down-Counting)

26.3.1.2 Waveform Output by Compare Match

Compare match refers to when the GPTn.GTCNT counter value matches the GPTn.GTCCRA or GPTn.GTCCRB register value (n = 0 to 3). Low, high, or toggle output can be performed from the GTIOCnA or GTIOCnB pin in synchronization with the count clock after a compare match occurs.
In addition, the GTIOCnA or GTIOCnB pin output can be low, high, or toggled at the “end of the cycle” which is determined by the GPTn.GTPR register.
The end of the cycle is:

- For saw waves in up-counting: When the GPTn.GTCNT counter value changes from the GPTn.GTPR register value to 0 (overflow)
- For saw waves in down-counting: When the GPTn.GTCNT counter value changes from 0000h to the GPTn.GTPR register value (underflow)
- For triangle waves: When the GPTn.GTCNT counter value changes from 0000h to 0001h (trough)

(1) Low Output and High Output

Figure 26.6 shows an example of low output and high output operation by a compare match of GTCCRA and GTCCRB.
In this example, the GPT0.GTCNT counter performs up-counting, and settings have been made so that high is output from the GTIOC0A pin by a GPT0.GTCCRA compare match, and low is output from the GTIOC0B pin by a GPT0.GTCCRB compare match. The pin level does not change when the specified level and pin level match.

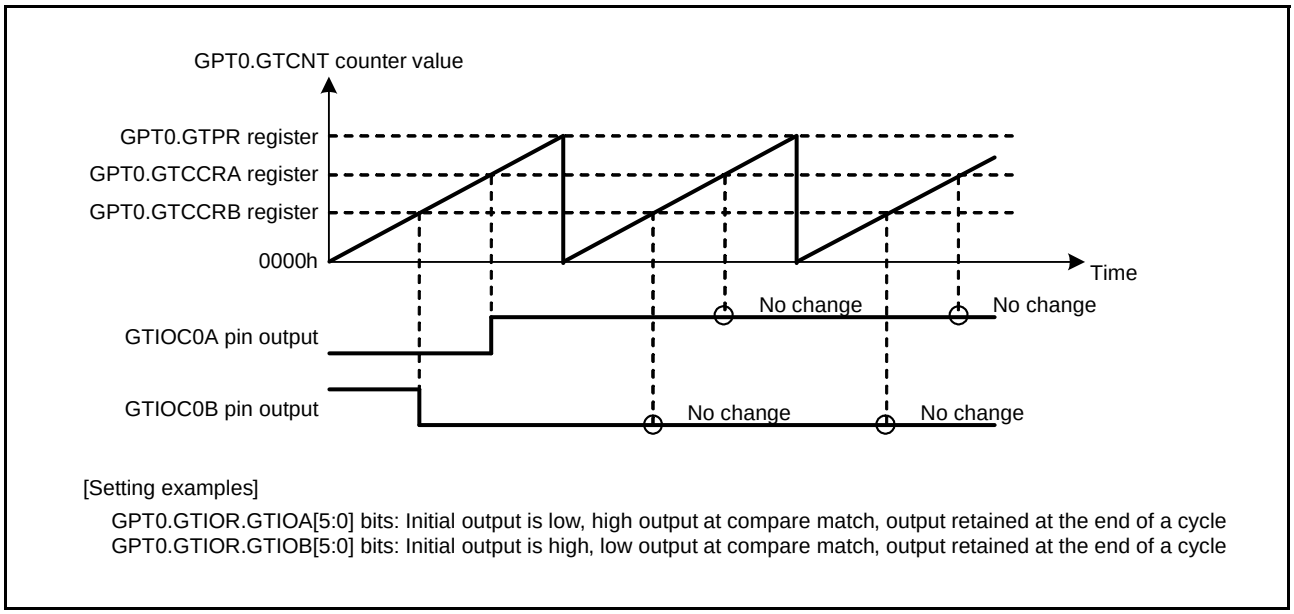


Figure 26.6 Example of Low Output and High Output Operation

Figure 26.7 shows an example for setting low output and high output operation.

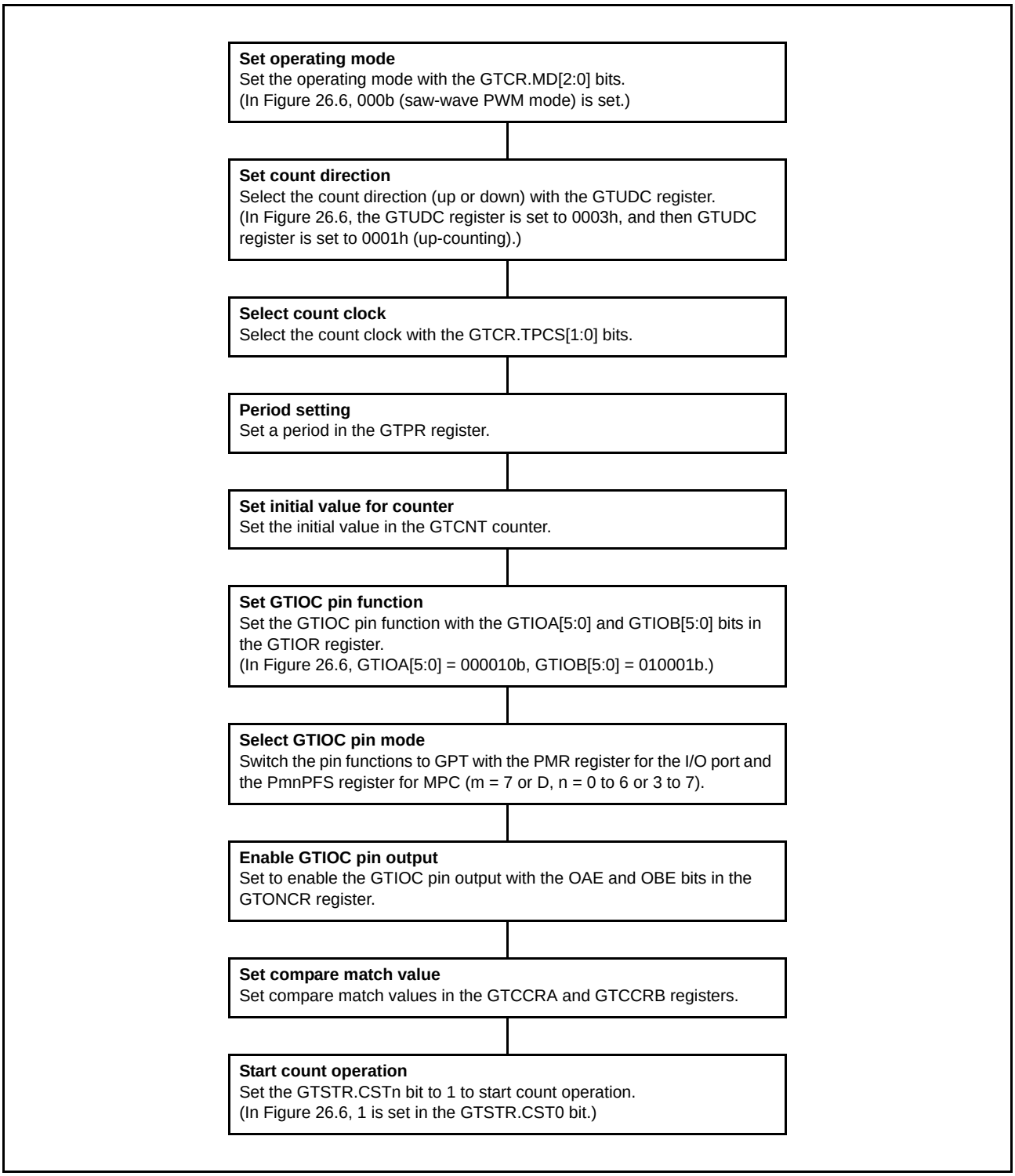
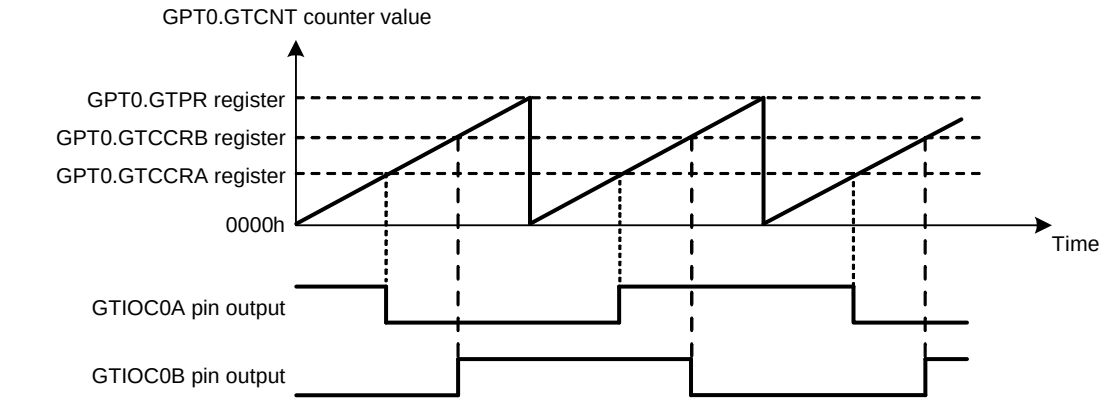


Figure 26.7 Example for Setting Low Output and High Output Operation

(2) Toggled Output

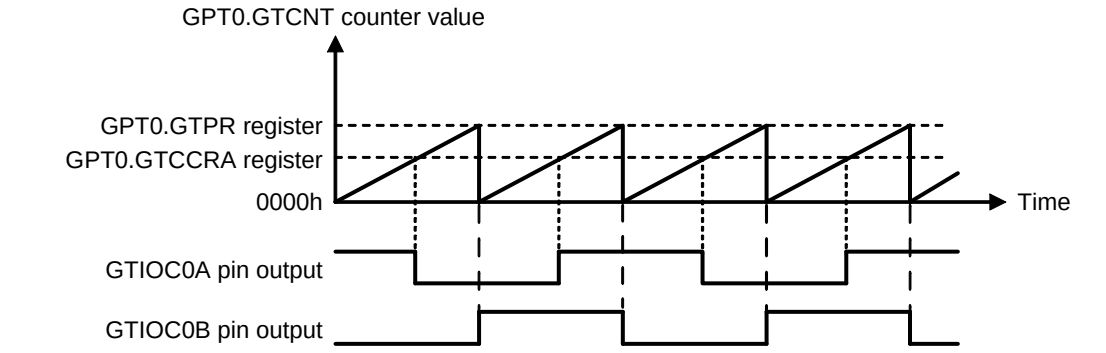
Figure 26.8 and Figure 26.9 show examples of toggled output operation by compare matches of GTCCRA and GTCCRB. In Figure 26.8, the GPT0.GTCNT counter performs up-counting, and settings have been made so that the GTIOC0A pin output by a GPT0.GTCCRA compare match and GTIOC0B pin output by a GPT0n.GTCCRB compare match are toggled.

In Figure 26.9, the GPT0.GTCNT counter performs up-counting, and settings have been made so that the GTIOC0A output is toggled by a compare match of GPT0.GTCCRA and the GTIOC0B output is toggled at the end of the cycle.



[Setting examples]
GPT0.GTIOR.GTIOA[5:0] bits: Initial output is high, output toggled at compare match, output retained at the end of a cycle
GPT0.GTIOR.GTIOB[5:0] bits: Initial output is low, output toggled at compare match, output retained at the end of a cycle

Figure 26.8 Example of Toggled Output Operation (1)



[Setting examples]
GPT0.GTIOR.GTIOA[5:0] bits: Initial output is high, output toggled at compare match, output retained at the end of a cycle
GPT0.GTIOR.GTIOB[5:0] bits: Initial output is low, output retained at compare match, output toggled at the end of a cycle

Figure 26.9 Example of Toggled Output Operation (2)

Figure 26.10 shows an example for setting toggled output operation.

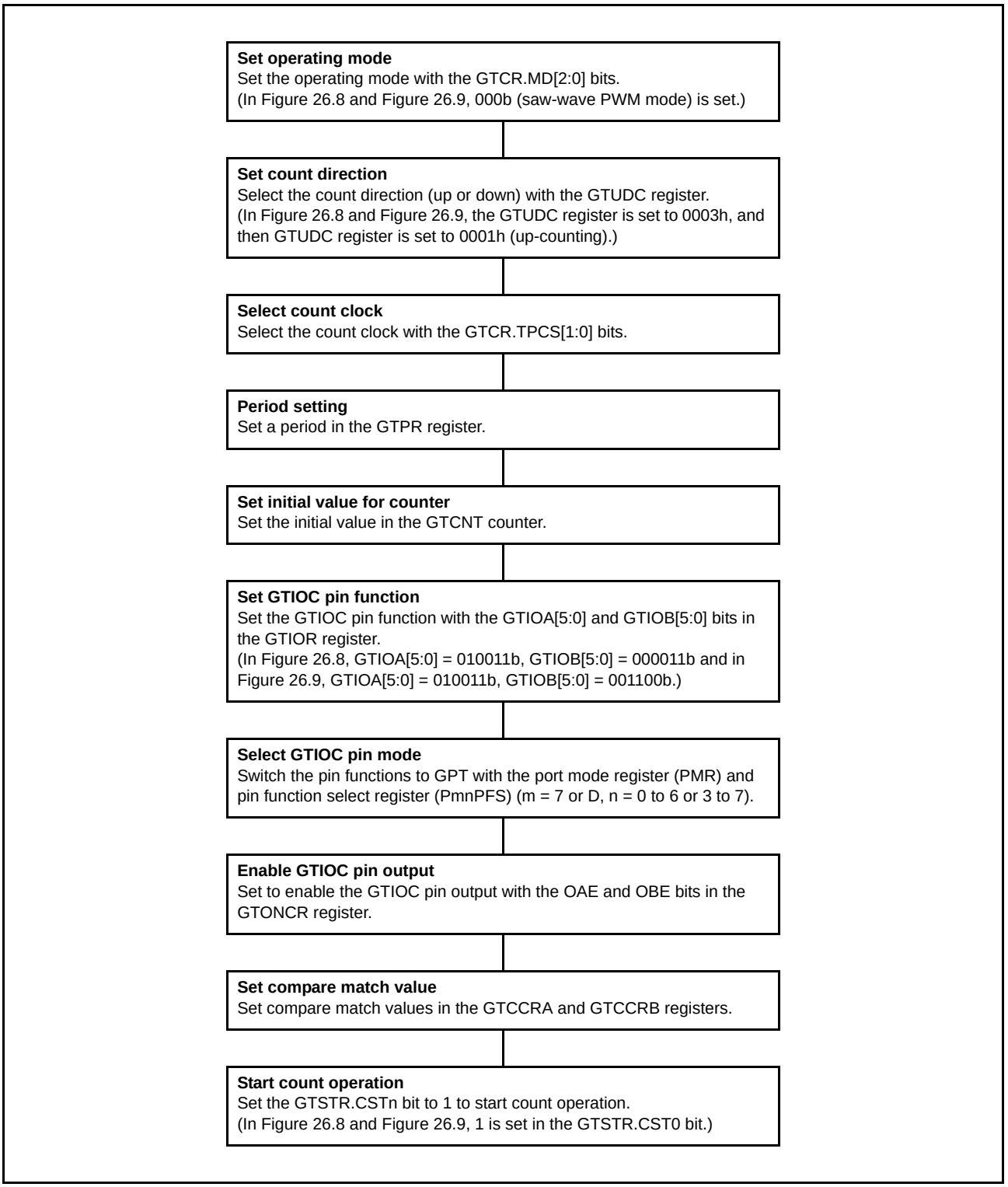


Figure 26.10 Example for Setting Toggled Output Operation

26.3.1.3 Input Capture Function

The GPTn.GTCNT counter value can be transferred to either GPTn.GTCCRA or GPTn.GTCCRB on detection of the input edge of the GTIOCnA input pin or GTIOCnB input pin, respectively (n = 0 to 3). The rising edge, falling edge, or both edges can be selected as the detection edge.

Figure 26.11 shows an example of the input capture function. In this example, the GPT0.GTCNT counter performs up-counting, and settings have been made so that an input capture is performed at both edges of the GTIOC0A input pin and at the rising edge of the GTIOC0B input pin.

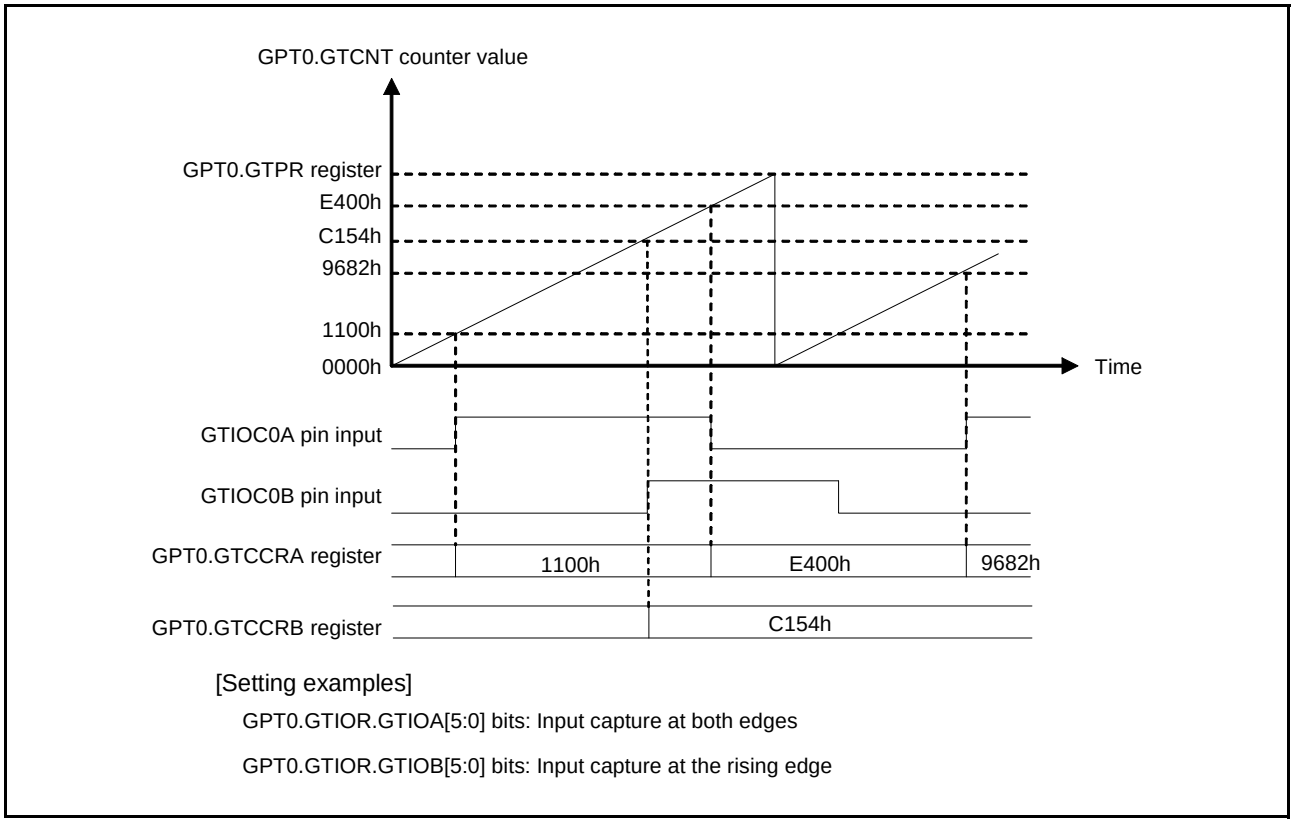


Figure 26.11 Example of Input Capture Operation

Figure 26.12 shows an example for setting input capture operation.

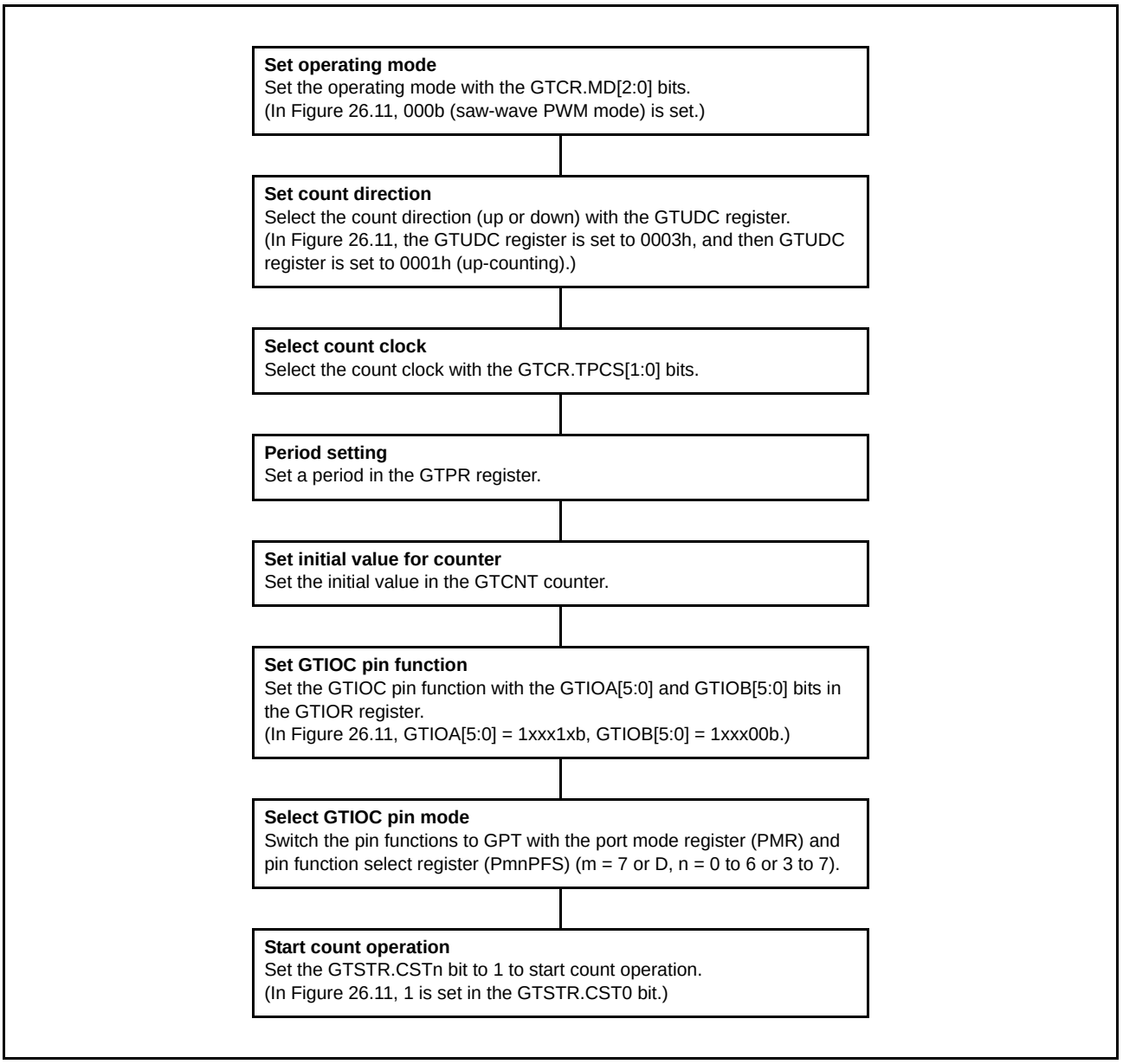


Figure 26.12 Example for Setting Input Capture Operation

26.3.2 Buffer Operation

The following buffer operation can be set with the GTBER register.

- Buffer operation with the GTPR, GTPBR, and GTPDBR registers used together
- Buffer operation with the GTCCRA, GTCCRC, and GTCCRD registers used together
- Buffer operation with the GTCCRB, GTCCRE, and GTCCRF registers used together
- Buffer operation with the GTADTRA, GTADTBRA, and GTADTDBRA registers used together
- Buffer operation with the GTADTRB, GTADTBRB, and GTADTDBRB registers used together

The following buffer operation can be set with the GTDTCR register.

- Buffer operation with the GTDVU and GTDBU registers used together
- Buffer operation with the GTDVD and GTDBD registers used together

26.3.2.1 GTPR Register Buffer Operation

The GTPBR register can function as a buffer register for the GTPR register, and the GTPDBR register can function as a buffer register for the GTPBR register (double buffer register for the GTPR register). When the count direction is down-counting in saw-wave mode, buffer operation setting is prohibited.
The buffer transfer is performed at an overflow or count clearing in saw-wave mode, and at a trough in triangle-wave mode.
To set the GTPR register to function as double buffer, set the GTBER.PR[1:0] bits to 10b or 11b. For single buffer operation, set 01b. Not to function as buffer, set 00b.
Figure 26.13 to Figure 26.14 show examples of GTPR buffer operation and Figure 26.15 shows an example for setting GTPR buffer operation.

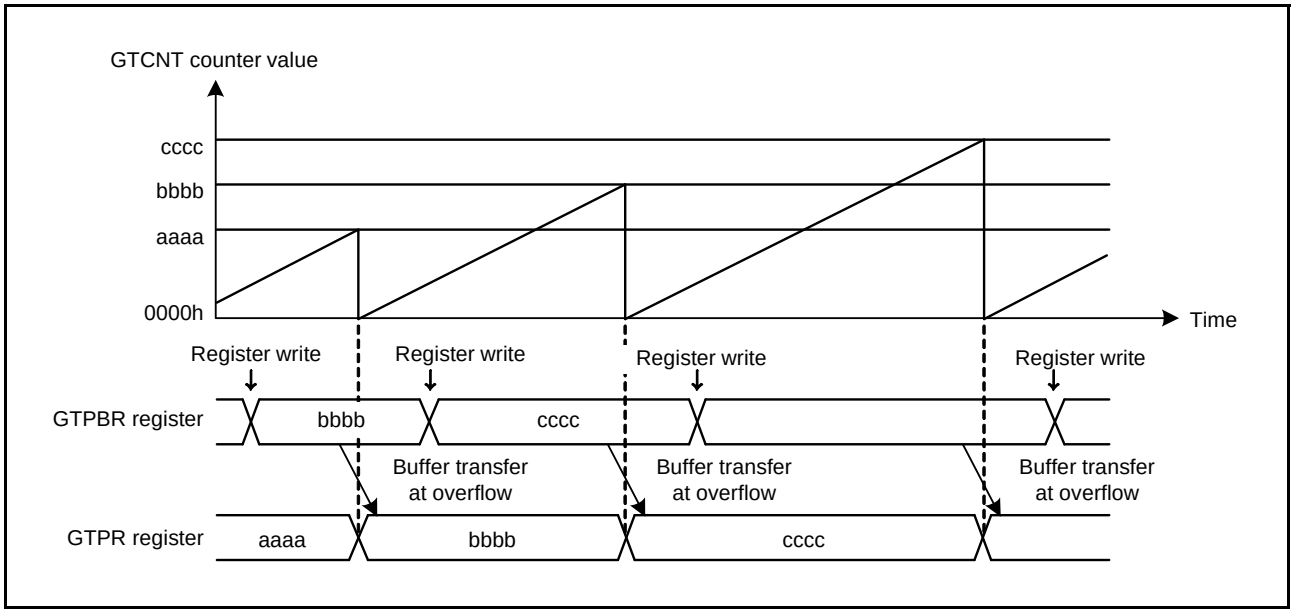


Figure 26.13 Example of GTPR Buffer Operation (Saw Waves in Up-Counting)

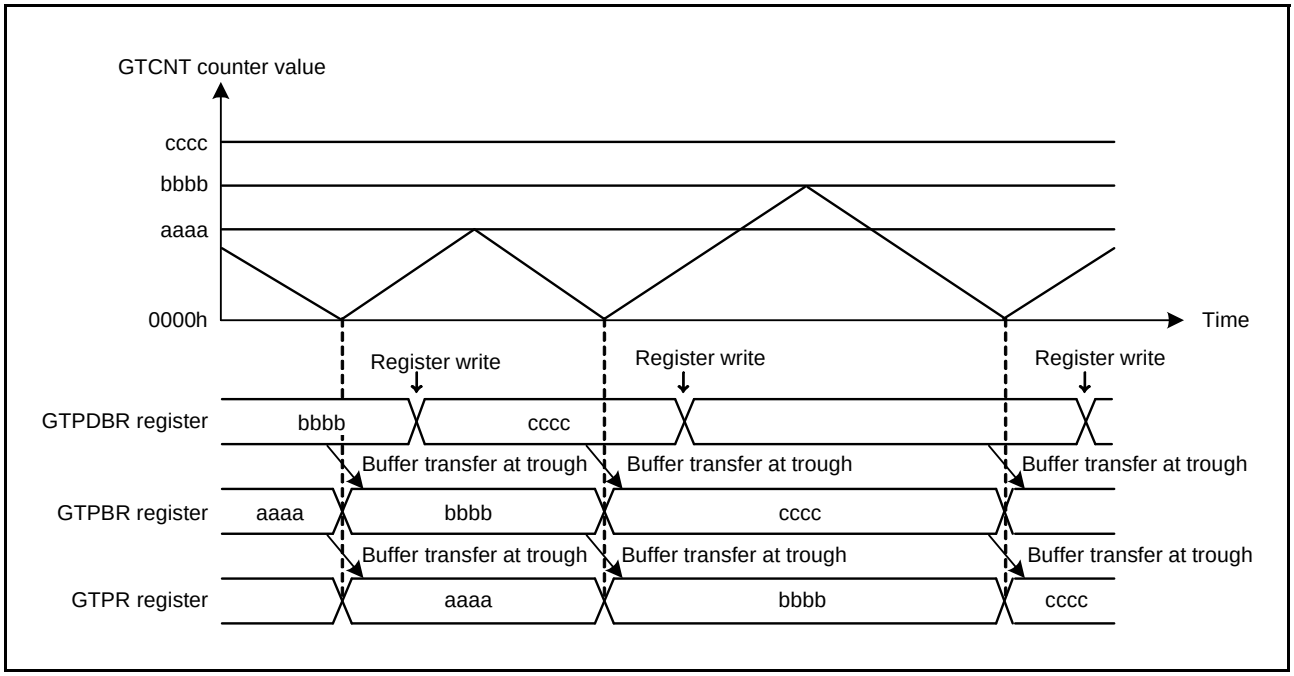


Figure 26.14 Example of GTPR Double Buffer Operation (Triangle Waves)

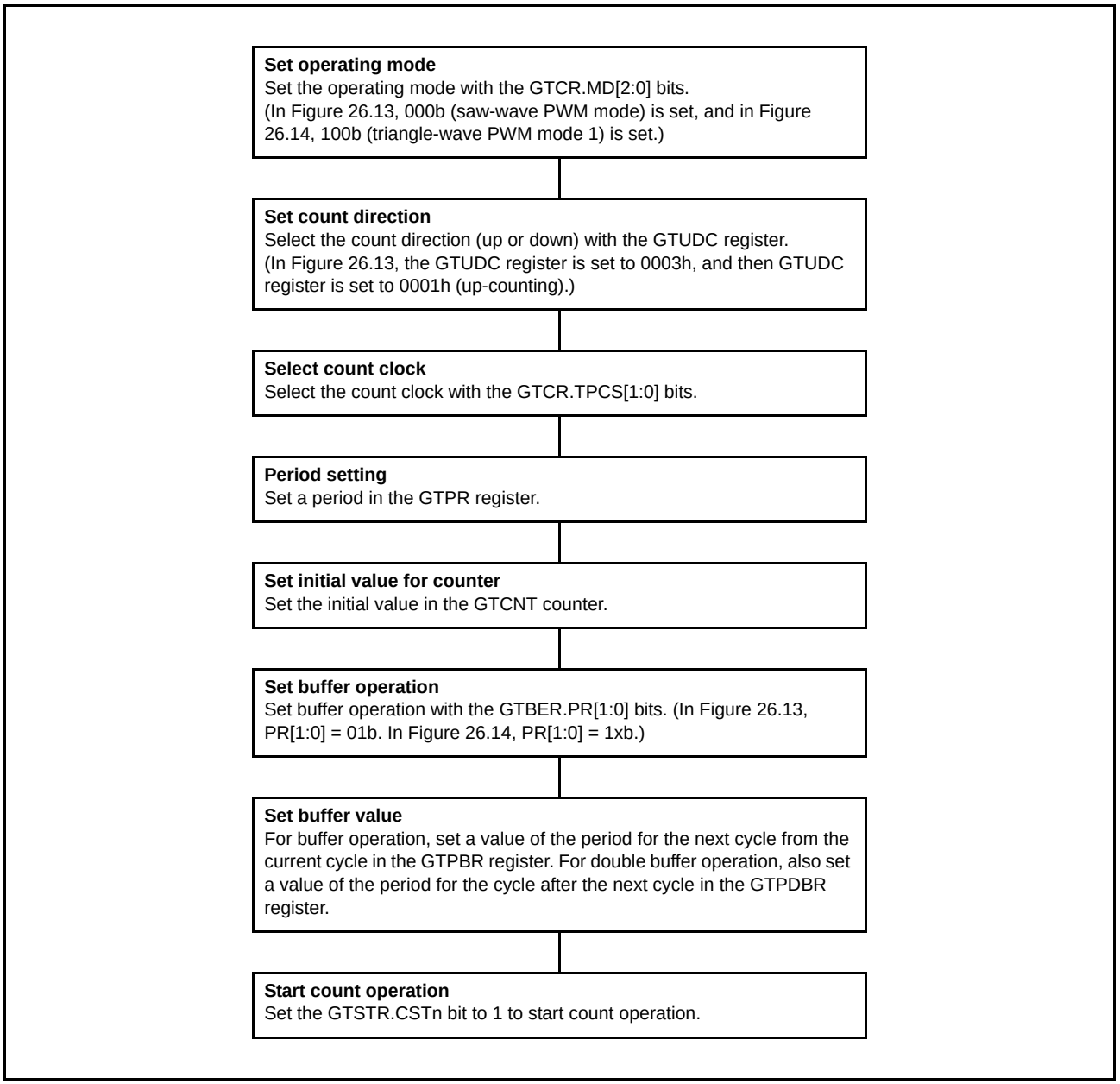


Figure 26.15 Example for Setting GTPR Buffer Operation

26.3.2.2 Buffer Operation for the GTCCRA and GTCCRB Registers

The GTCCRC register can function as the GTCCRA buffer register and the GTCCRD register can function as the GTCCRC buffer register (double buffer register for the GTCCRA register). Similarly, the GTCCRE register can function as the GTCCRB buffer register and the GTCCRF register can function as the GTCCRE buffer register (double buffer register for the GTCCRB register).
To set the GTCCRA or GTCCRB register to function as a double buffer, set the GTBER.CCRA[1:0] or GTBER.CCRB[1:0] bits to 10b or 11b. For single buffer operation, set 01b. Not to function as buffer, set 00b.
The following describes buffer operation during output compare and input capture operation.

(1) When the GTCCRA or GTCCRB Register Functions as Output Compare Register

Buffer transfer is performed at an overflow (in up-counting), underflow (in down-counting), count clearing in saw-wave mode, and at a crest or trough in triangle-wave mode. In both saw-wave mode and triangle-wave mode, buffer transfer of the GTCCRA and GTCCRB registers is forcibly performed by writing 1 to the GTBER.CCRSWT bit while counting is stopped.
In saw-wave one-shot pulse mode and triangle-wave PWM mode 3, buffer transfer from the GTCCRD register to temporary register A and the GTCCRF register to temporary register B is performed using the forcible buffer operation while counting is stopped.
Figure 26.16 to Figure 26.18 show examples of GTCCRA and GTCCRB buffer operation and Figure 26.19 shows an example for setting GTCCRA and GTCCRB buffer operation.

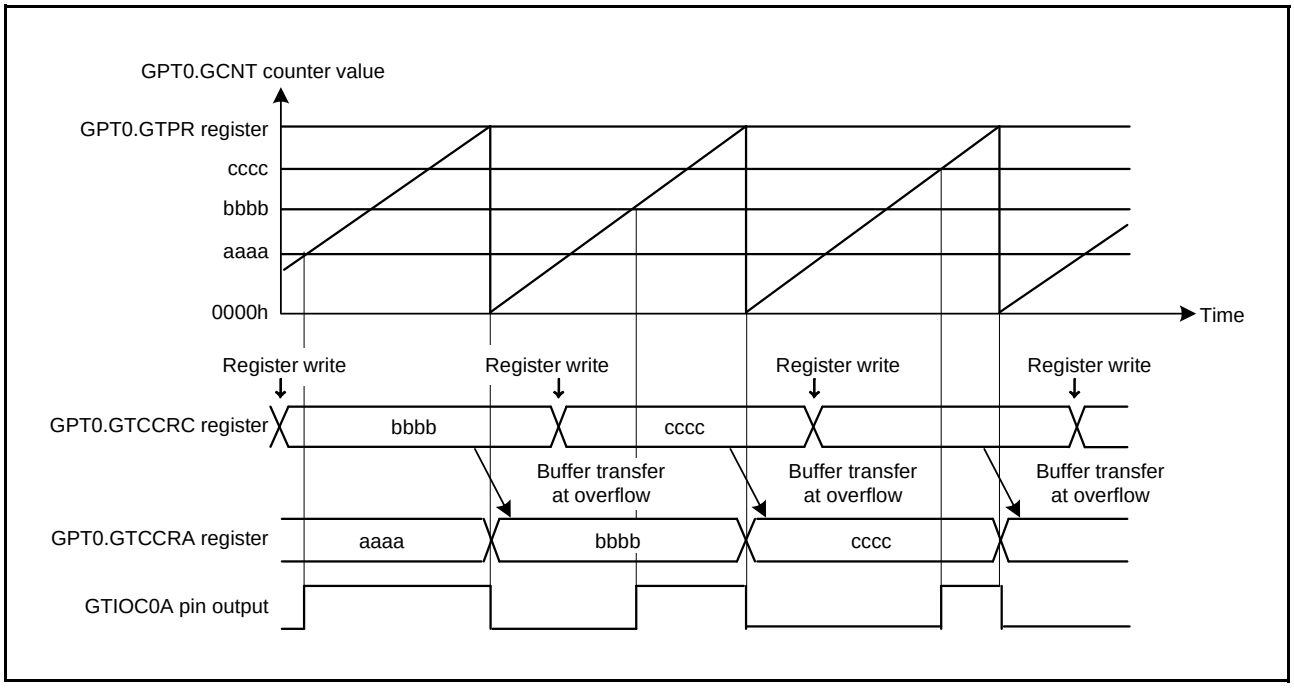


Figure 26.16 Example of GTCCRA and GTCCRB Buffer Operation (Output Compare, Saw Waves in Up-Counting, High Output at GTCCRA Compare Match, Low Output at the End of the Cycle)

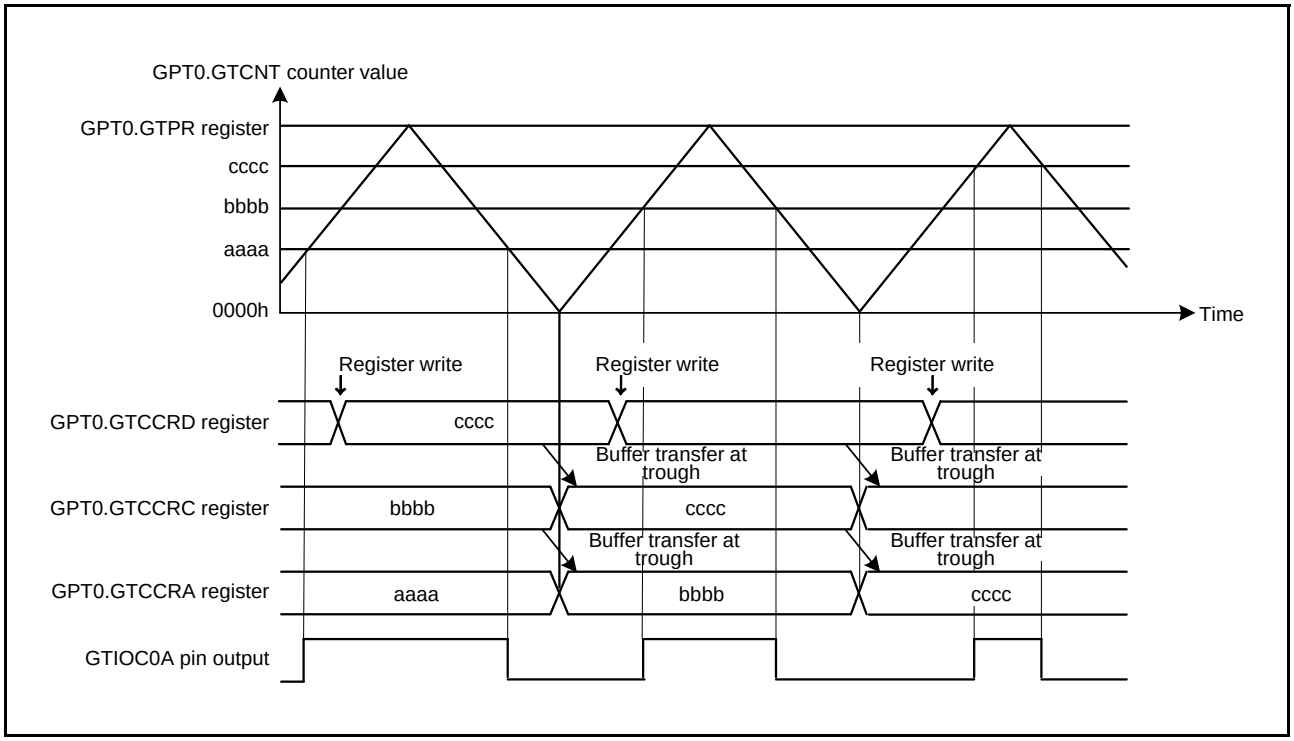


Figure 26.17 Example of GTCCRA and GTCCRB Double Buffer Operation (Output Compare, Triangle Waves, Buffer Operation at Trough, Output Toggled at GTCCRA Compare Match, Output Retained at the End of the Cycle)

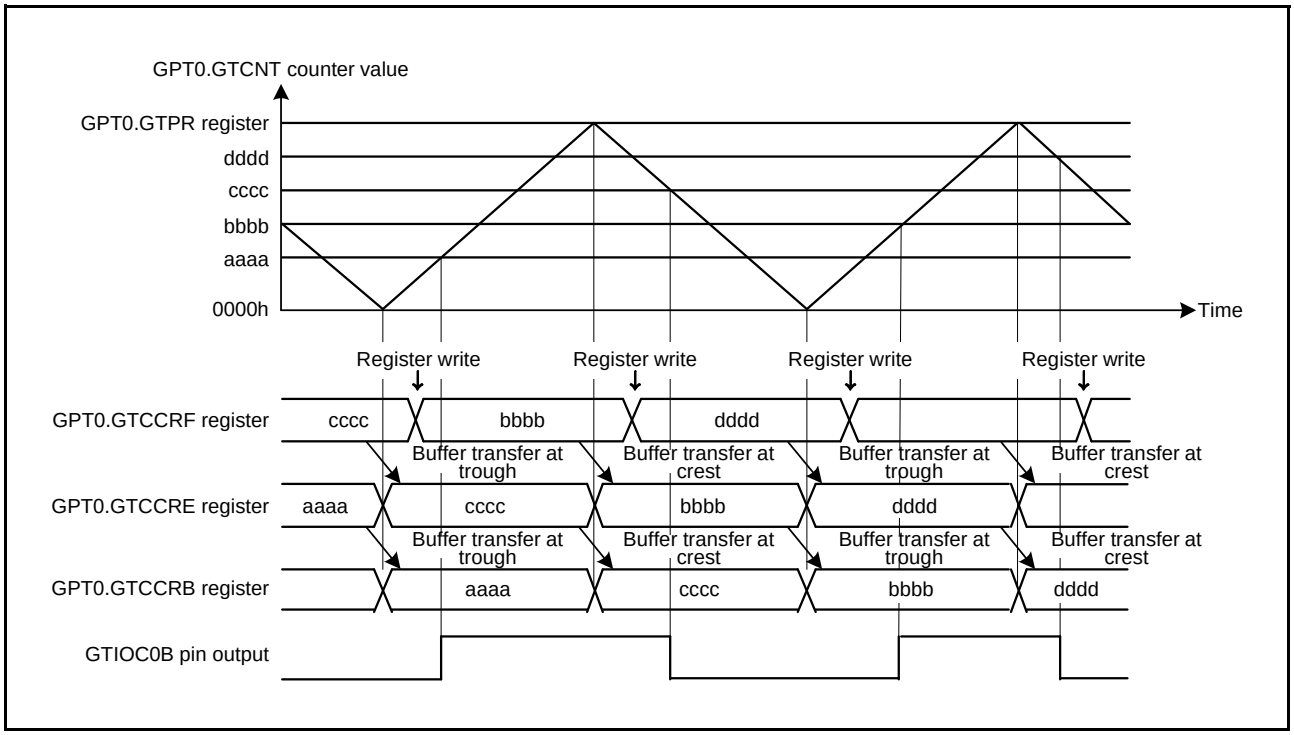


Figure 26.18 Example of GTCCRA and GTCCRB Double Buffer Operation (Output Compare, Triangle Waves, Buffer Operation at Both Troughs and Crests, Output Toggled at GTCCRB Compare Match, Output Retained at the End of the Cycle)

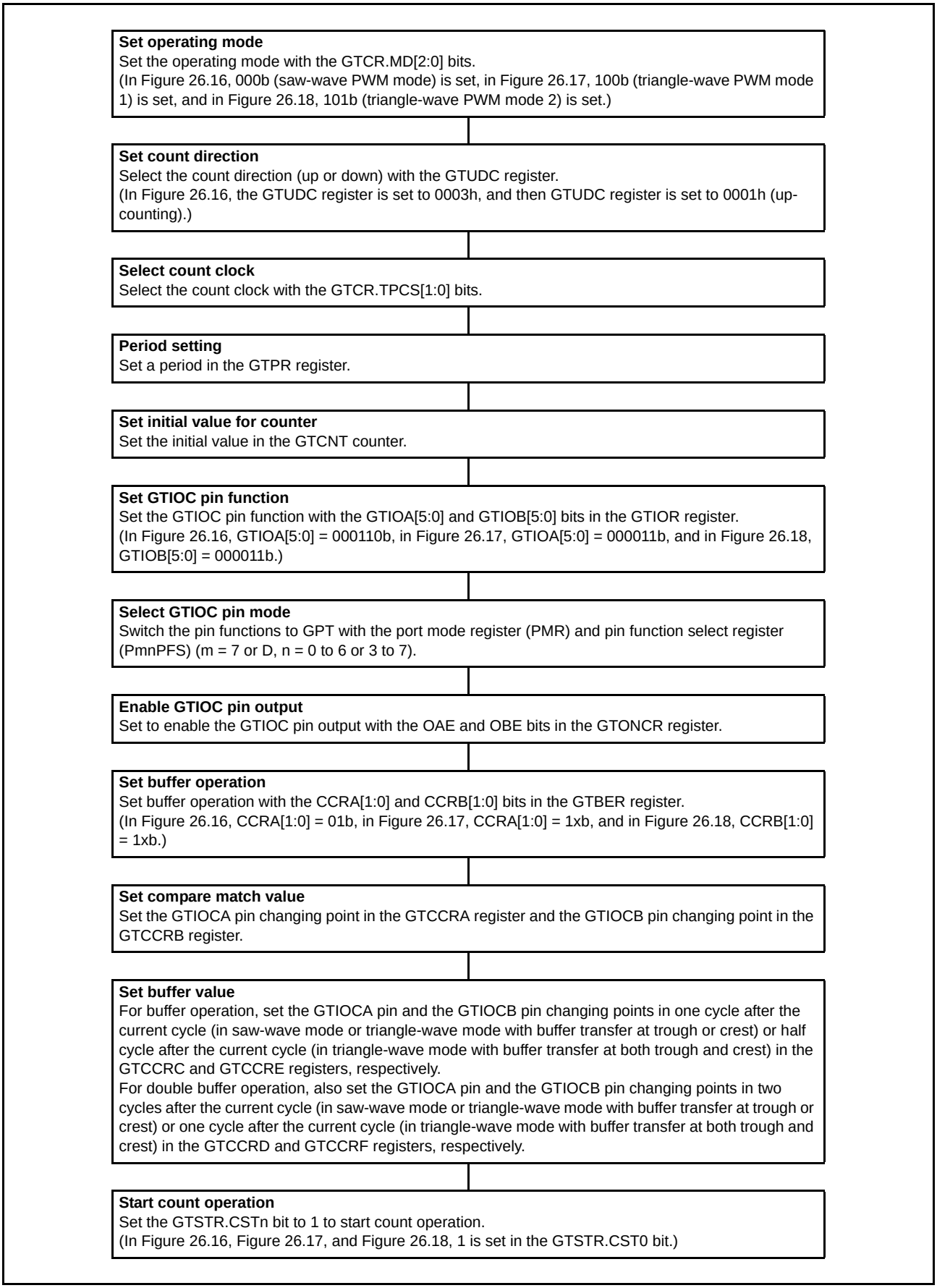


Figure 26.19 Example for Setting GTCCRA and GTCCRB Buffer Operation (for Output Compare)

(2) When the GTCCRA or GTCCRB Register Functions as Input Capture Register

When an input capture is generated, the GTCNT counter value is transferred to GTCCRA and GTCCRB and the stored GTCCRA and GTCCRB register values are transferred to buffer registers.
Figure 26.20 and Figure 26.21 show examples of GTCCRA and GTCCRB buffer operation and Figure 26.22 shows an example for setting GTCCRB buffer operation.

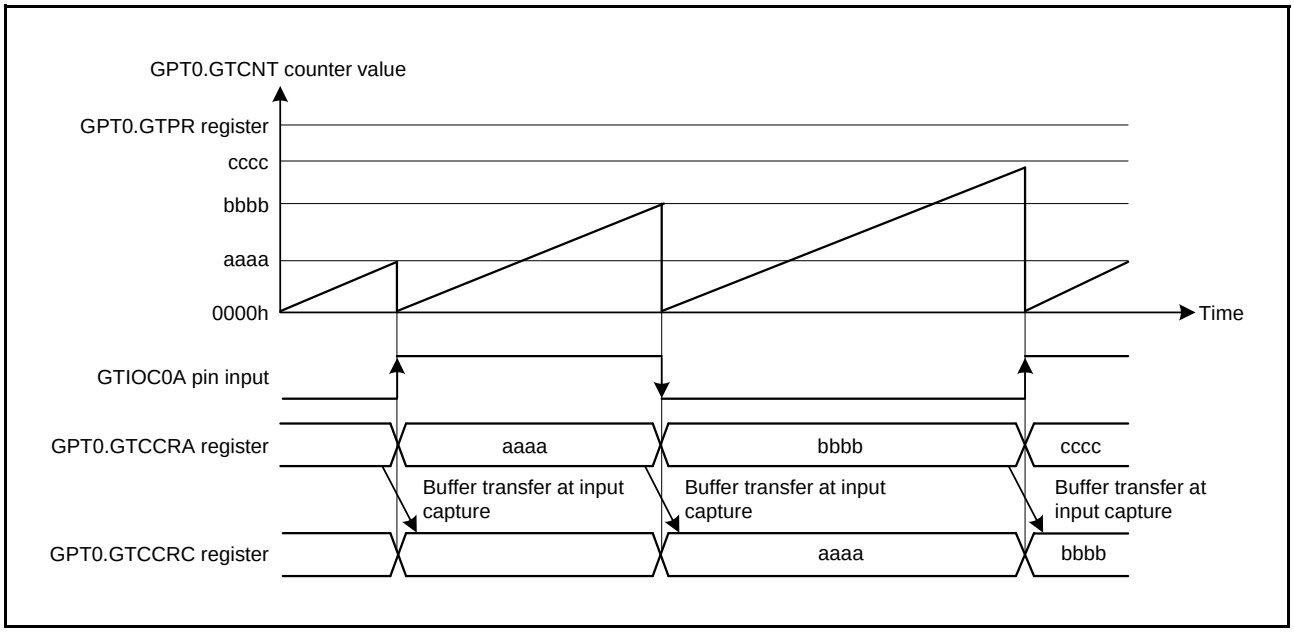


Figure 26.20 Example of GTCCRA and GTCCRB Buffer Operation (Input Capture at Both Edges of GTIOC0A Input, Saw Waves in Up-Counting, GTCNT Counter Cleared at GTCCRA Input Capture)

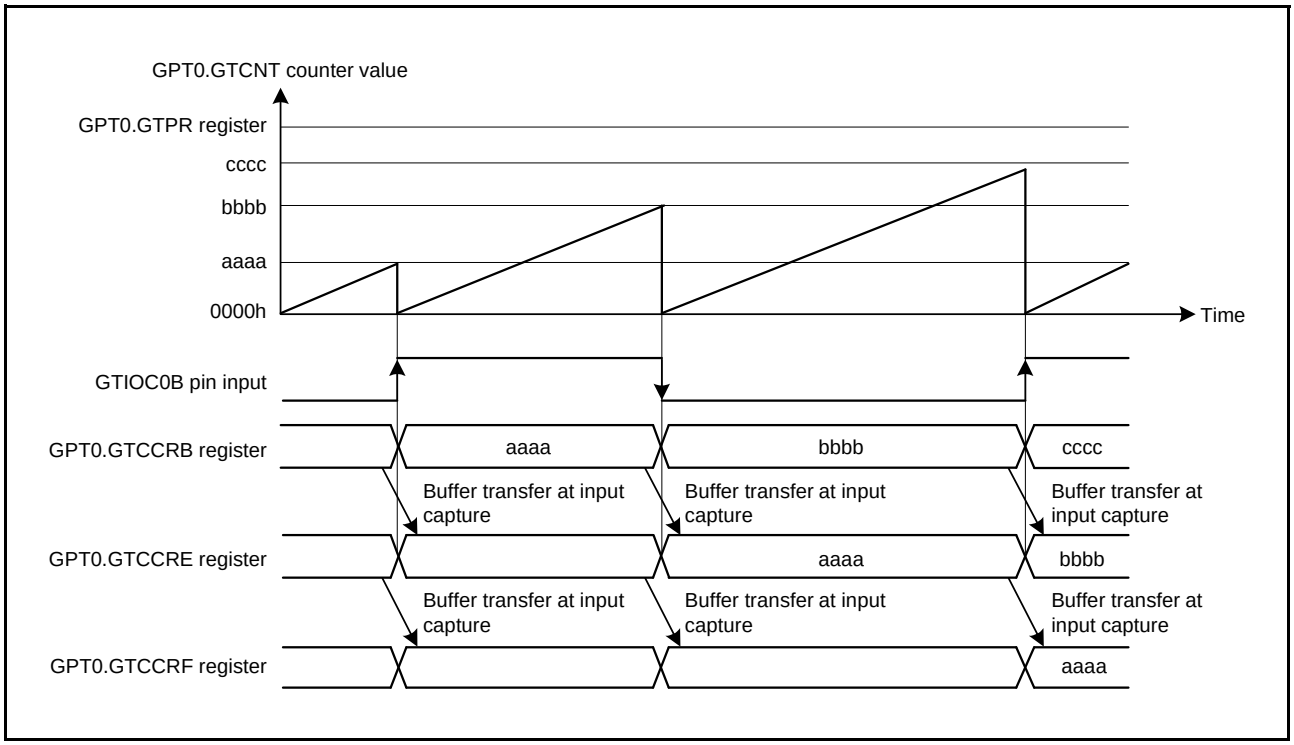


Figure 26.21 Example of GTCCRA and GTCCRB Double Buffer Operation (Input Capture at Both Edges of GTIOC0B Input, Saw Waves in Up-Counting, GTCNT Counter Cleared at GTCCRB Input Capture)

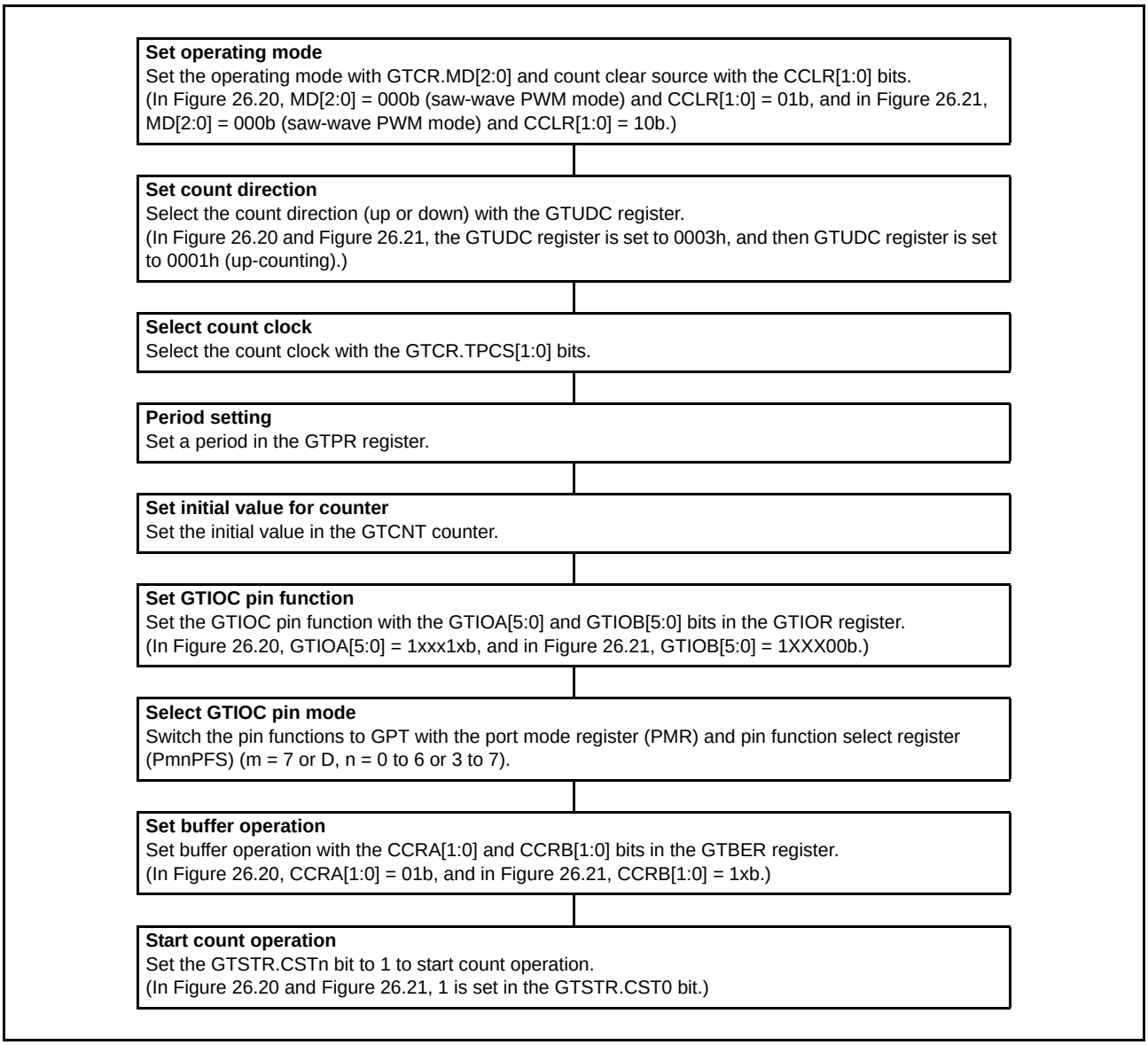


Figure 26.22 Example for Setting GTCCRA and GTCCRB Buffer Operation (for Input Capture)

26.3.2.3 Buffer Operation for the GTADTRA and GTADTRB Registers

The GTADTBRA register can function as the GTADTRA buffer register and the GTADTDBRA register can function as the GTADTBRA buffer register (double buffer register for the GTADTRA register). Similarly, the GTADTBRB register can function as the GTADTRB buffer register and the GTADTDBRB register can function as the GTADTBRB buffer register (double buffer register for the GTADTRB register).

To set the GTADTRA or GTADTRB register to function as a double buffer, set the GTBER.ADTDA or GTBER.ADTDB bit to 1. For single buffer operation, set 0. Not to function as buffer, set the GTBER.ADTTA[1:0] or GTBER.ADTTB[1:0] bits to 00b.

The buffer transfer timing can be set with the GTBER.ADTTn[1:0] bits. For saw waves, overflows (during up-counting), underflows (during down-counting), or count clearing can be selected. For triangle waves, crests are selected when GTBER.ADTTn[1:0] = 01b, troughs are selected when GTBER.ADTTn[1:0] = 10b, and both crests and troughs are selected when GTBER.ADTTn[1:0] = 11b (n =A, B).

Figure 26.23 to Figure 26.25 show examples of GTADTRA and GTADTRB buffer operation and Figure 26.26 shows an example for setting GTDTRA and GTADTRB buffer operation.

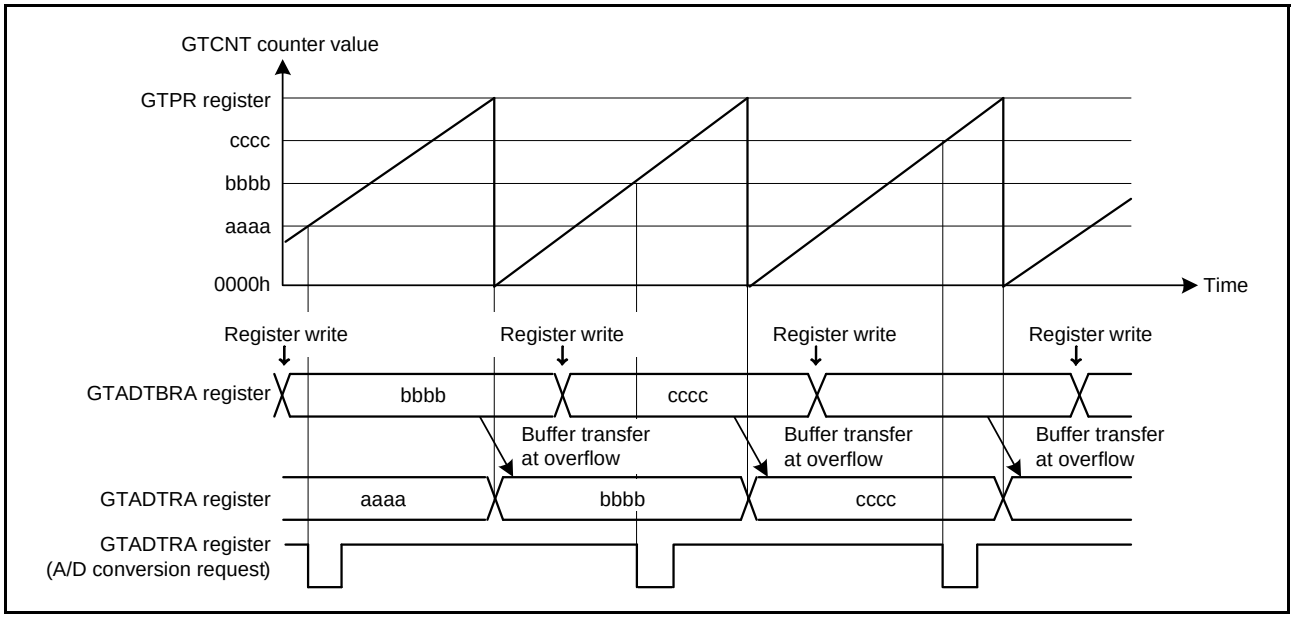


Figure 26.23 Example of GTADTRA and GTADTRB Buffer Operation (Saw Waves in Up-Counting, A/D Converter Start Request Generated by Up-Counting)

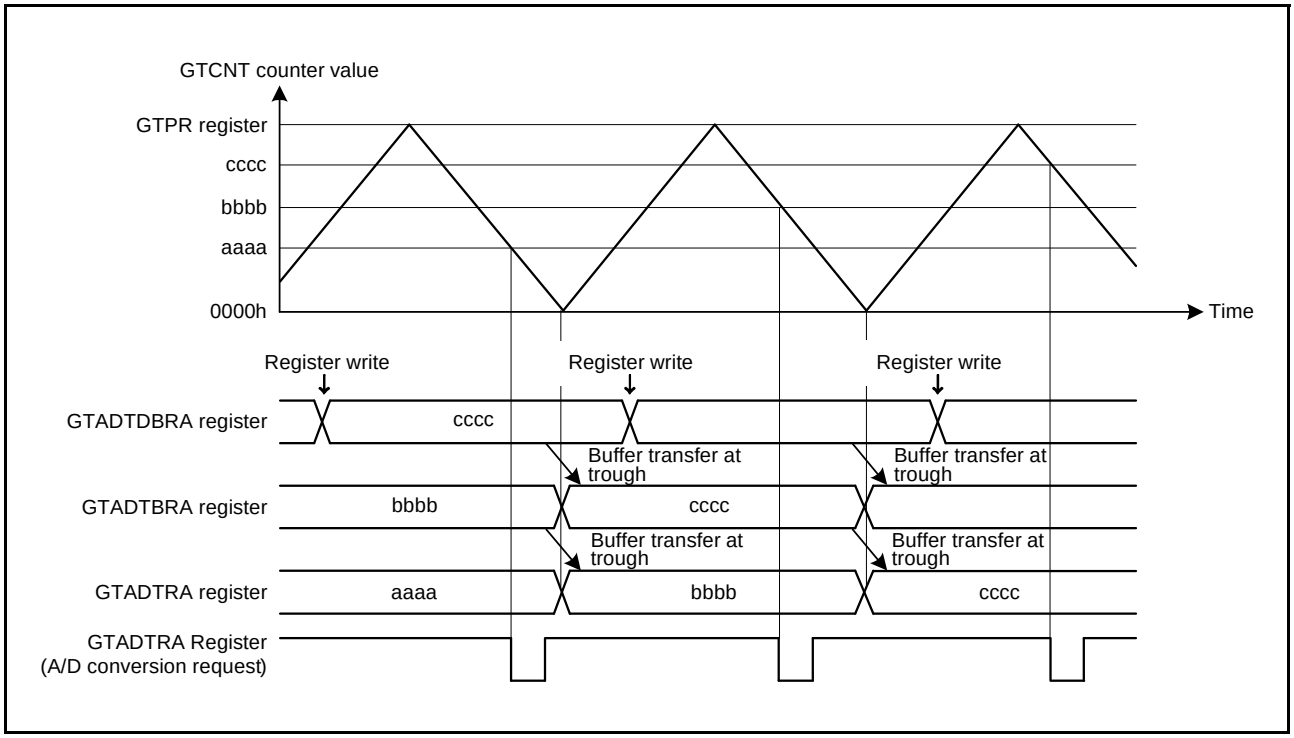


Figure 26.24 Example of GTADTRA and GTADTRB Double Buffer Operation (Triangle Waves, Buffer Transfer at Troughs, A/D Converter Start Request Generated by Down-Counting)

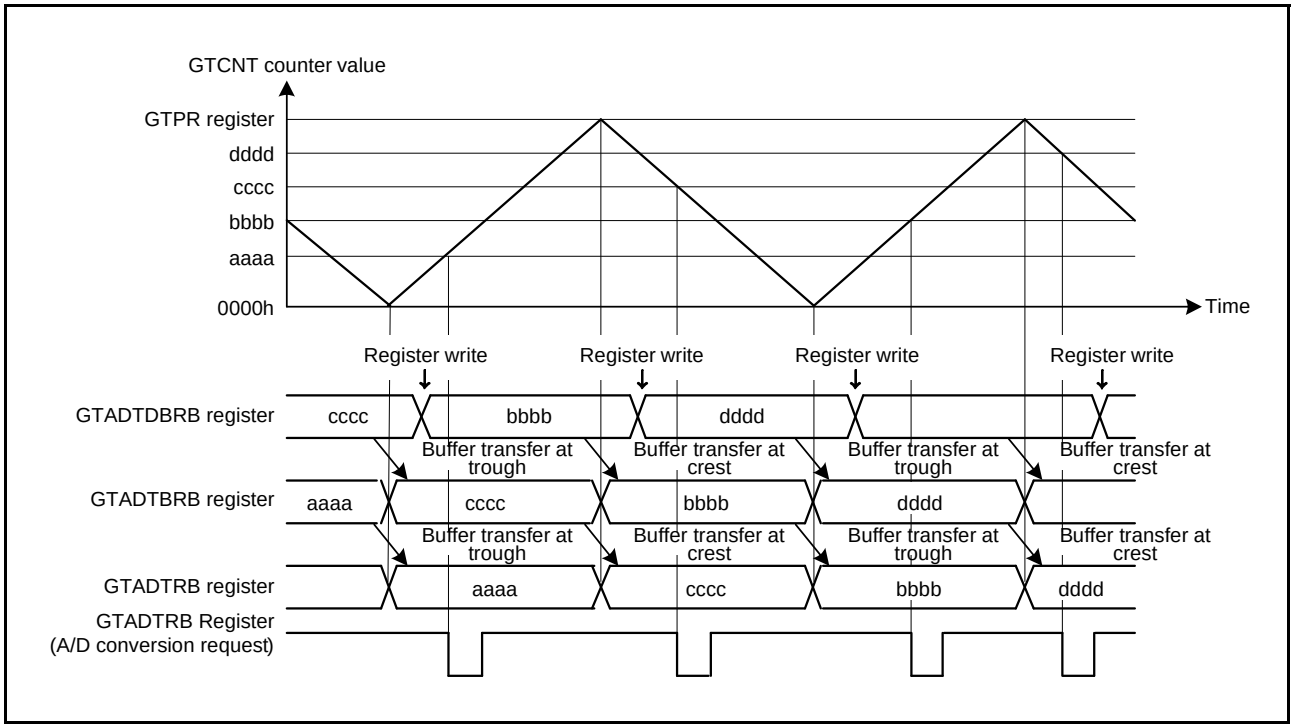


Figure 26.25 Example of GTADTRA and GTADTRB Double Buffer Operation (Triangle Waves, Buffer Transfer at Both Troughs and Crests, A/D Converter Start Request Generated by Both Up- and Down-Counting)

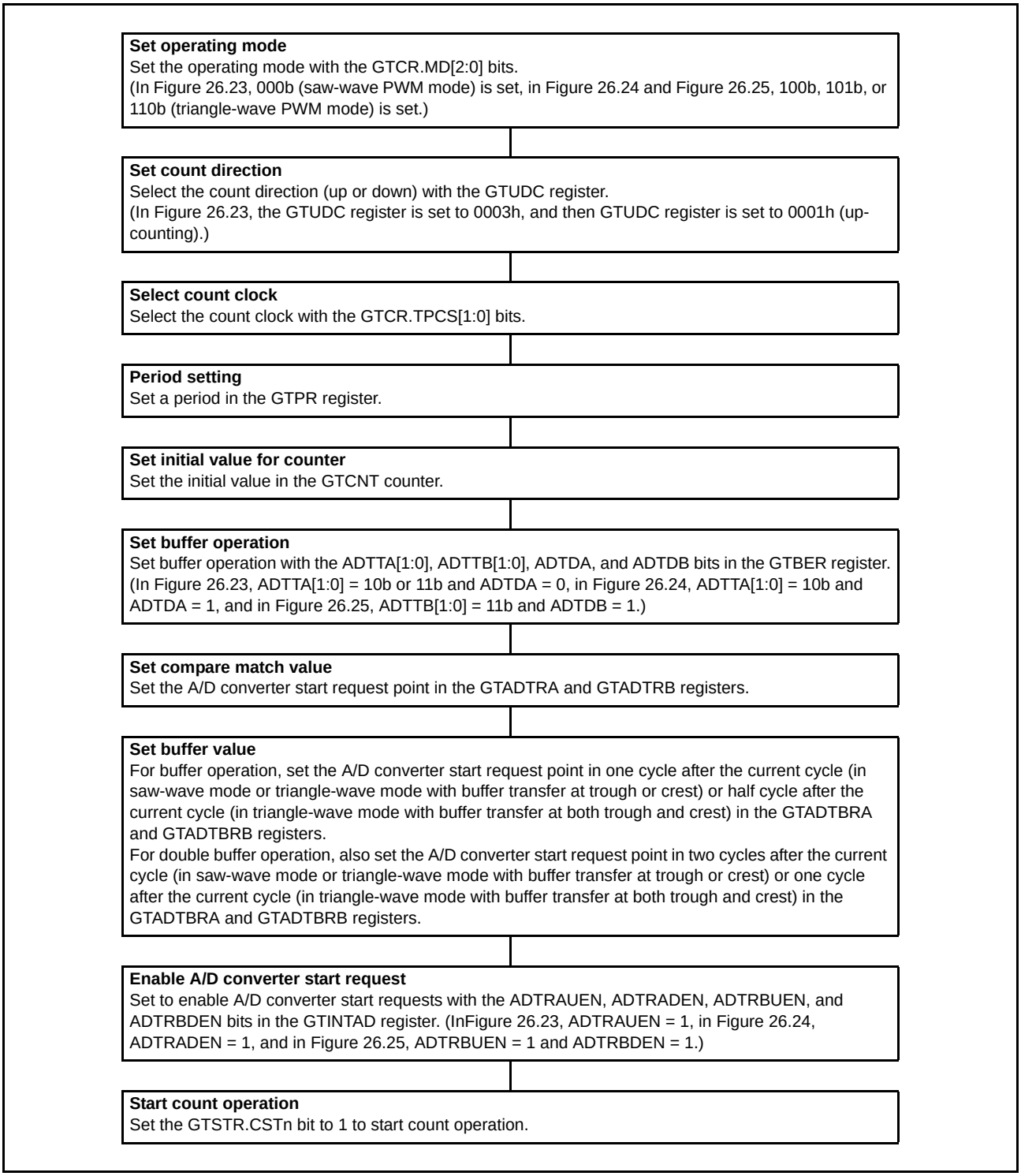


Figure 26.26 Example for Setting GTADTRA and GTADTRB Buffer Operation

26.3.3 PWM Output Operating Mode

The GPT can output PWM waveforms to the GTIOCnA pin or GTIOCnB pin by a compare match between the GPTn.GTCNT counter and the GPTn.GTCCRA or GPTn.GTCCRB register (n = 0 to 3). An operating mode can be set independently for each channel, and synchronous operation on channels is also possible.
By setting the GTDTCR, GTDVU, and GTDVD registers, the compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

(1) Saw-Wave PWM Mode (GTCR.MD = 000b)

In saw-wave PWM mode, the GPTn.GTCNT counter performs saw-wave (half-wave) operation by setting the period in the GTPR register and a PWM waveform is output to the GTIOCnA or GTIOCnB pin when a GPTn.GTCCRA or GPTn.GTCCRB compare match occurs (n = 0 to 3). The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR setting
Figure 26.27 shows an example of saw-wave PWM mode operation, and Figure 26.28 shows an example for setting saw-wave PWM mode.

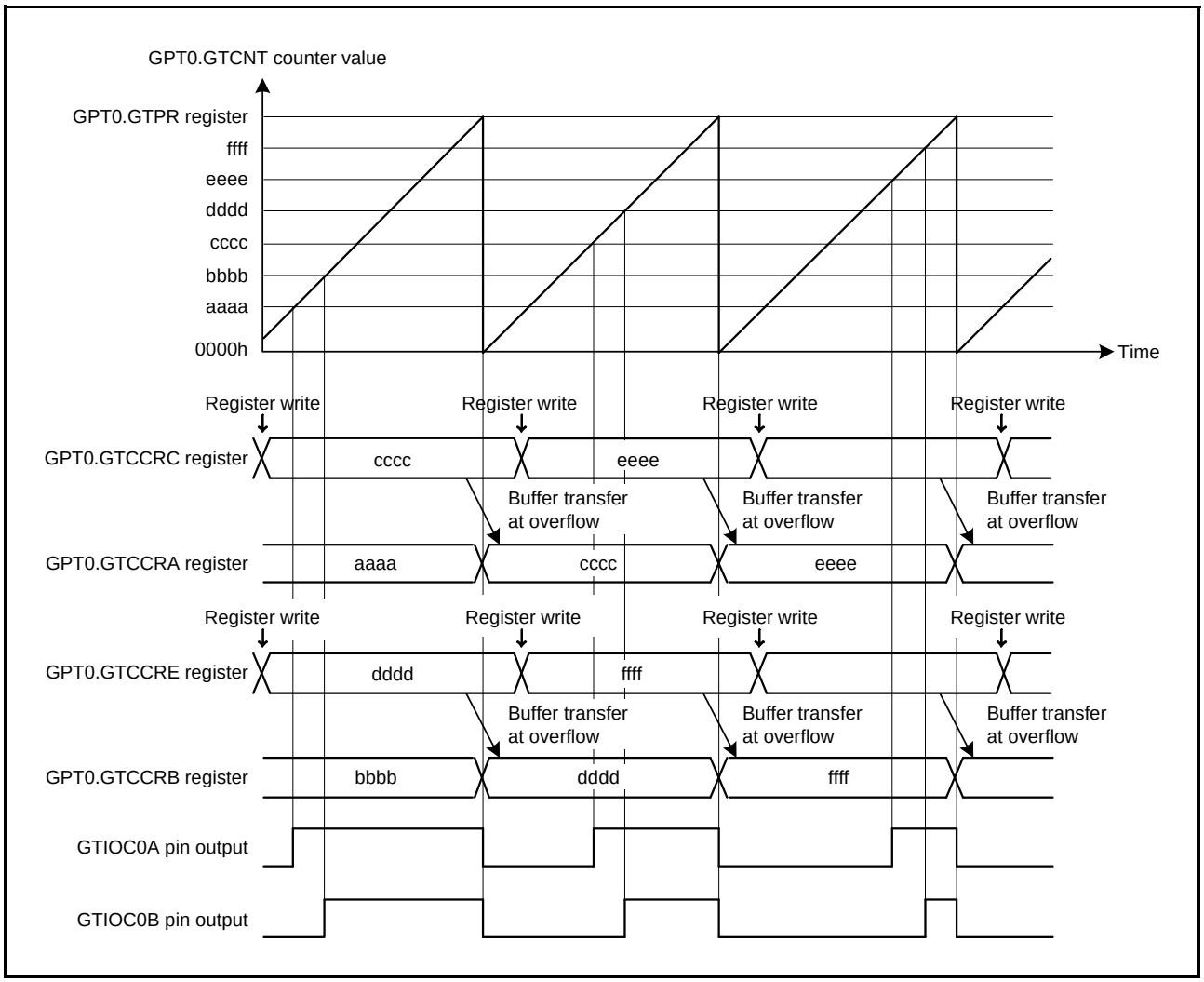


Figure 26.27 Example of Saw-Wave PWM Mode Operation (Up-Counting, Buffer Operation, High Output at GTCCRA/GTCCRB Compare Match, Low Output at the End of the Cycle)

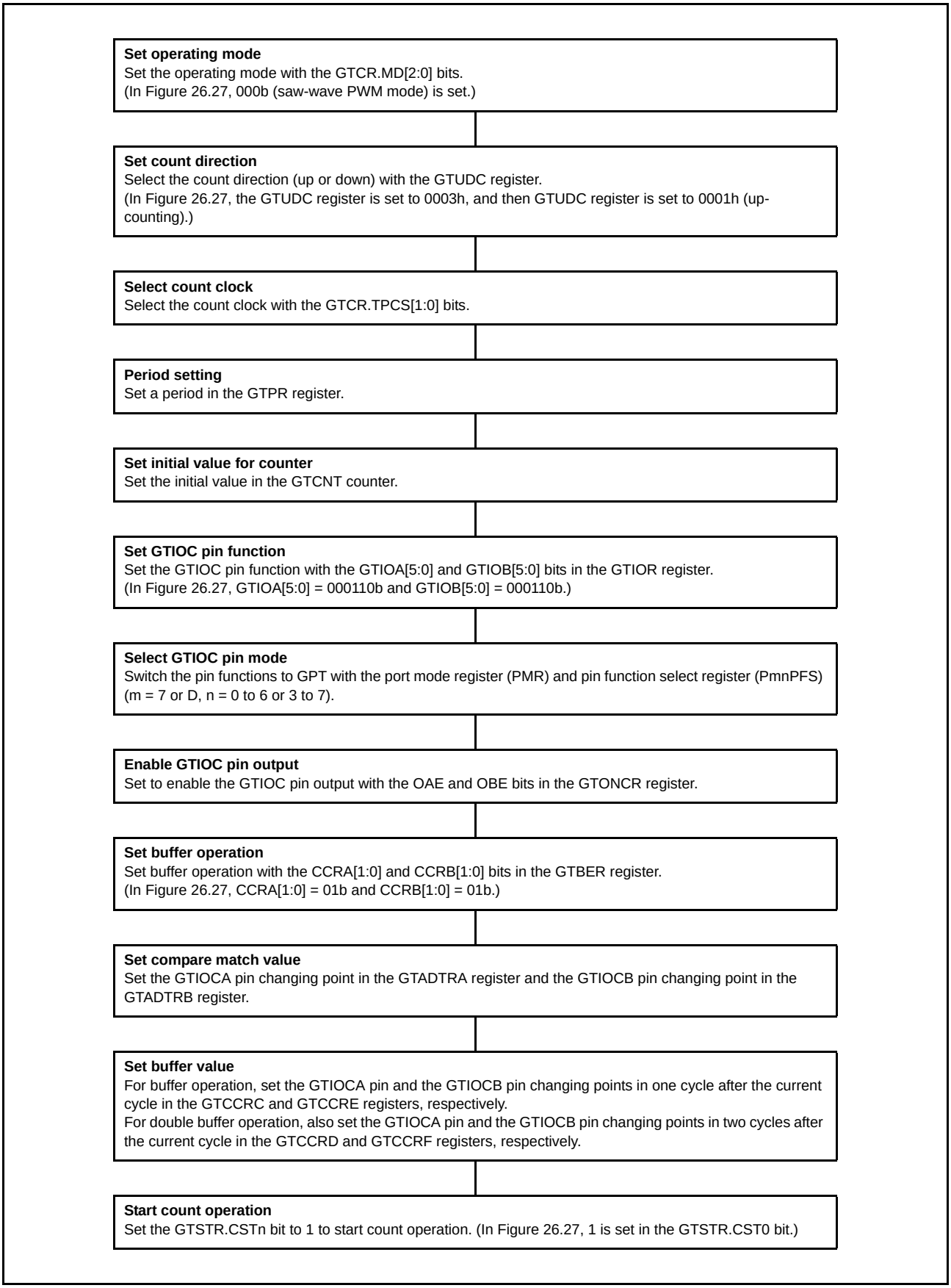


Figure 26.28 Example for Setting Saw-Wave PWM Mode

(2) Saw-Wave One-Shot Pulse Mode

The saw-wave one-shot pulse mode is a mode in which the period is set in the GPTn.GTPR register, the GPTn.GTCNT counter performs saw-wave (half-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin at a compare match of the GPTn.GTCCRA or GPTn.GTCCRB register with buffer operation fixed (n = 0 to 3).

Buffer operation in saw-wave one-shot pulse mode is different from the usual buffer operation. Buffer transfer is performed from GTCCRC to GTCCRA, from GTCCRE to GTCCRB, from GTCCRD to temporary register A, and from GTCCRF to temporary register B at the end of the cycle, and from temporary register A to GTCCRA at a GTCCRA compare match and from temporary register B to GTCCRB at a GTCCRB compare match. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and the end of the cycle according to the GTIOR setting.

By setting GTDTCR, GTDVU, and GTDVD, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 26.29 shows an example of saw-wave one-shot pulse mode operation, and Figure 26.30 shows an example for setting saw-wave one-shot pulse mode.

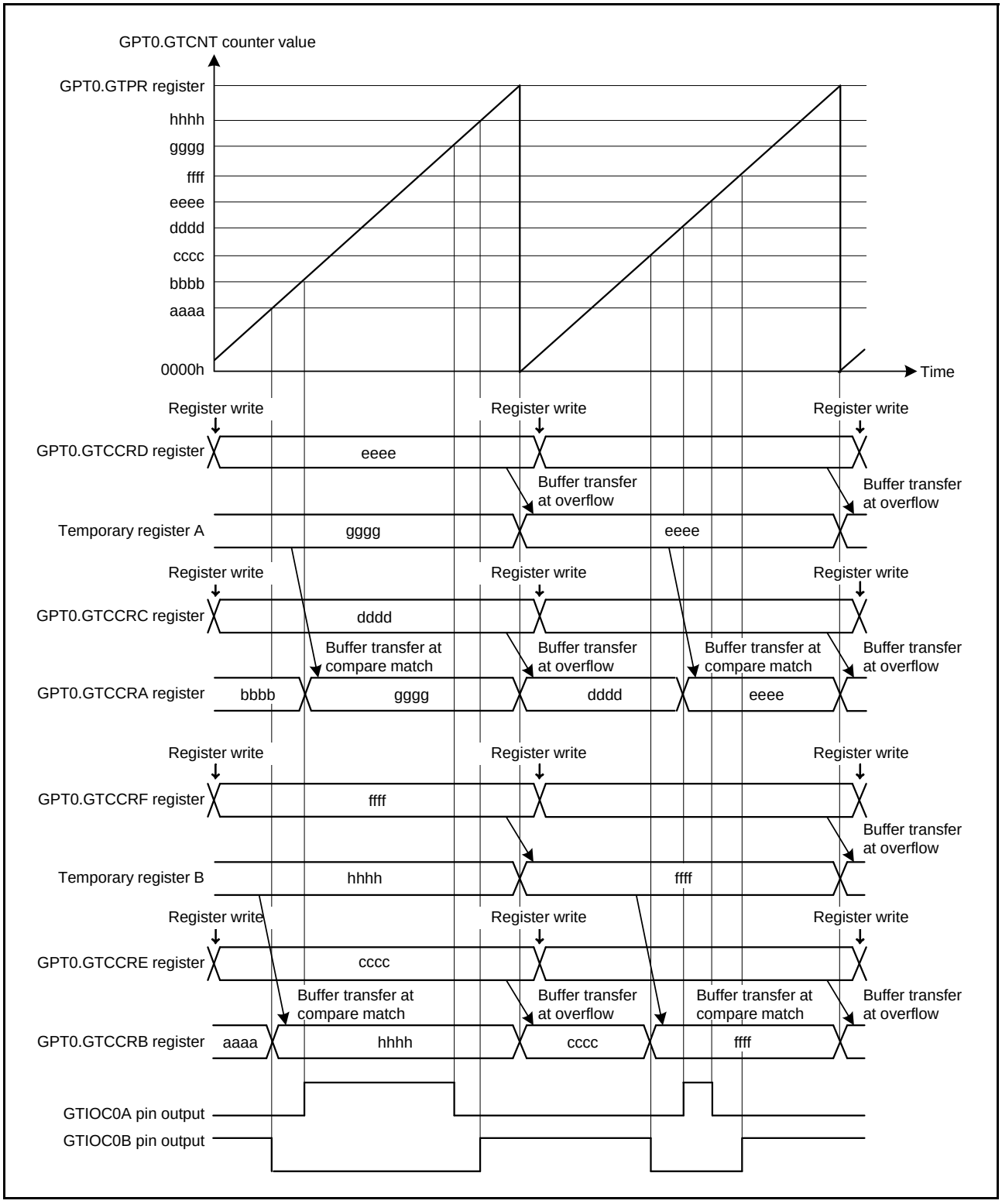


Figure 26.29 Example of Saw-Wave One-Shot Pulse Mode Operation (Up-Counting, Low Output from the GTIOC0A Pin and High Output from the GTIOC0B Pin at Initial Output, Output Toggled at GTCCRA/GTCCRB Compare Match, Output Retained at the End of the Cycle)

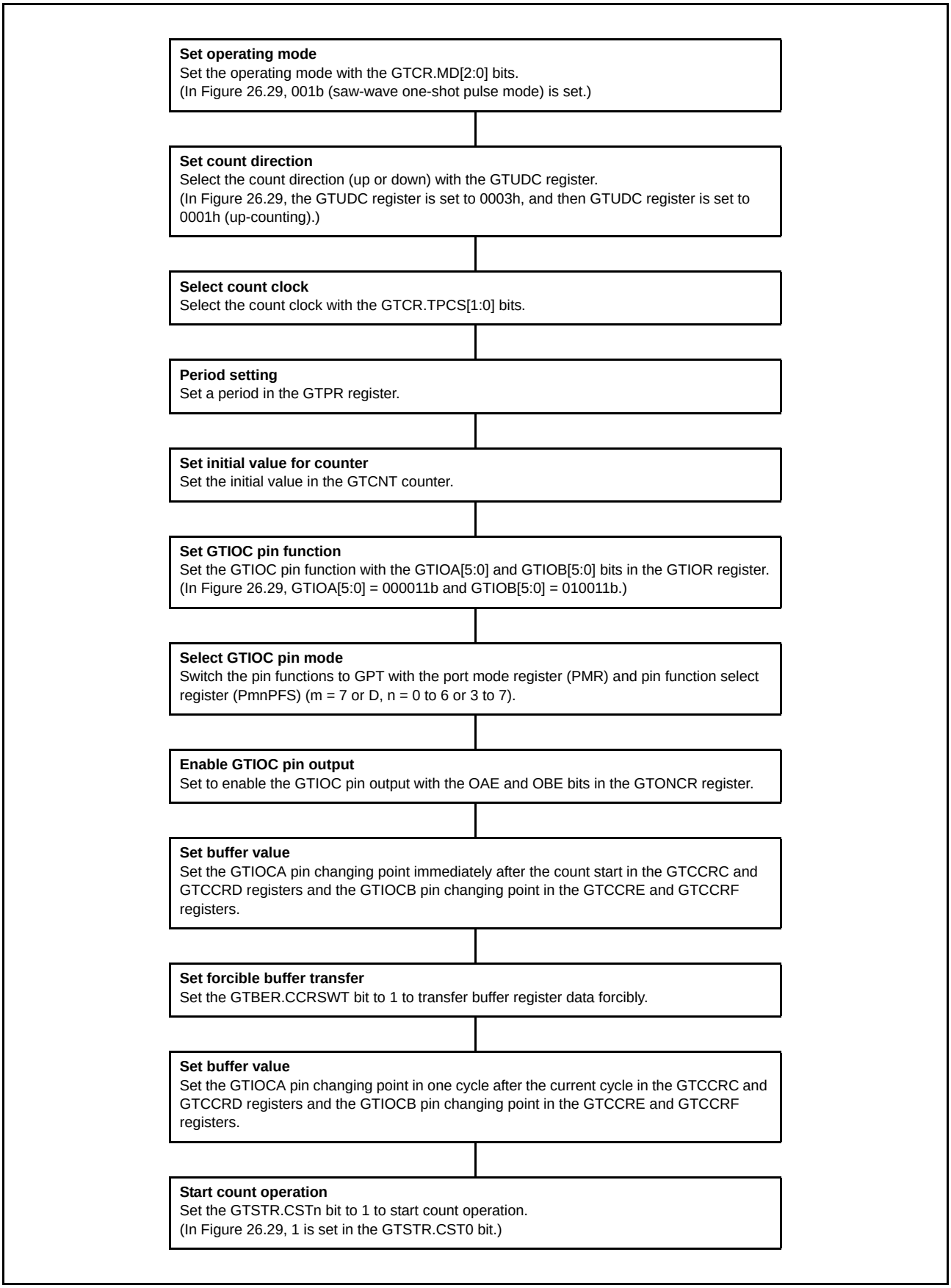


Figure 26.30 Example for Setting Saw-Wave One-Shot Pulse Mode

(3) Triangle-Wave PWM Mode 1 (16-Bit Transfer at Trough)

The triangle-wave PWM mode 1 is a mode in which the period is set in the GPTn.GTPR register, the GPTn.GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin when a GPTn.GTCCRA or GPTn.GTCCRB compare match occurs (n = 0 to 3). Buffer transfer is performed at the trough. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR setting.

By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

Figure 26.31 shows an example of triangle-wave PWM mode 1 operation, and Figure 26.32 shows an example for setting triangle-wave PWM mode 1.

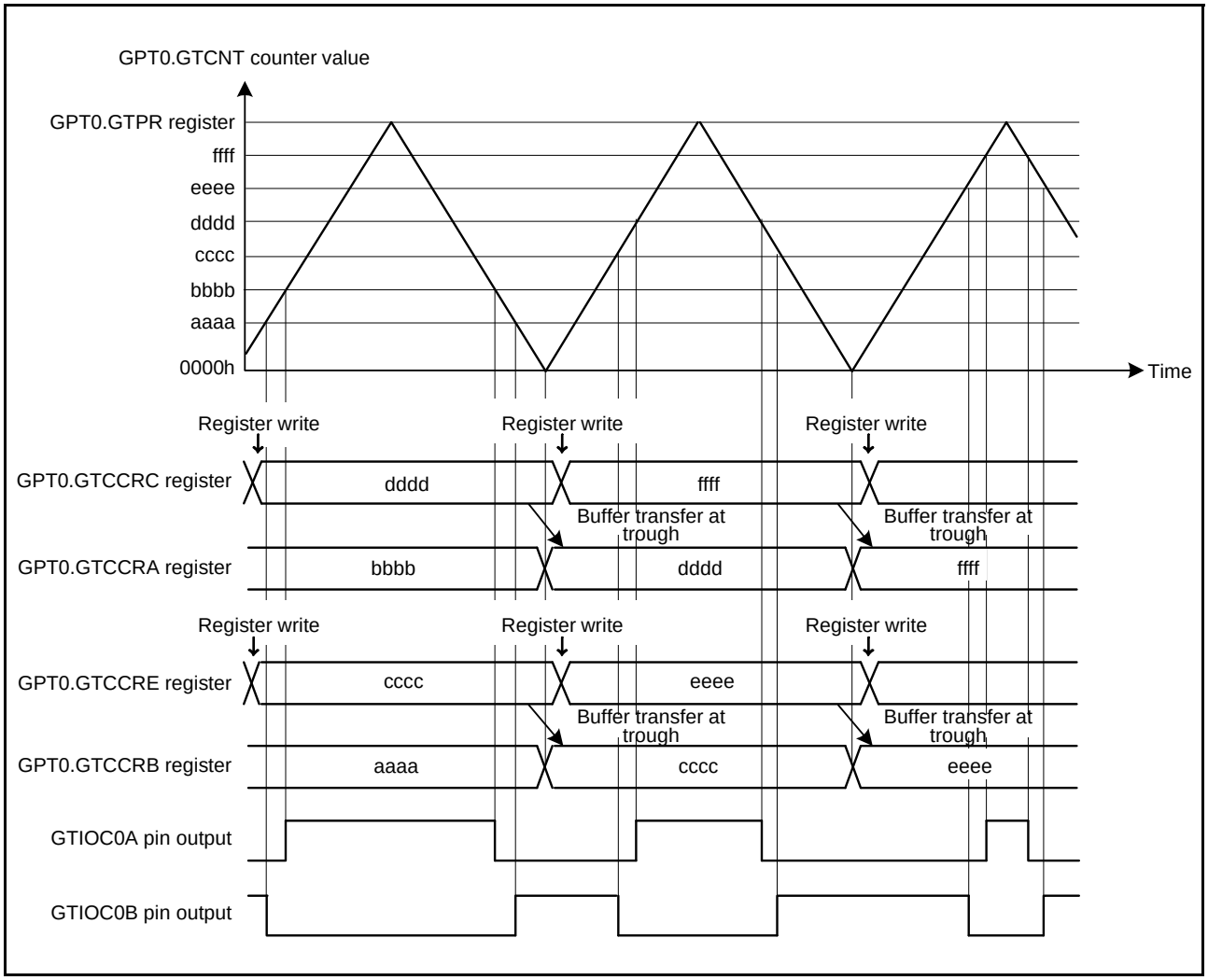


Figure 26.31 Example of Triangle-Wave PWM Mode 1 Operation (Buffer Operation, Low Output from the GTIOC0A Pin and High Output from the GTIOC0B Pin at Initial Output, Output Toggled at GTCCRA/GTCCRB Register Compare Match, Output Retained at the End of the Cycle)

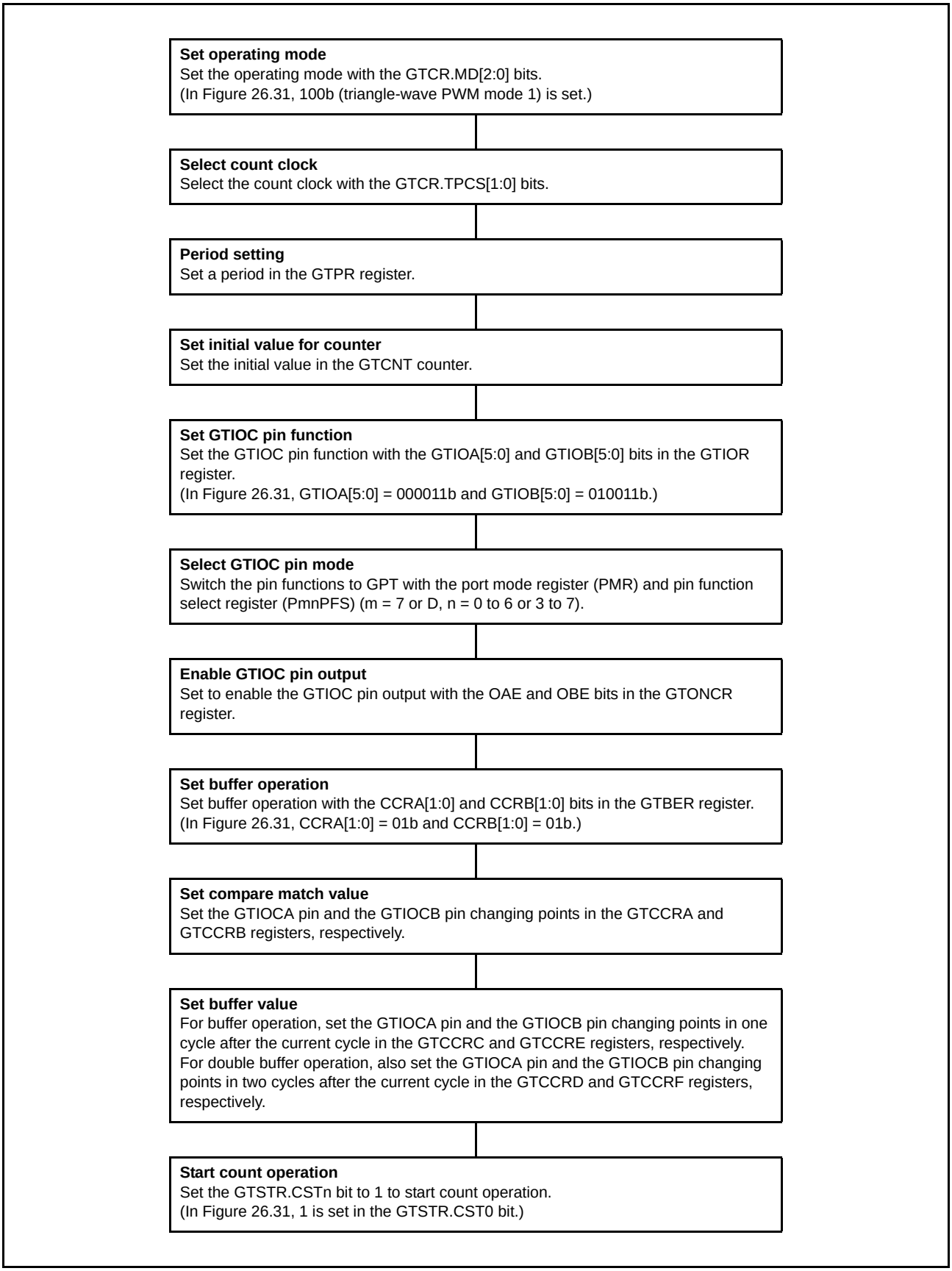


Figure 26.32 Example for Setting Triangle-Wave PWM Mode 1

(4) Triangle-Wave PWM Mode 2 (16-Bit Transfer at Crest and Trough)

Similarly to triangle-wave PWM mode 1, in triangle-wave PWM mode 2 the period is set in the GPTn.GTPR register, the GPTn.GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin when a GPTn.GTCCRA or GPTn.GTCCRB compare match occurs (n = 0 to 3). The buffer transfer is performed at both crests and troughs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR setting.
By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.
Figure 26.33 shows an example of triangle-wave PWM mode 2 operation, and Figure 26.34 shows an example for setting triangle-wave PWM mode 2.

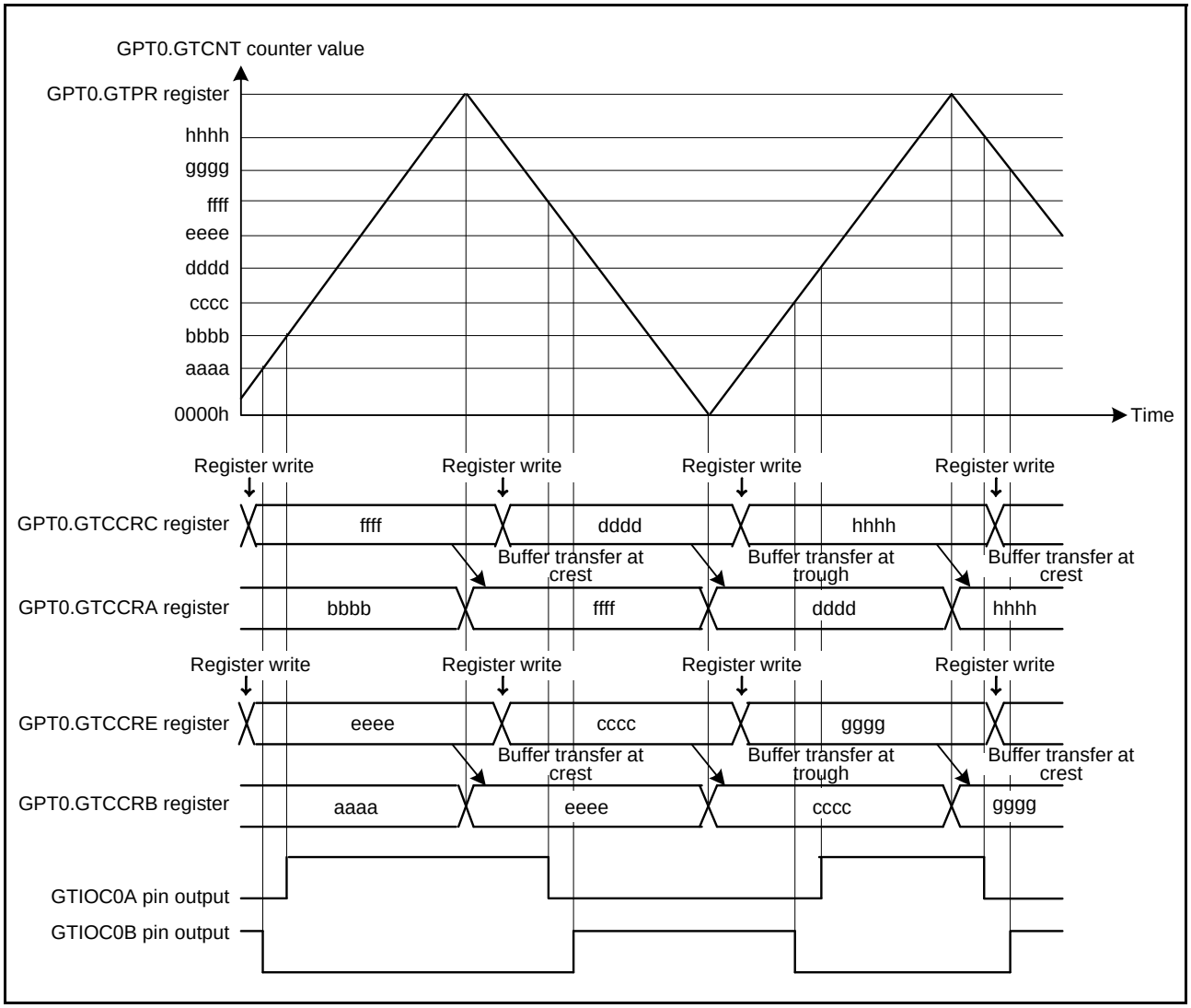


Figure 26.33 Example of Triangle-Wave PWM Mode 2 Operation (Buffer Operation, Low Output from the GTIOC0A Pin and High Output from the GTIOC0B Pin at Initial Output, Output Toggled at GTCCRA/GTCCRB Compare Match, Output Retained at the End of the Cycle)

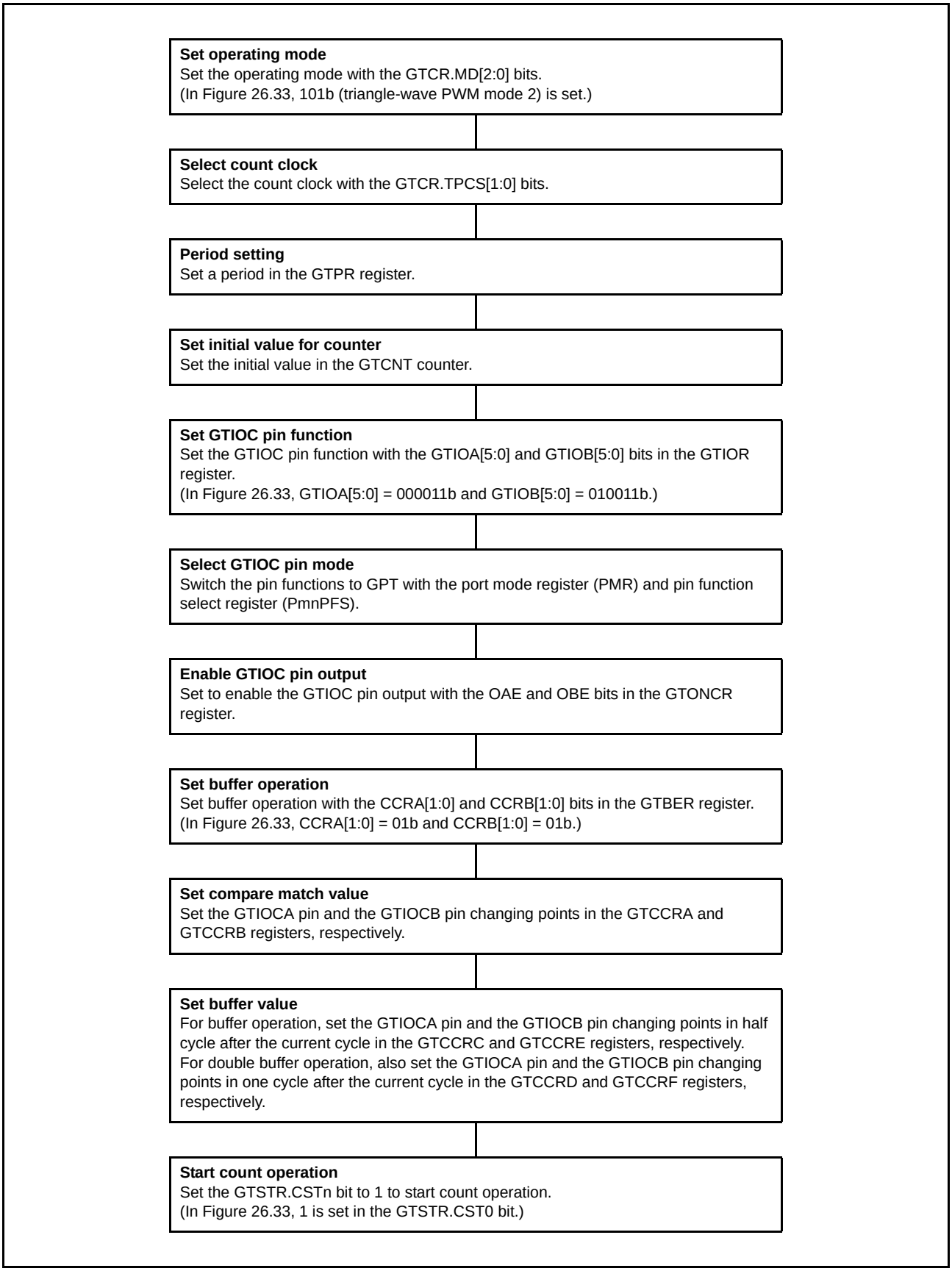


Figure 26.34 Example for Setting Triangle-Wave PWM Mode 2

(5) Triangle-Wave PWM Mode 3 (32-Bit Transfer at Trough)

The triangle-wave PWM mode 3 is a mode in which the period is set in the GPTn.GTPR register, the GPTn.GTCNT counter performs triangle-wave (full-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin at a compare match of the GPTn.GTCCRA or GPTn.GTCCRB register with buffer operation fixed (n = 0 to 3). Buffer operation in triangle-wave PWM mode 3 is different from the usual buffer operation. Buffer transfer is performed from the GTCCRC register to the GTCCRA register, from the GTCCRE register to the GTCCRB register, from the GTCCRD register to temporary register A, and from the GTCCRF register to temporary register B at the trough, and from temporary register A to the GTCCRA register and from temporary register B to the GTCCRB register at the crest. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR setting.

By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

Figure 26.35 shows an example of triangle-wave PWM mode 3 operation, and Figure 26.36 shows an example for setting triangle-wave PWM mode 3.

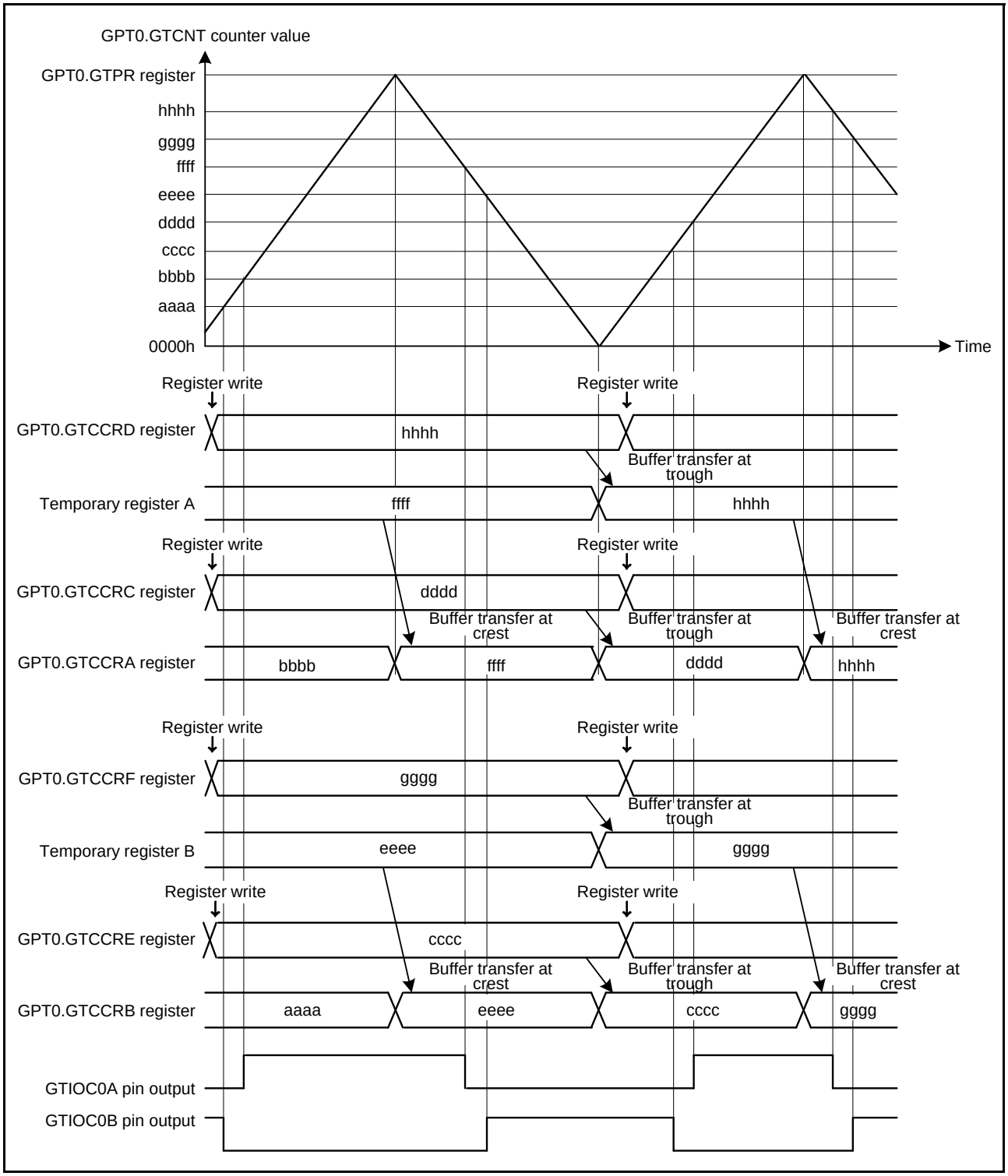


Figure 26.35 Example of Triangle-Wave PWM Mode 3 Operation (Low Output from the GTIOC0A Pin and High Output from the GTIOC0B Pin at Initial Output, Output Toggled at GTCCRA/GTCCRB Compare Match, Output Retained at the End of the Cycle)

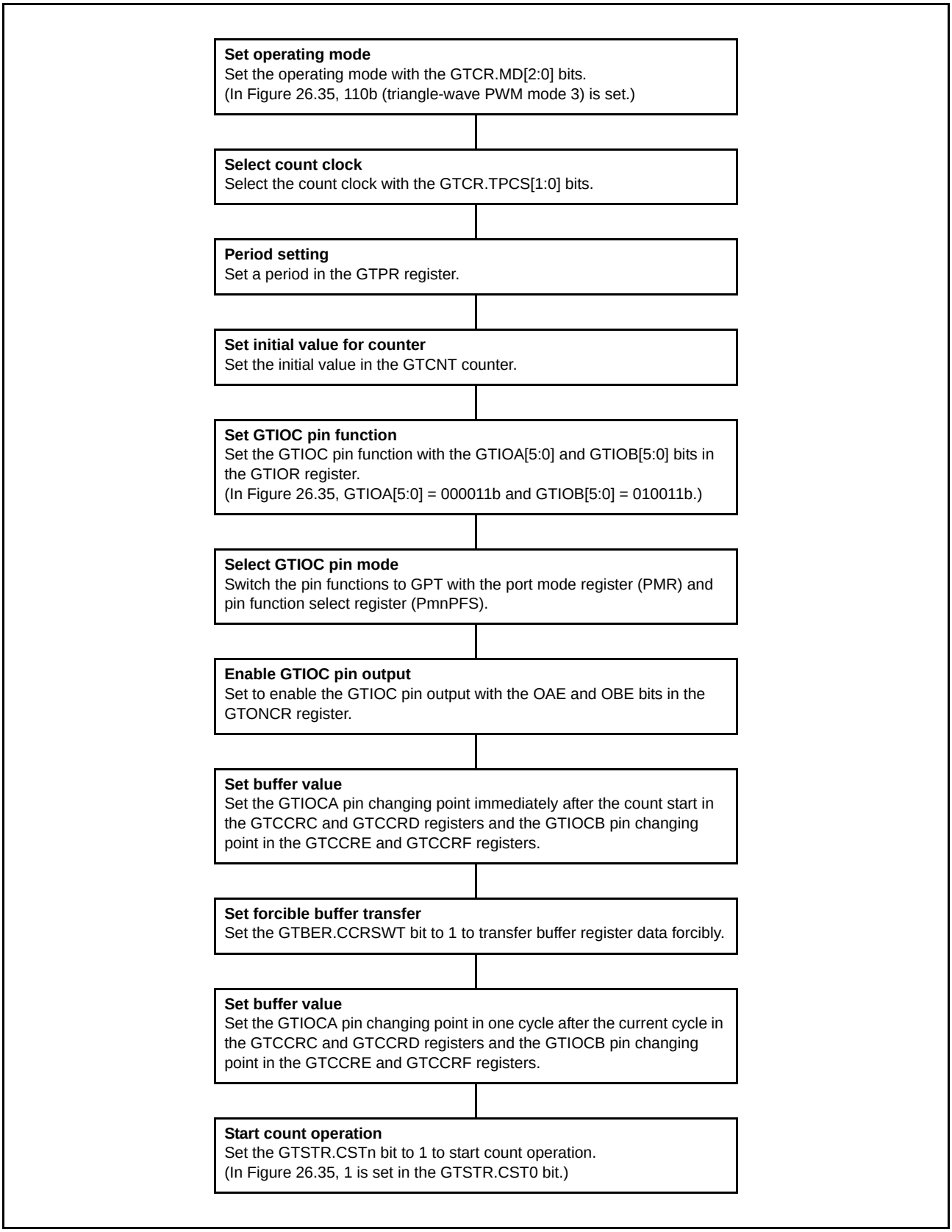


Figure 26.36 Example for Setting Triangle-Wave PWM Mode 3

26.3.4 Automatic Dead Time Setting Function

By setting the GTDTCR register, a compare match value for a negative waveform with dead time obtained by a compare match value for a positive waveform (the GTCCRA value) and specified dead time values (the GTDVU and GTDVD values) can automatically be set to the GTCCRB register.

The automatic dead time setting function can be used in saw-wave one-shot pulse mode and all the triangle PWM modes. Dead time can be separately set for the first half and second half of a waveform. Dead time for the changing point in the first half of a negative waveform is set in the GTDVU register and that in the second half is set in the GTDVD register. The same dead time can also be set for the first and second halves.

The GTDBU register can be used as a buffer register of the GTDVU register, and the GTDBD register can be used as a buffer register of the GTDVD register. Buffer transfer is performed at the end of the cycle (in saw-wave mode, either of an overflow of the GTCNT counter (up-counting), an underflow (down-counting), or counter clearing by a hardware source, software, event input, or synchronous clearing; in triangle-wave mode, a trough).

The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. Writing to the GTCCRB register is prohibited when the automatic dead time setting function is used. In addition, do not set the dead-time that makes the change point of the waveform exceeding the count period.

Figure 26.37 to Figure 26.40 show examples of automatic dead time setting function operation. Figure 26.41 and Figure 26.42 show the setting examples.

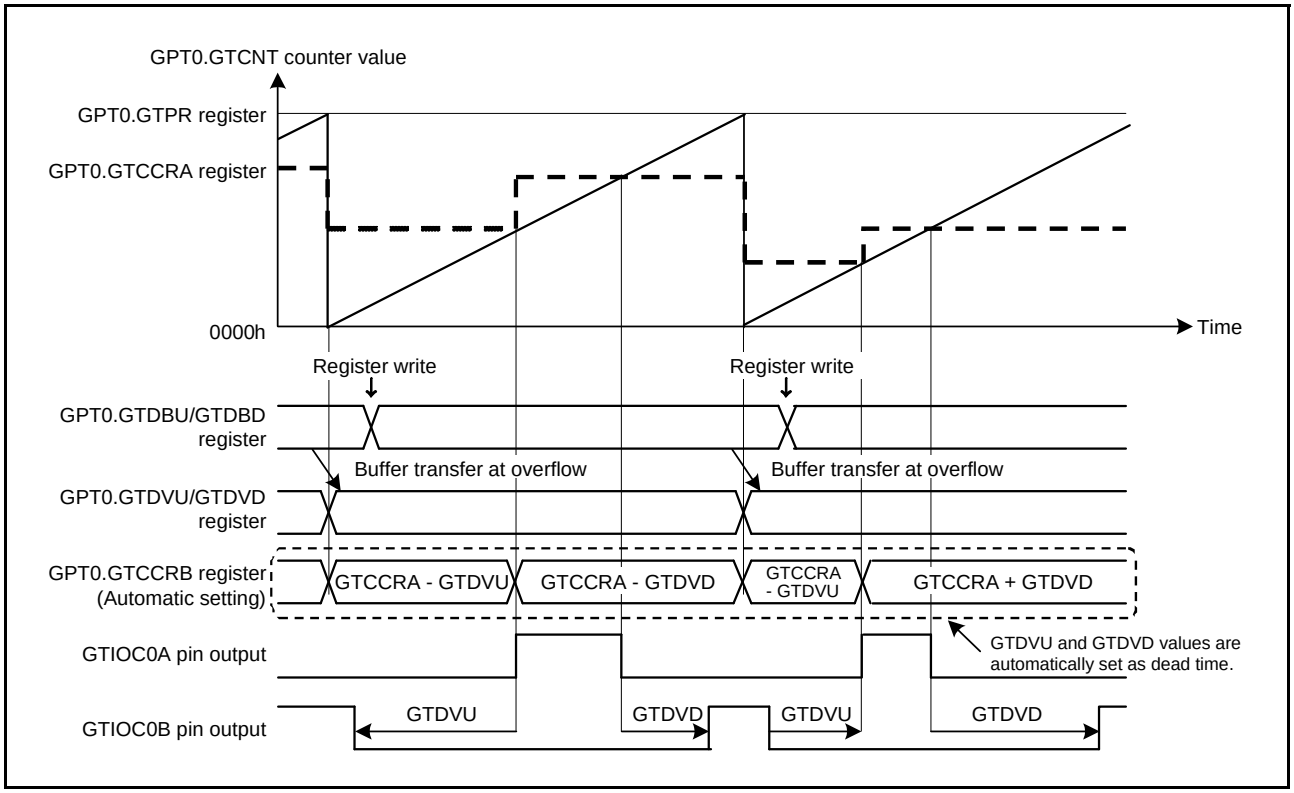


Figure 26.37 Example of Automatic Dead Time Setting Function Operation (Saw-Wave One-Shot Pulse Mode, Up-Counting, GTDVU and GTDVD Set to Buffer Operation, Active Level: High)

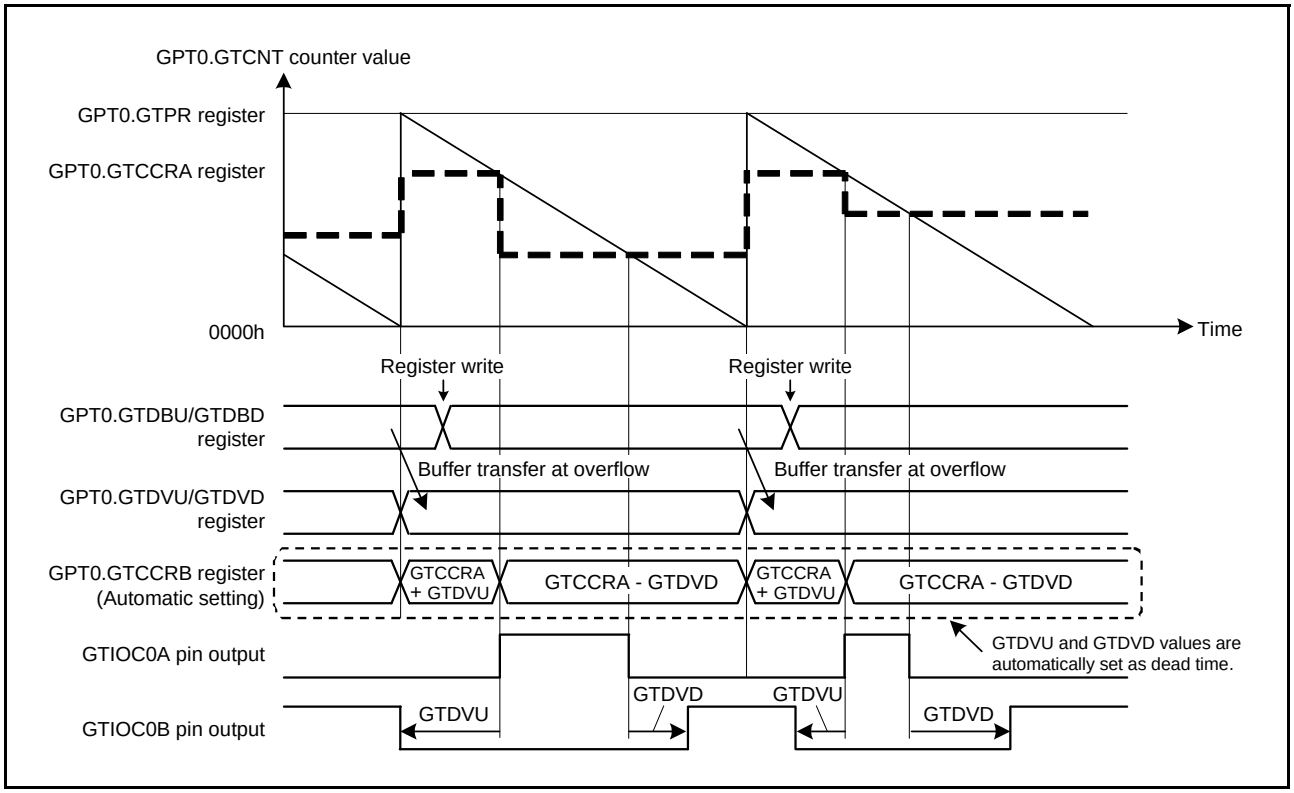


Figure 26.38 Example of Automatic Dead Time Setting Function Operation (Saw-Wave One-Shot Pulse Mode, Down-Counting, GTDVU and GTDVD Set to Buffer Operation, Active Level: High)

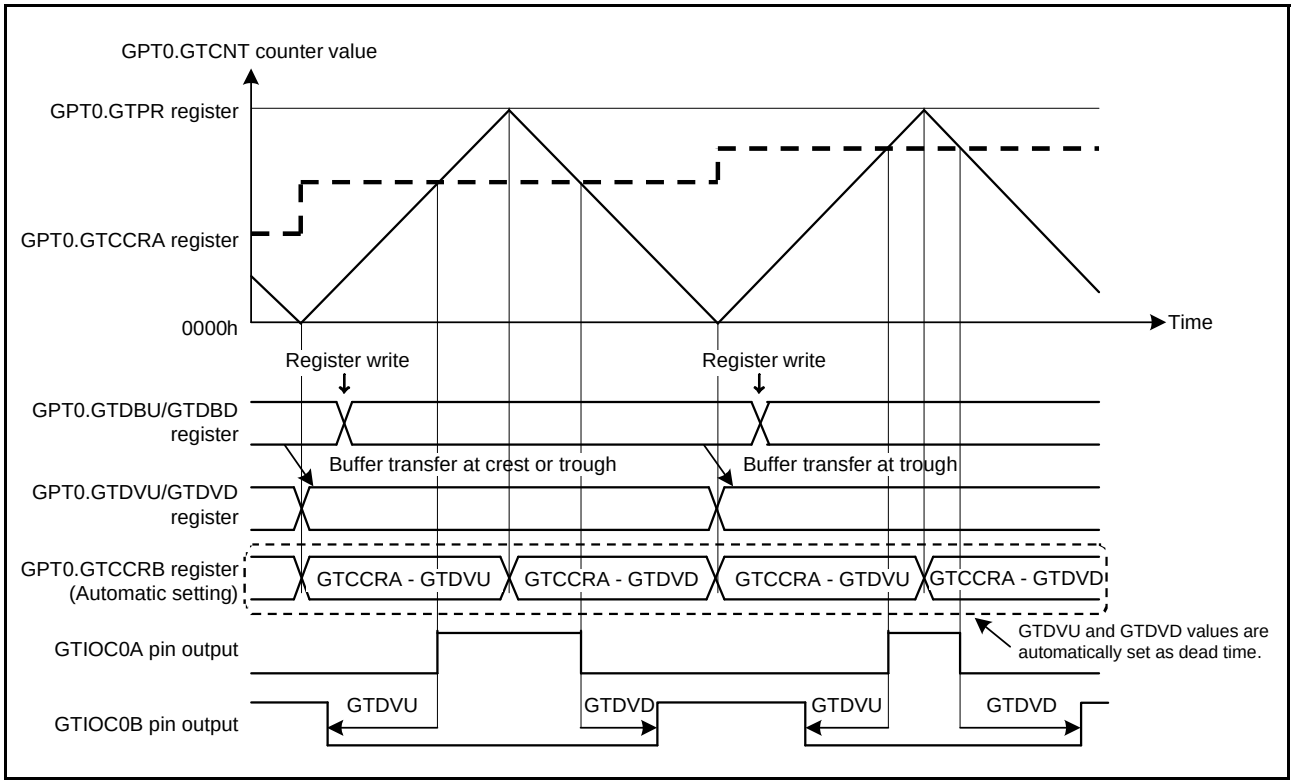


Figure 26.39 Example of Automatic Compare-Match Value Setting Function with Dead Time (Triangle-Wave PWM Mode 1, GTDVU and GTDVD Set to Buffer Operation, Active Level: High)

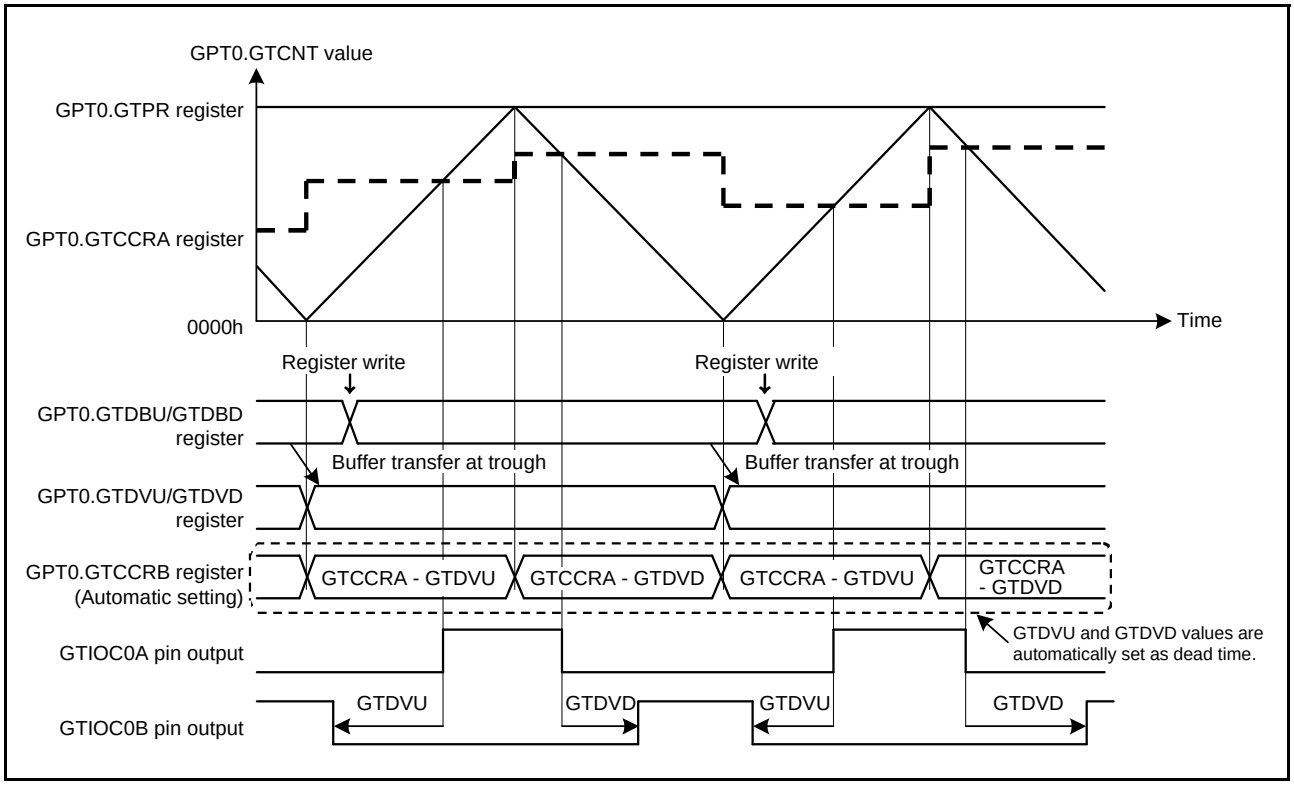


Figure 26.40 Example of Automatic Compare-Match Value Setting Function with Dead Time (Triangle-Wave PWM Mode 2 or 3, GTDVU and GTDVD Set to Buffer Operation, Active Level: High)

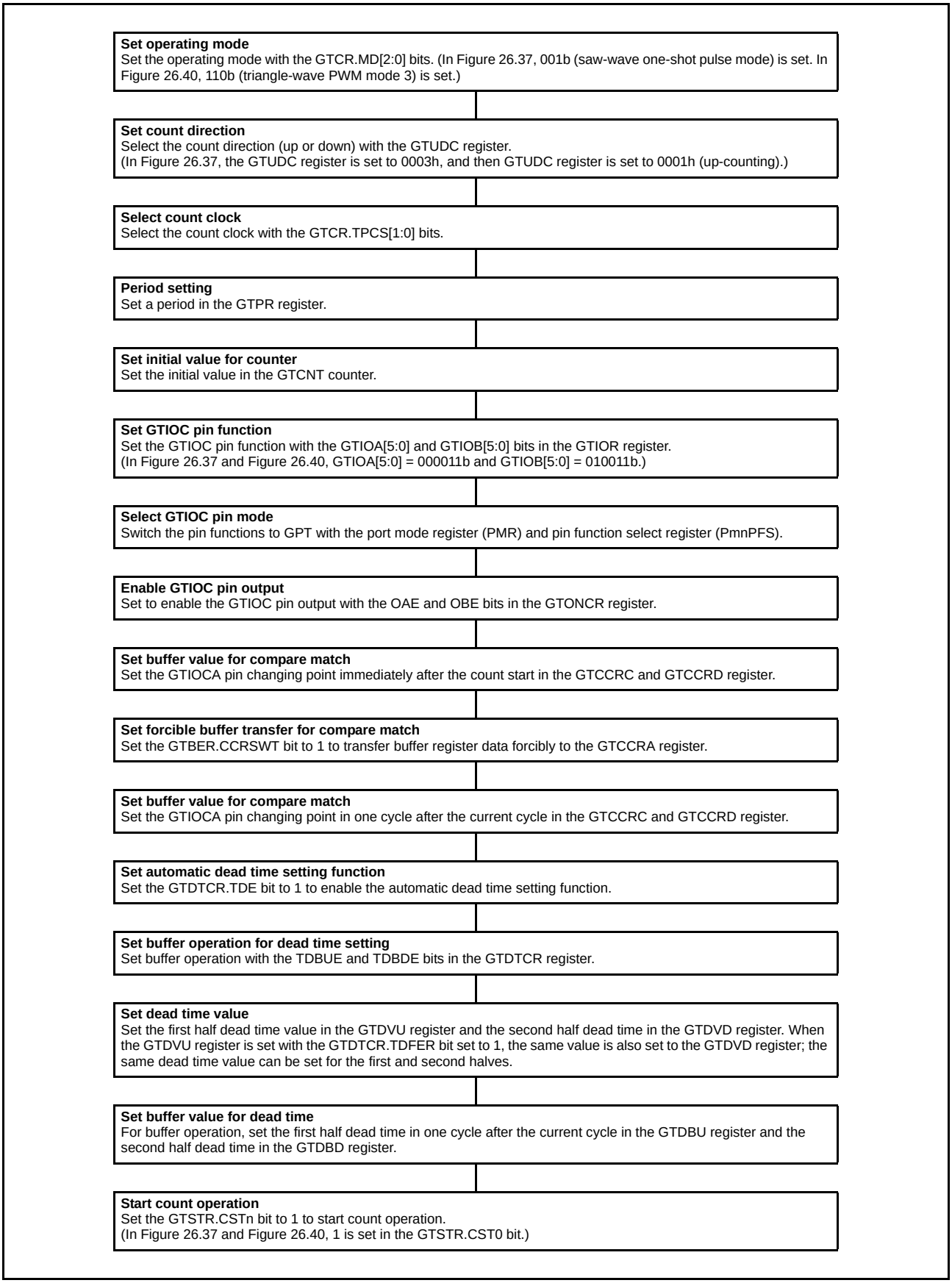


Figure 26.41 Example for Setting Automatic Dead Time Setting Function (Saw-Wave One-Shot Pulse Mode, Triangle-Wave PWM Mode 3)

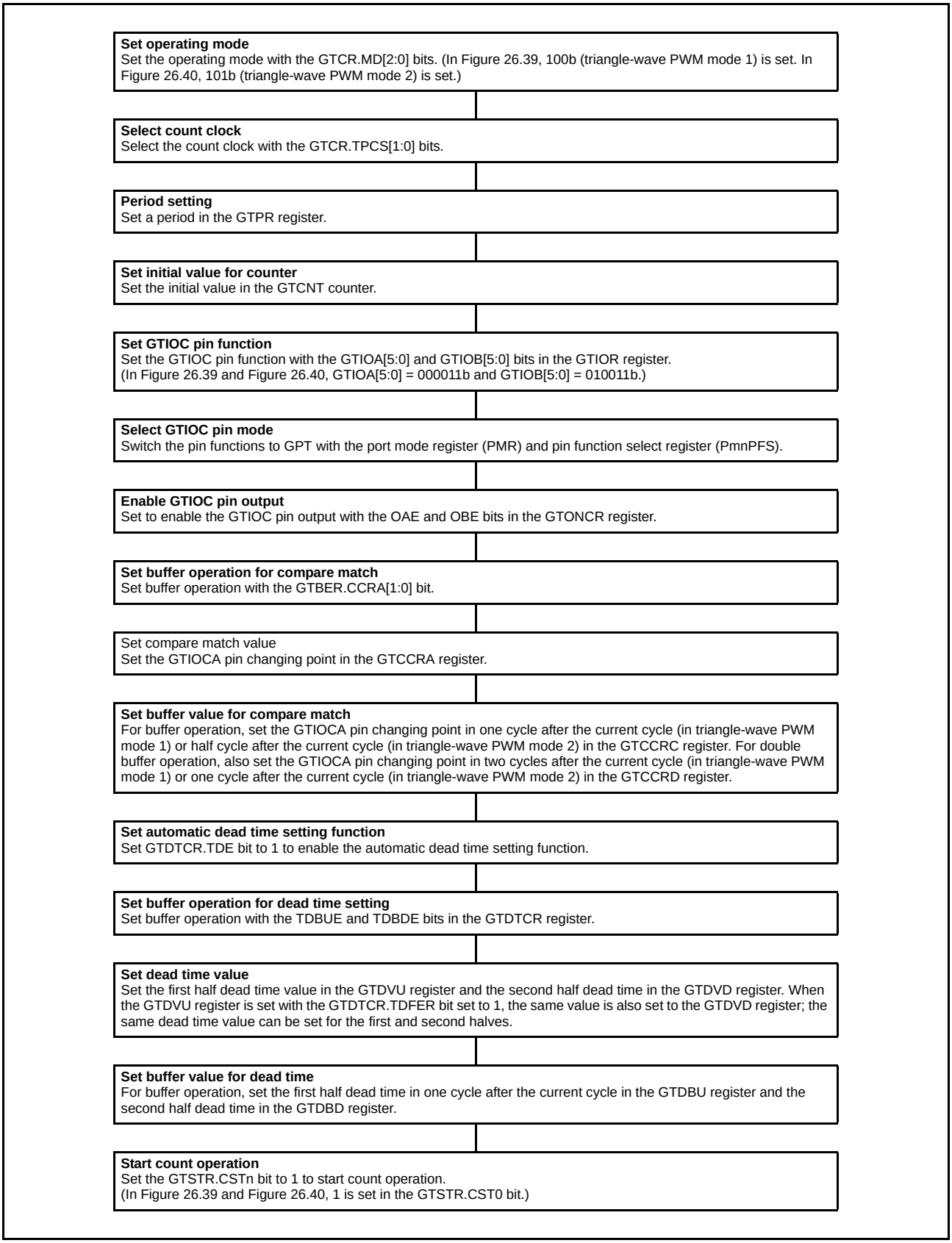


Figure 26.42 Example for Setting Automatic Dead Time Setting Function (Triangle-Wave PWM Mode 1 or 2)

26.3.5 Count Direction Changing Function

The count direction of the GTCNT counter can be changed by modifying the GTUDC.UD bit.
In saw-wave mode, if the GTUDC.UD bit is modified during count operation, the count direction is changed at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the UD bit is modified while count operation is stopped and the GTUDC.UDF bit is 0, the UD bit modification is not reflected at the start of counting and the count direction is changed at an overflow or an underflow. If the UDF bit is set to 1 while count operation is stopped, the UD bit value at that time is reflected at the start of counting.
In triangle-wave mode, the count direction is not changed even though the GTUDC.UD bit is modified during count operation. Similarly, even though the UD bit is modified while count operation is stopped and UDF bit is 0, the UD bit value is not reflected to the count operation. If the UDF bit is set to 1 while count operation is stopped, the UD bit value at that time is reflected at the start of counting.
If the count direction is changed from down-counting to up-counting during count operation in saw-wave mode, the GTST.TUCF flag changes from 0 to 1 at the end of the down-counting period (count clock while GTCNT counter value is 0). The GTPR register value after the start of up-counting is reflected in the count period, and the GTCNT counter starts up-counting from 0.
If the count direction is changed from up-counting to down-counting during count operation in saw-wave mode, the GTST.TUCF flag changes from 1 to 0 at the end of the up-counting period (count clock while GTCNT counter value is GTPR value). The GTPR register value before the start of down-counting is reflected in the count period, and the GTCNT counter starts down-counting from the GTPR register value.

Figure 26.43 shows an example of count direction changing function operation.

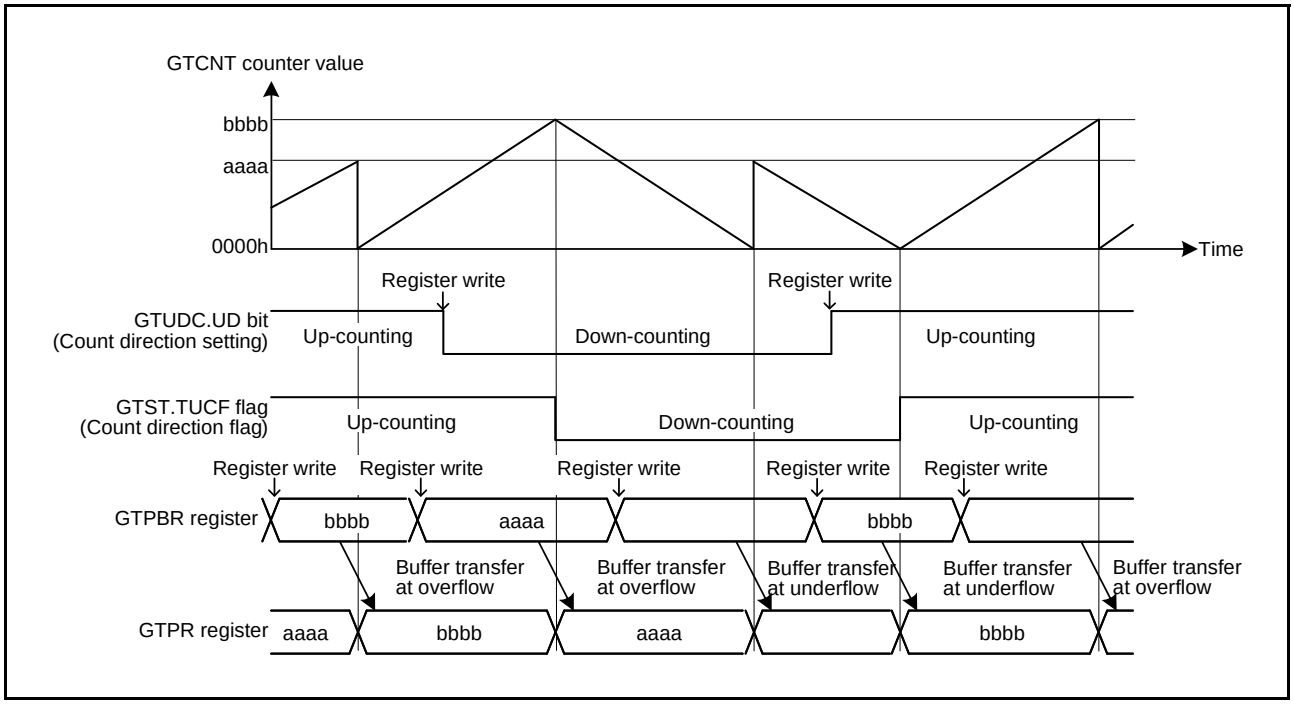


Figure 26.43 Example of Count Direction Changing Function Operation (during Buffer Operation)

26.3.6 Hardware Count Start/Count Stop and Clear Operation

The GTCNT counter can be started, stopped, or cleared by hardware sources in this MCU.
There are 3 types of hardware sources, including GTETRG pin input, GTIOC3A and GTIOC3B pin inputs, and GTIOC3A and GTIOC3B pin internal outputs (output compare).
The GTCNT counter can also be cleared by the GTCCRA or GTCCRB input capture.

26.3.6.1 Hardware Start Operation

The GTCNT counter can be started by a hardware source. Select a hardware source to start counting using the GTHSSR.CSHSLn[3:0] bits (n = 0 to 3), set the edge polarity for the hardware source with the GTHSCR.CSHWn[1:0] bits, and then enable to start counting.
Figure 26.44 shows an example of count start operation by a hardware source. Figure 26.45 shows the setting example.

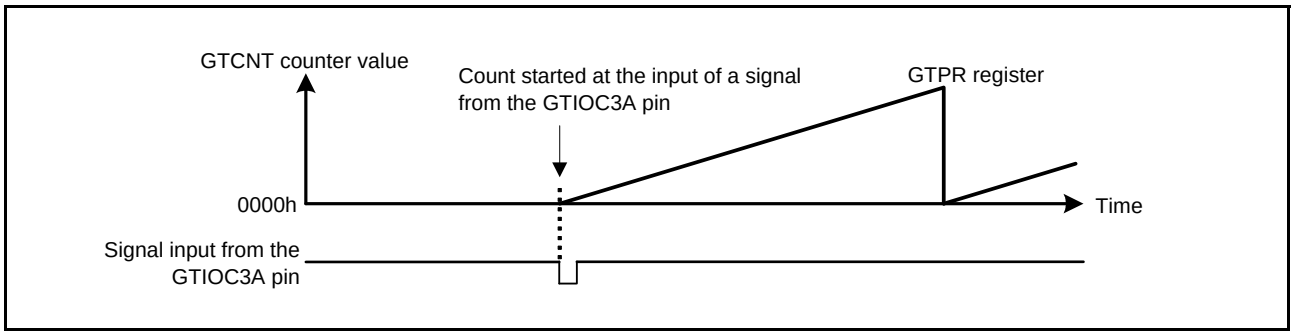


Figure 26.44 Example of Count Start Operation by Hardware Source (Started at Input of Signal from the GTIOC3A Pin)

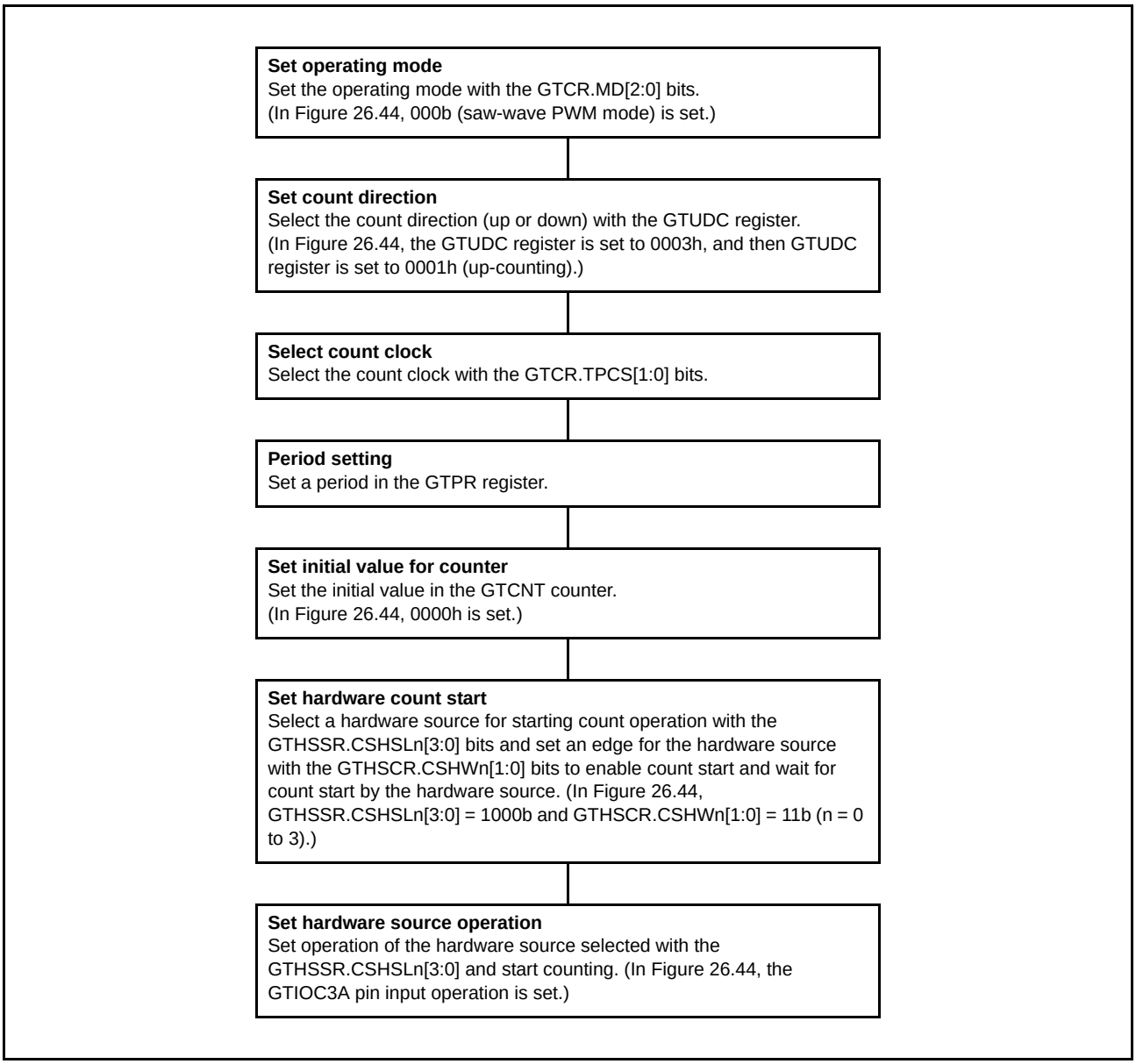


Figure 26.45 Example for Setting Count Start Operation by Hardware Source

26.3.6.2 Hardware Stop Operation

The GTCNT counter can be stopped by a hardware source. Select a hardware source to stop counting using the GTHPSR.CSHPLn[3:0] bits (n = 0 to 3), set the edge polarity for the hardware source with the GTHSCR.CPHWn[1:0] bit, and then enable to stop counting.

Figure 26.46 shows an example of count stop operation by a hardware source. Figure 26.47 shows the setting example. In this example, the count operation is stopped at both edges of the GTIOC3A pin internal output (output compare) and is restarted at both edges of the GTIOC3B pin internal output (output compare).

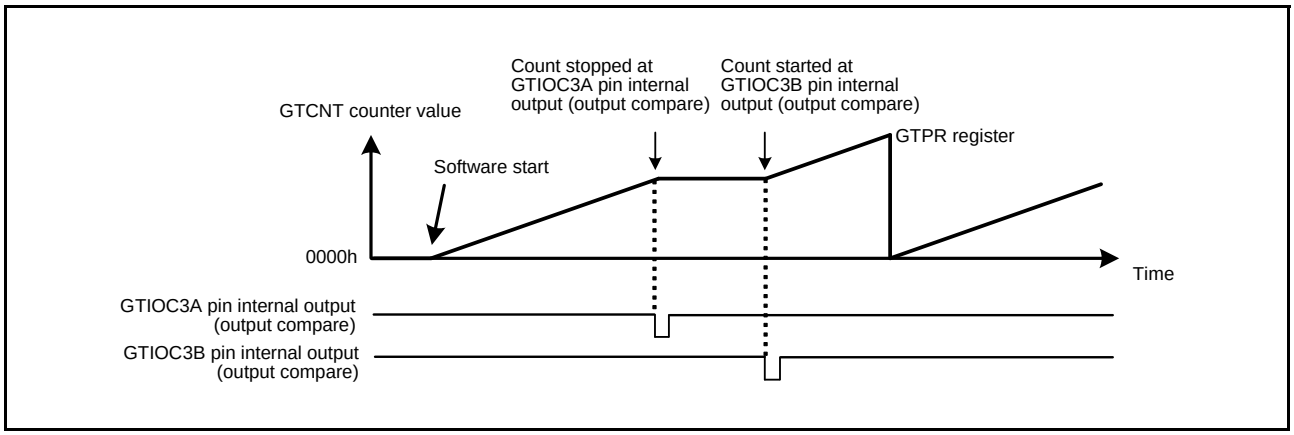


Figure 26.46 Example of Count Stop Operation by Hardware Source (Started by Software, Stopped at GTIOC3A Pin Internal Output (Output Compare), Restarted at GTIOC3B Pin Internal Output (Output Compare))

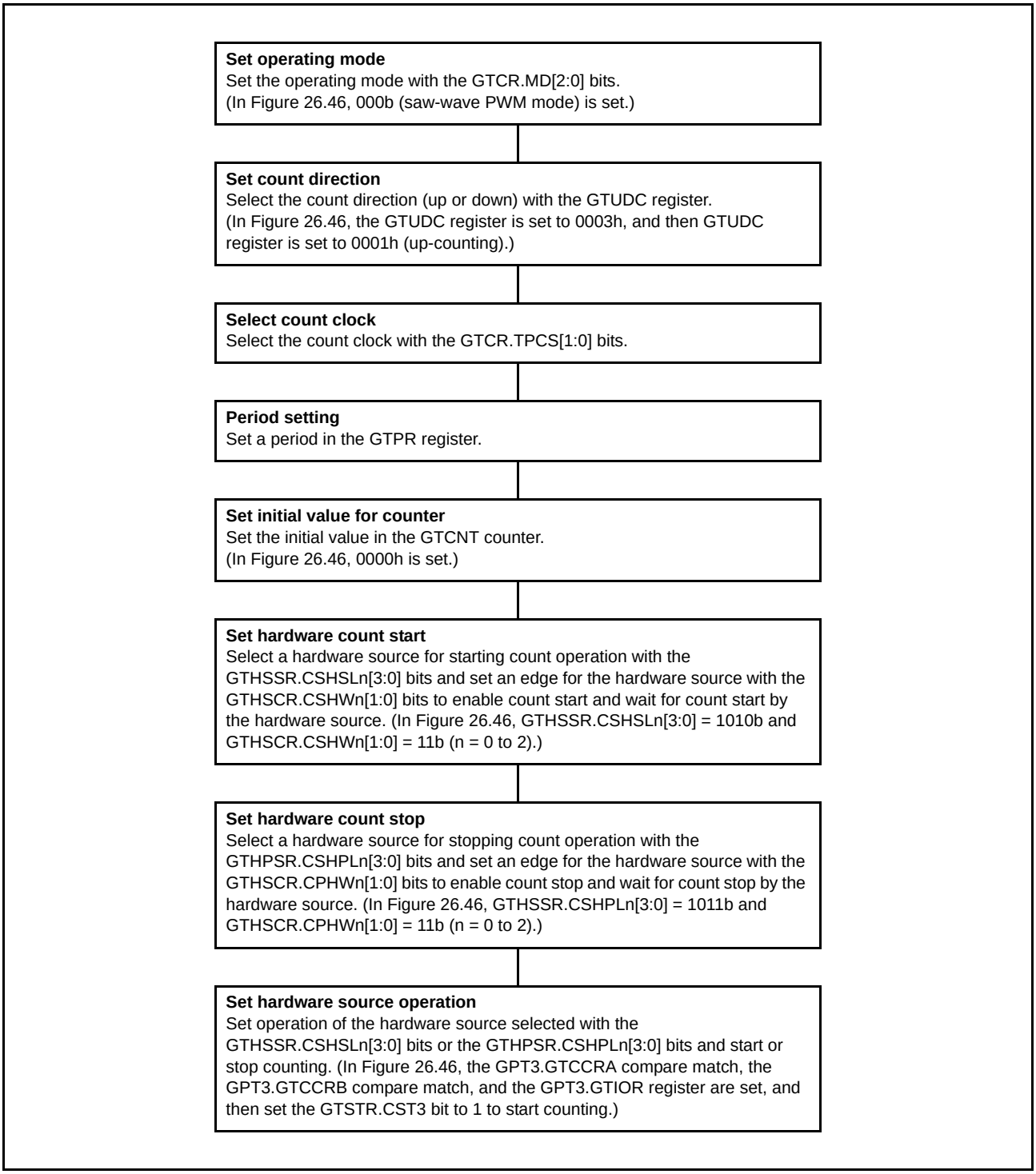


Figure 26.47 Example for Setting Count Stop Operation by Hardware Source

Figure 26.48 shows an example of count start/stop operation by a hardware source. Figure 26.49 shows the setting example. In this example, the counter operates during the high-level periods of the external trigger input GTETRG.

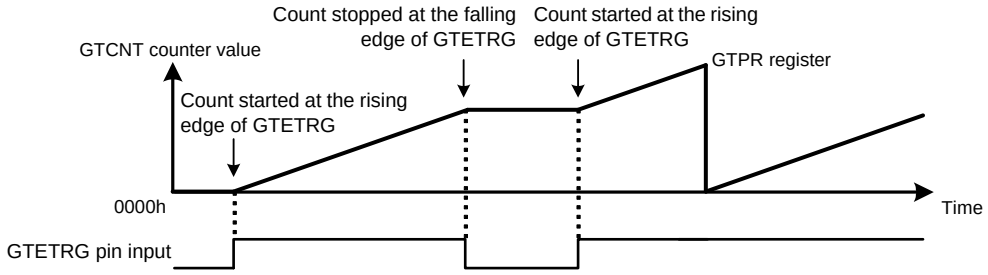


Figure 26.48 Example of Count Start/Stop Operation by Hardware Source (Started at Rising Edge of GTETRG Pin Input, Stopped at Falling Edge of GTETRG Pin Input)

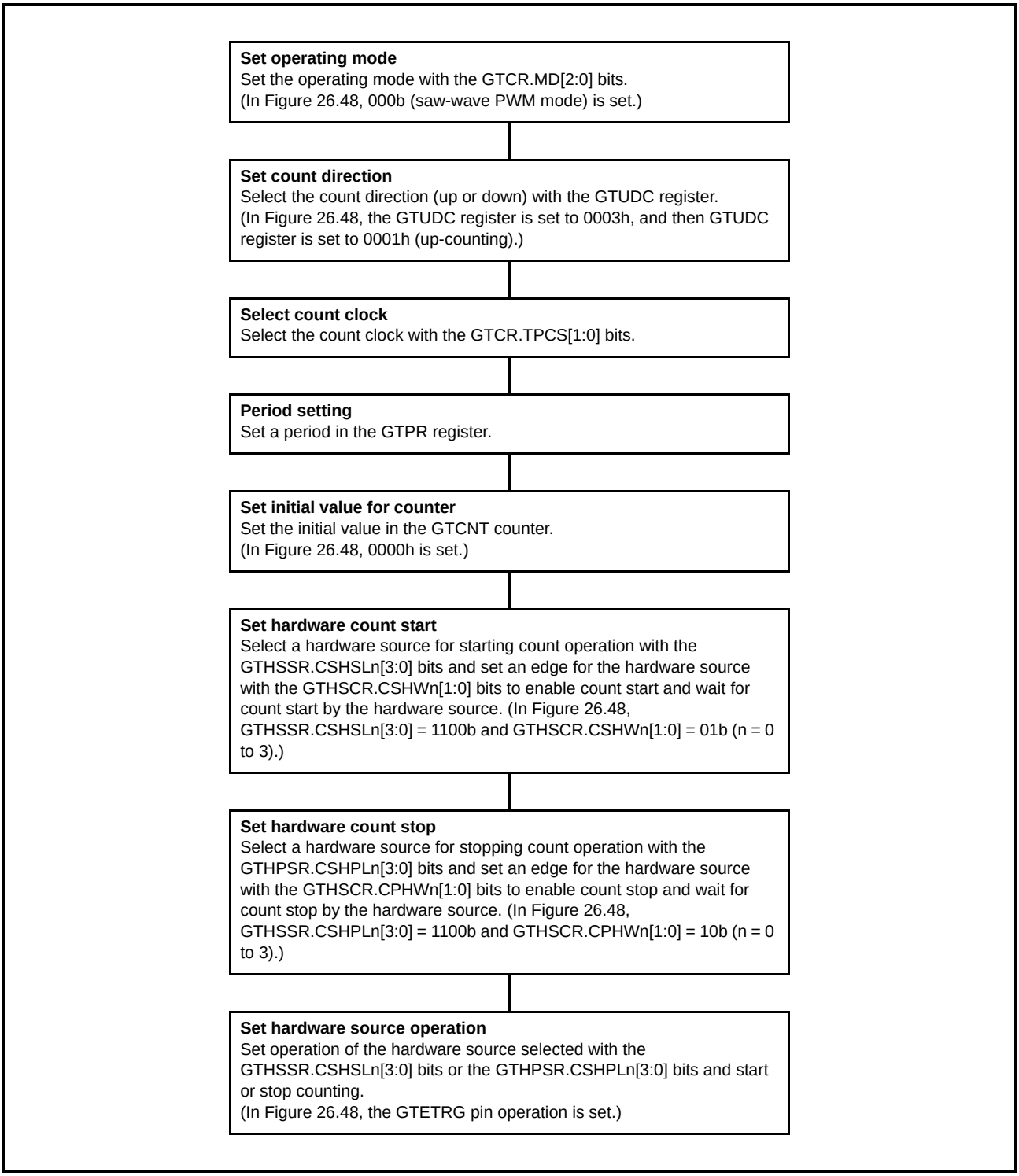


Figure 26.49 Example for Setting Count Start/Stop Operation by Hardware Source

26.3.6.3 Hardware Clear Operation

The GTCNT counter can be cleared by a hardware source. Select a hardware source to clear the counter using the GTHPSR.CSHPLn[3:0] bits (n = 0 to 3), set the edge polarity for the hardware source with the GTHCCR.CCHWn[1:0] bits, and enable clearing the counter.

The GTCNT counter can also be cleared by a GTCCRA or GTCCRB input capture by setting the GTCR.CCLR[1:0] bits.

When the count direction is down-counting (GTST.TUCF flag is 0) in saw-wave mode, the GTCNT counter is set to the value set in the GTPR register. In other cases, the GTCNT counter is set to 0000h when count clearing is executed.

Note that the GTCIV/GTCIU interrupt (overflow/underflow interrupt) is not generated when the GTCNT counter is cleared by a hardware source or by software.

Figure 26.50 and Figure 26.51 show examples of the GTCNT counter clearing operation by a hardware source. Figure 26.52 shows the setting example. Figure 26.53 shows the relationship between counter clearing by a hardware source and the GTCIV/GTCIU interrupt. In this example, the GTCNT counter is started at both edges of the GTIOC3A pin, and the counter is stopped/cleared at both edges of the GTOC3B pin input.

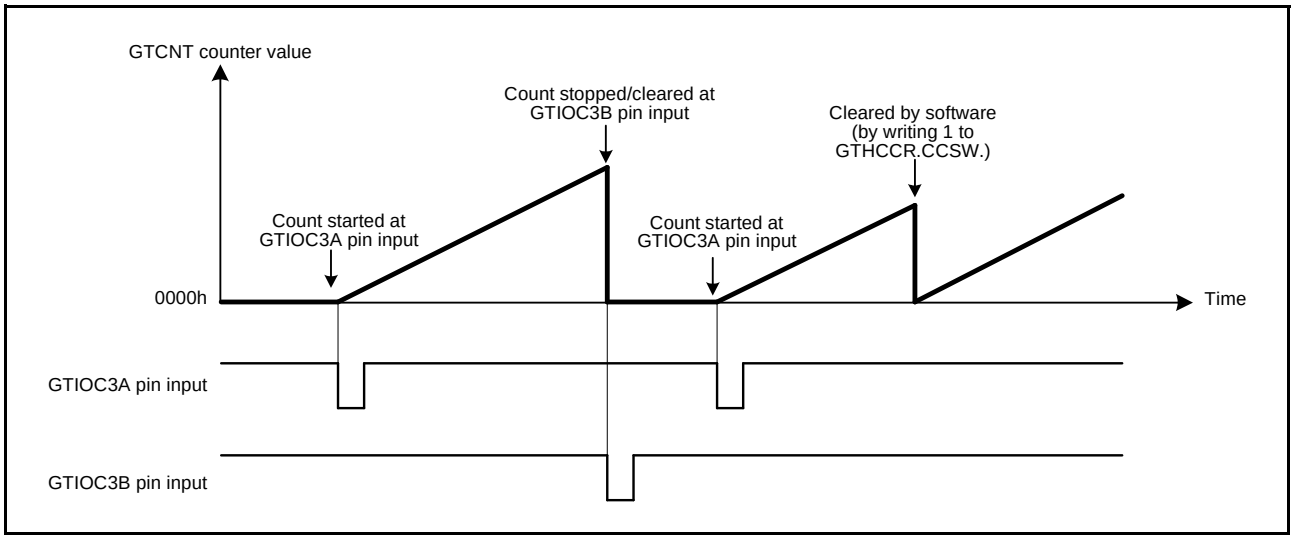


Figure 26.50 Examples of Count Clearing Operation by Hardware Source (Saw-Wave Up-Counting, Started at GTIOC3A Pin Input, Count Stopped/Count Cleared at GTIOC3B Pin Input)

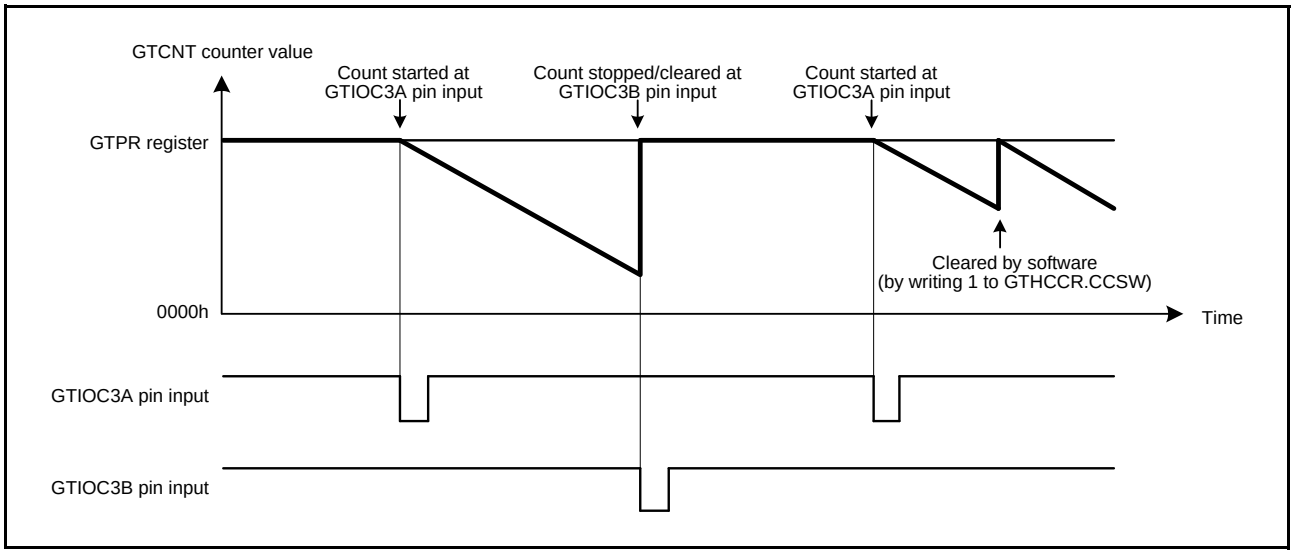


Figure 26.51 Examples of Count Clearing Operation by Hardware Source (Saw-Wave Down-Counting, Started at GTIOC3A Pin Input, Count Stopped/Count Cleared at GTIOC3B Pin Input)

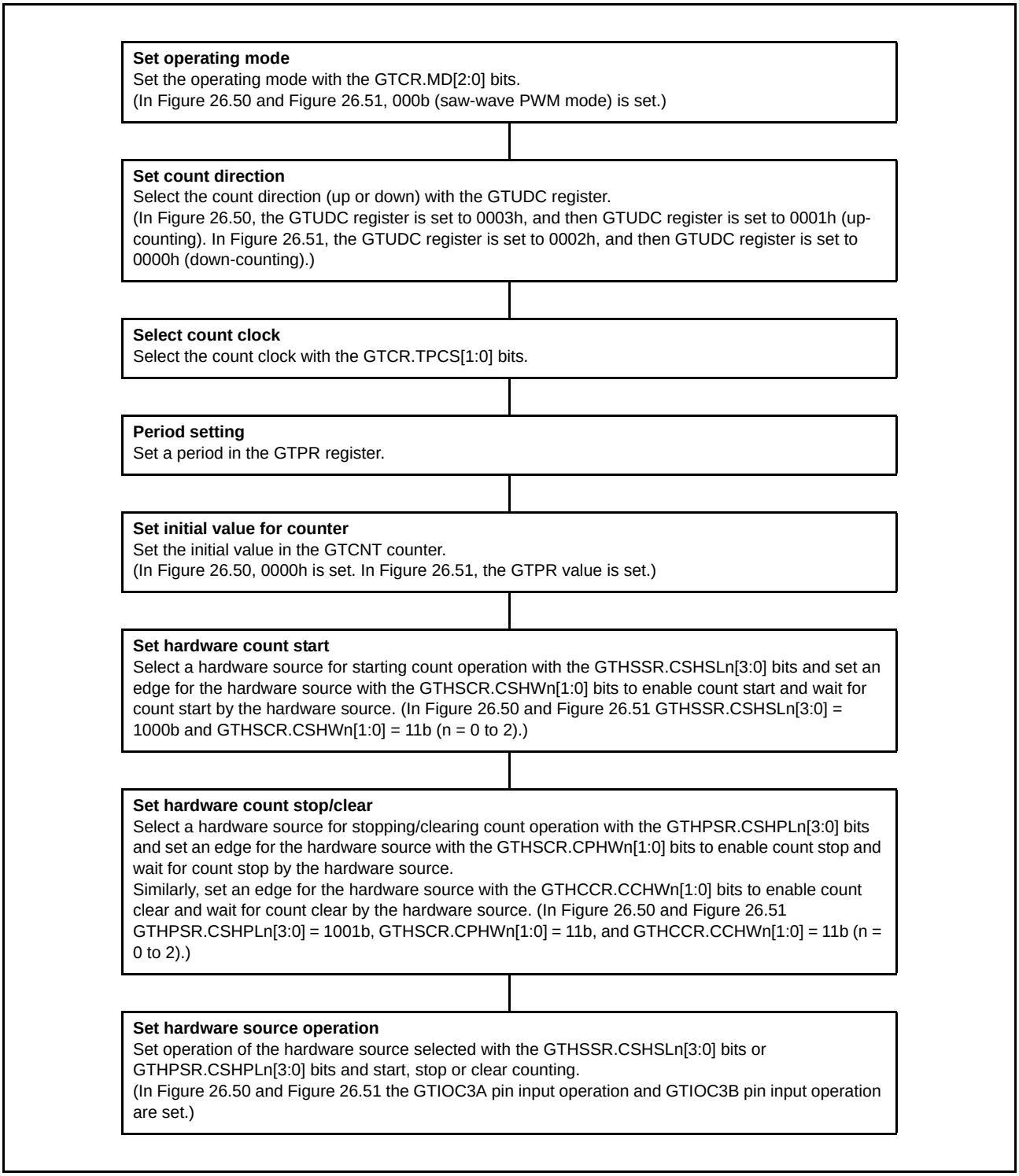


Figure 26.52 Example for Setting Count Clearing Operation by Hardware Source

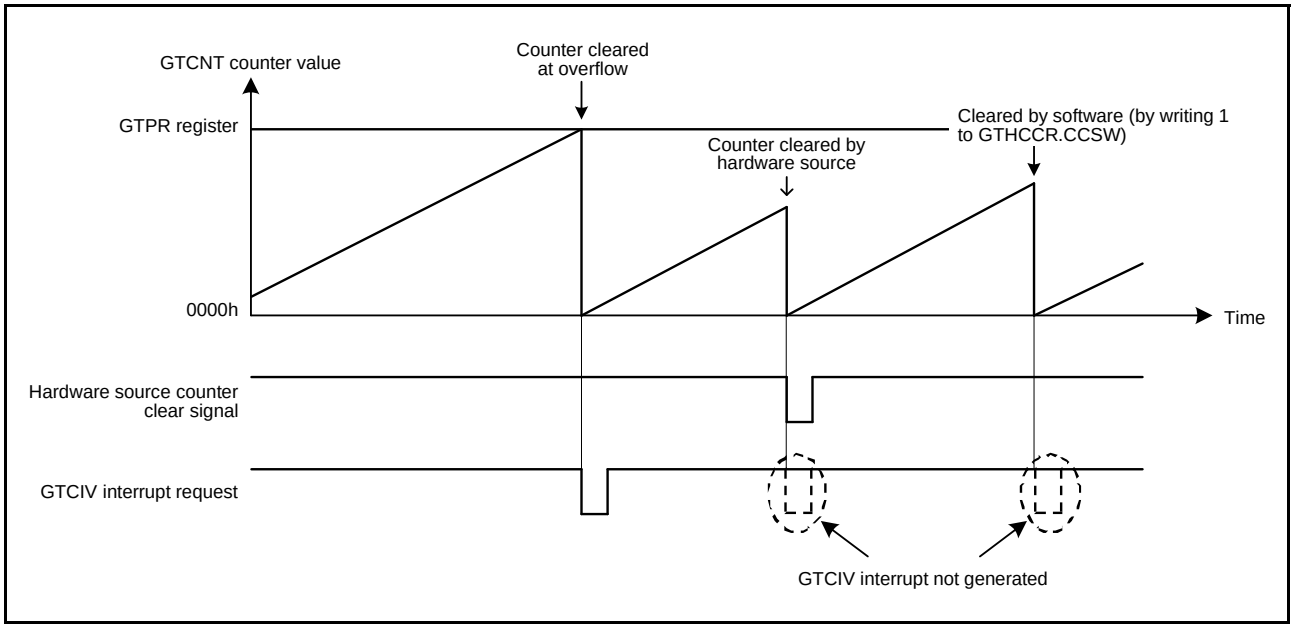


Figure 26.53 Relationship between Counter Clearing by Hardware Source and GTCIV Interrupt

26.3.7 Synchronous Operation

Synchronous operation on channels (synchronous clear operation, synchronous start operation) can be performed.

26.3.7.1 Synchronous Clear Operation

Synchronous clearing on channels can be controlled. Select which channels to be synchronously cleared by setting the GTCR.CCLR[1:0] bits of the pertinent channels to 11b and which channel clearing source to be used for synchronous clearing by setting the GTSYNC.SYNCn[1:0] bits (n = 0 to 3).

The sources used for synchronous clearing are a hardware source, software, event input, input capture, an overflow of saw waves (up-counting) and an underflow (down-counting).

Figure 26.54 shows an example of synchronous clear operation, and Figure 26.56 shows the setting example. In this example, the GPT1.GTCNT and GPT2.GTCNT counters are synchronously cleared by the GPT0.GTCNT clearing source (overflow).

Synchronous clearing of channels by a clear source does not cause synchronous clearing of another channel by the same clear source. (Synchronous clearing is not transmitted.)

Figure 26.55 shows an operation example in which two channels are synchronously cleared by the clear source of one of the channels and another channel is synchronously cleared by the clear source of the other one of the two channels.

Figure 26.56 shows the setting example. In this example, the GPT1.GTCNT counter is synchronously cleared by the GPT0.GTCNT counter clearing source (overflow), and the GPT2.GTCNT counter is synchronously cleared by the GPT1.GTCNT counter clearing source (overflow). Although the GPT1.GTCNT counter is synchronously cleared by the GPT0.GTCNT counter clearing source (overflow), the GPT2.GTCNT counter is not synchronously cleared when the GPT1.GTCNT counter is cleared by the GPT0.GTCNT counter clearing source.

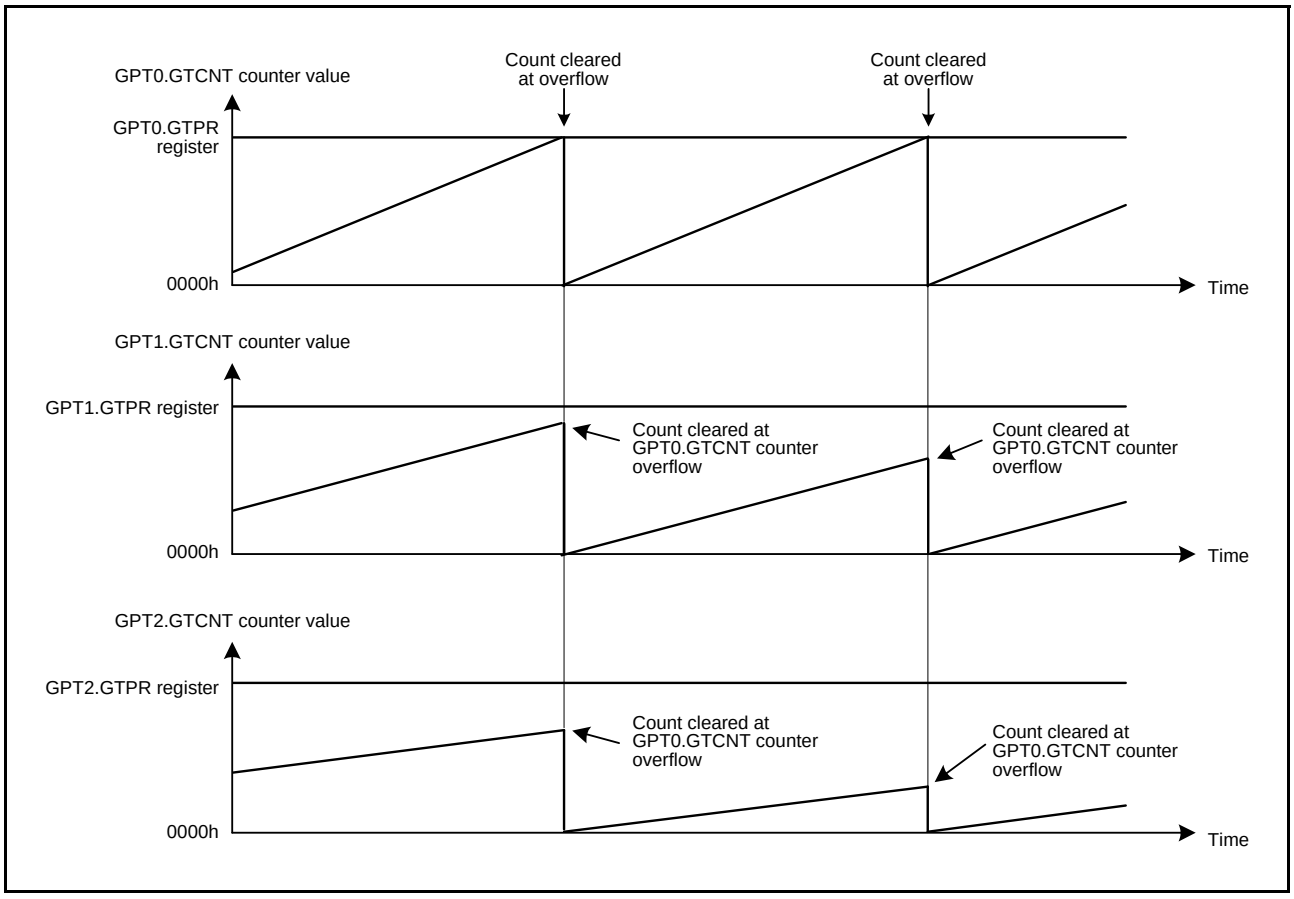


Figure 26.54 Example of Synchronous Clear Operation (GPT1.GTCNT and GPT2.GTCNT Counters are Synchronously Cleared by GPT0.GTCNT Counter Clearing Source)

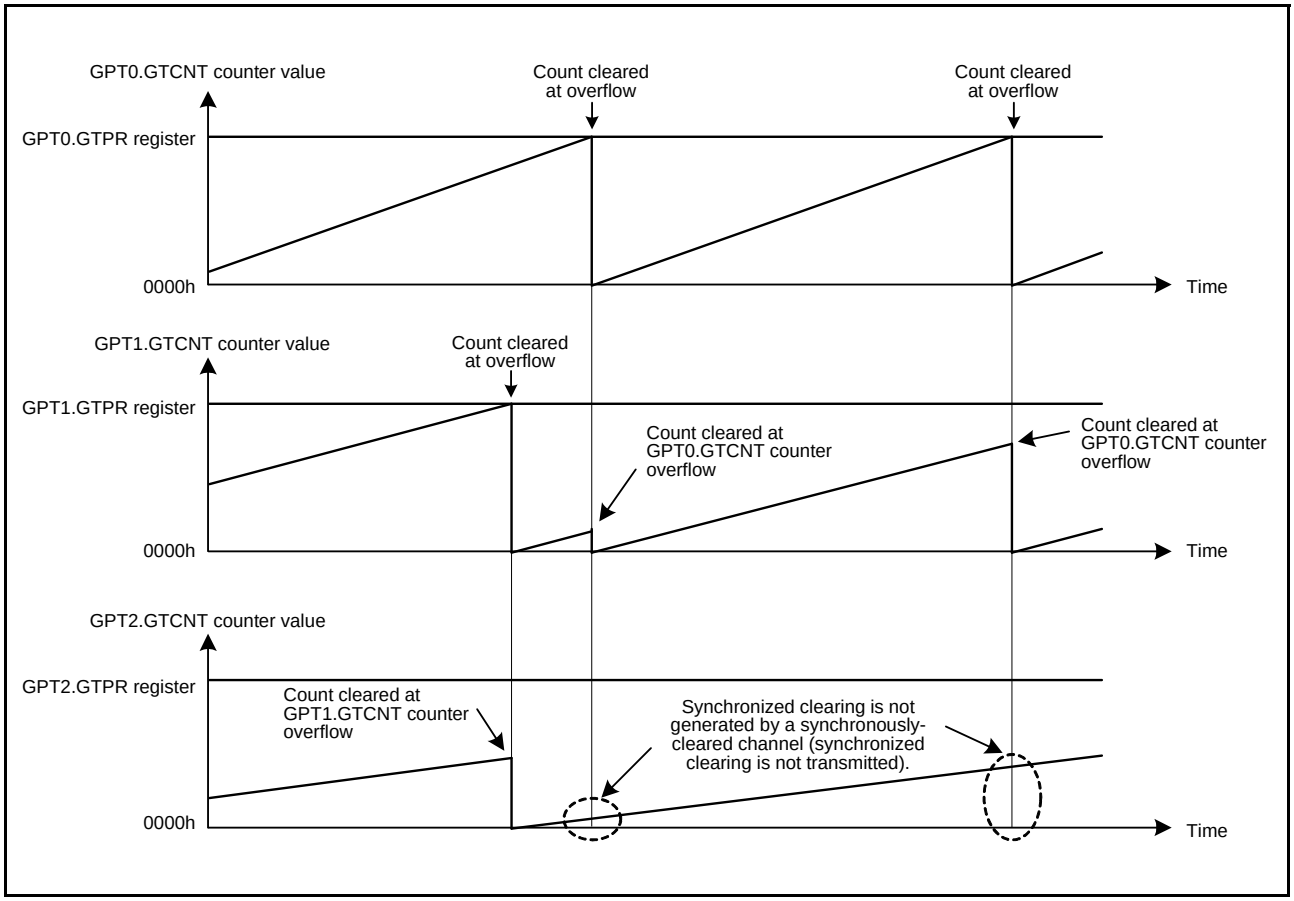


Figure 26.55 Example of Synchronous Clear Operation (GPT1.GTCNT Counter is Synchronously Cleared by GPT0.GTCNT Counter Clearing Source and GPT2.GTCNT Counter is Synchronously Cleared by GPT1.GTCNT Counter Clearing Source)

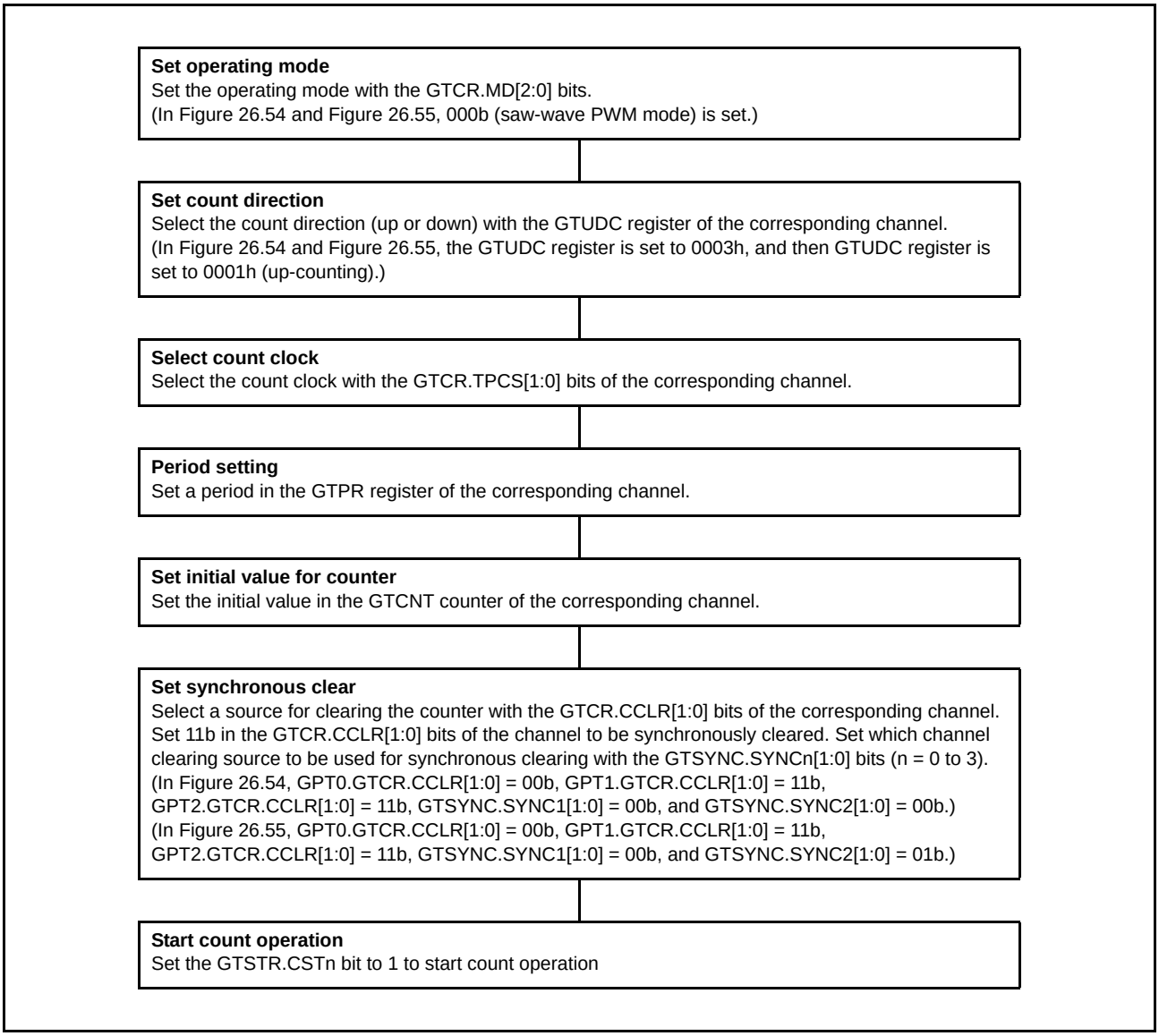


Figure 26.56 Example for Setting Synchronous Clear Operation

26.3.7.2 Synchronous Start Operation

(1) Simultaneous Start by Software

The GPTn.GTCNT counters can be started simultaneously on channels by simultaneously setting the GTSTR.CSTn bits which correspond to the channels to be started simultaneously to 1 (n = 0 to 3).
Figure 26.57 shows an example of simultaneous start by software.

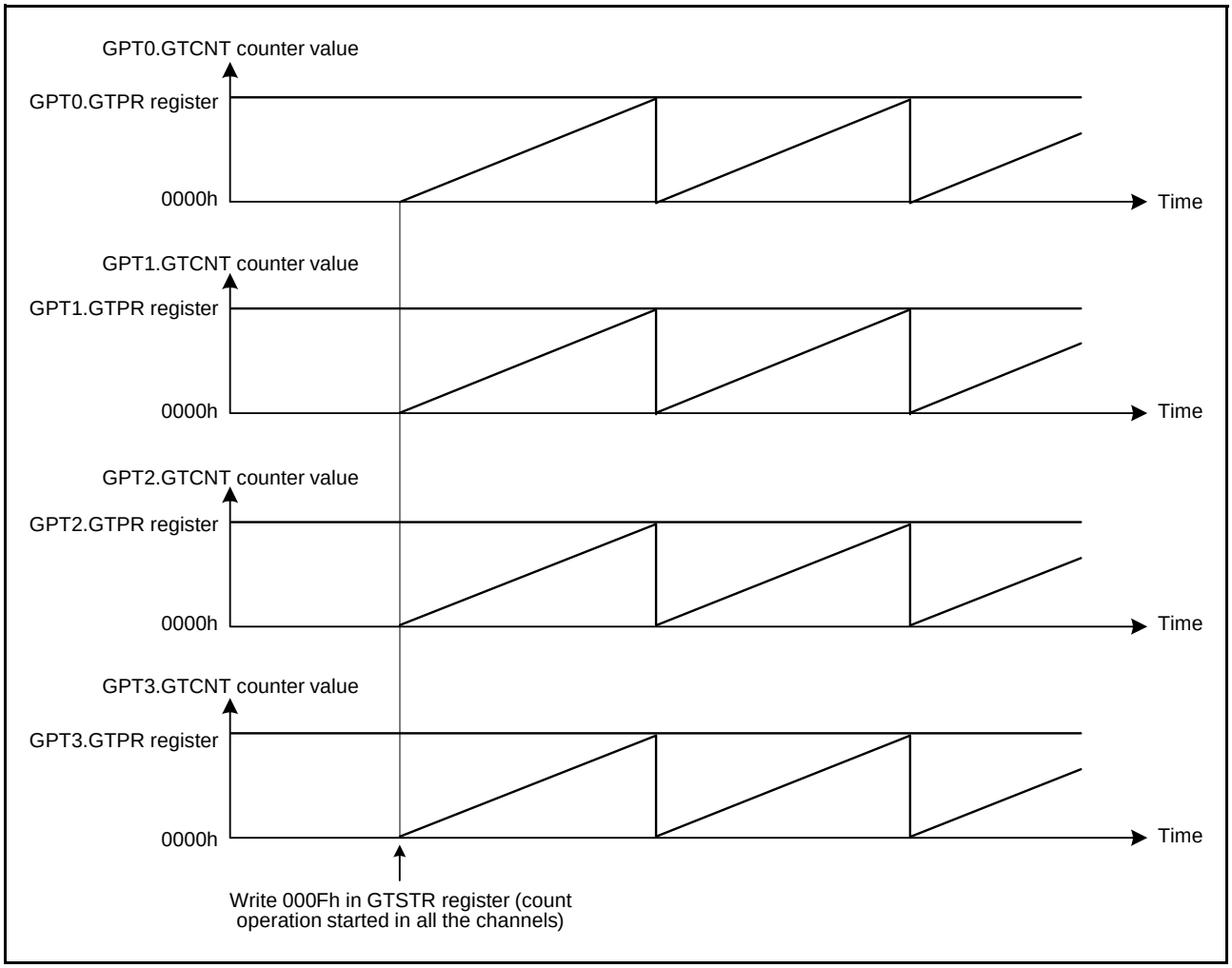


Figure 26.57 Example of Simultaneous Start by Software (with Same Count Period (GTPR Value))

(2) Phase Shift Start by Software

Count start with a phase difference is possible by setting the initial value in the GTCNT counter before counting starts and then simultaneously setting the GTSTR.CSTn bits which correspond to the channels to be started simultaneously to 1 (n = 0 to 3).

Figure 26.58 shows an example of phase shift start operation by software.

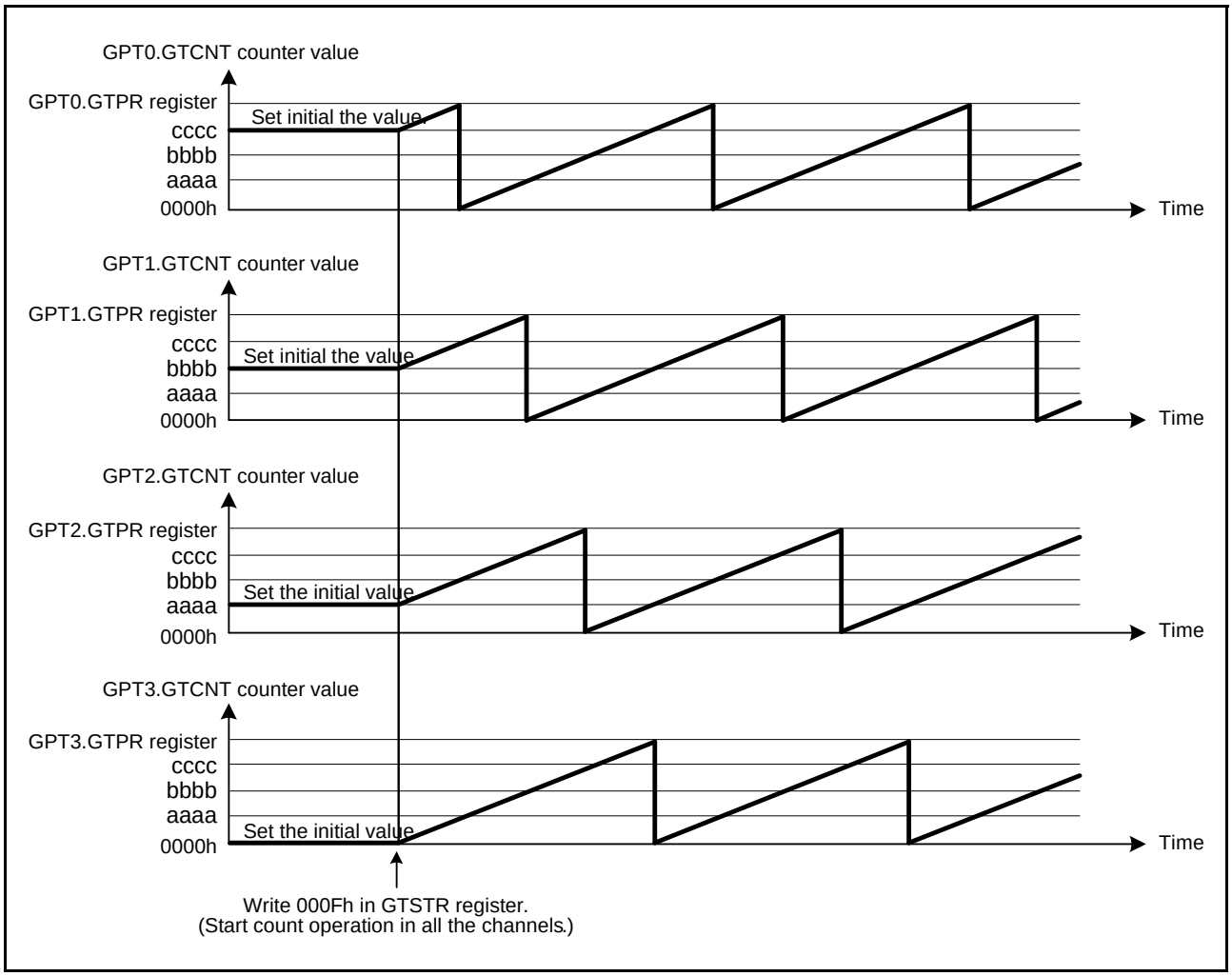


Figure 26.58 Example of Software Phase Shift Start (with Same Count Period (GTPR Register Value))

(3) Simultaneous Start by Hardware Source

The GPTn.GTCNT counters can be started simultaneously by following hardware sources: GTETRG pin input, GTIOC3A/GTIOC3B pin input, and GTIOC3A/GTIOC3B pin internal output (output compare) (n = 0 to 3). Figure 26.59 shows an example of simultaneous start operation by a hardware source and Figure 26.60 shows the setting example. In this example, count operation is started in all the channels by the input of a signal from the GTIOC3A pin.

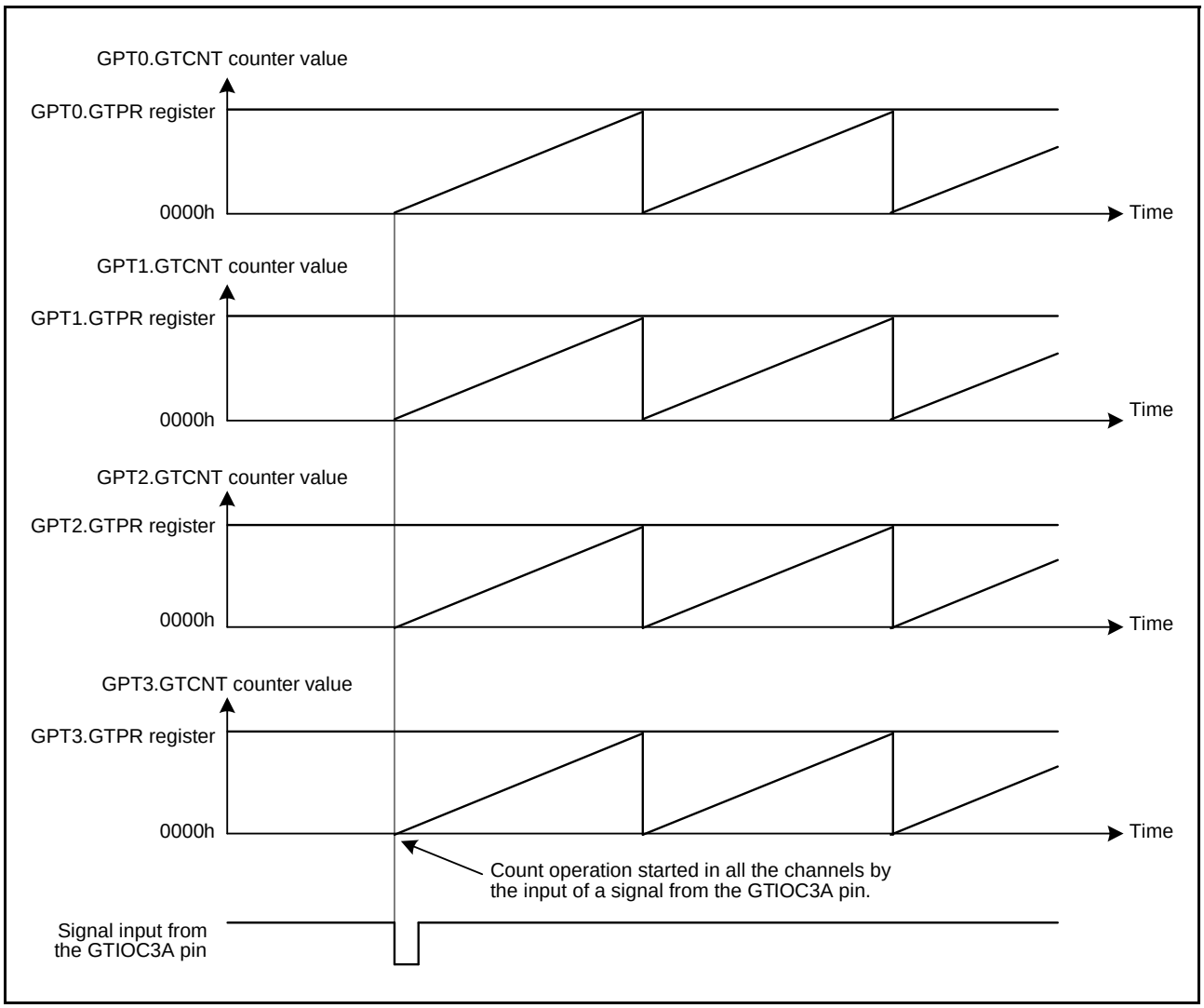


Figure 26.59 Example of Simultaneous Start Operation by Hardware Source (with Same Count Period (GTPR Value))

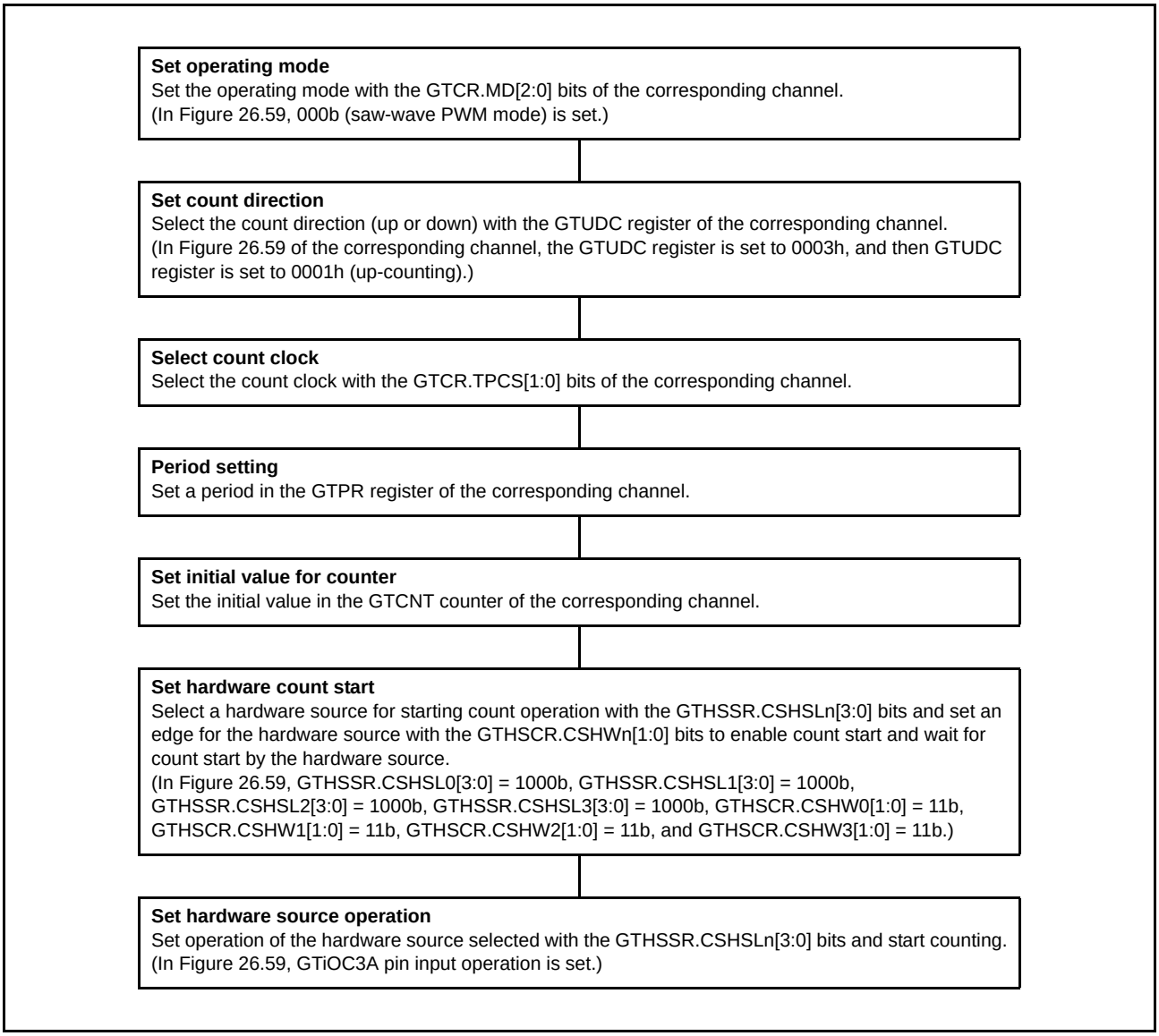


Figure 26.60 Example for Setting Simultaneous Start by Hardware Source

(4) Phase Shifting Start by Hardware Source

Count start with a phase difference is possible by following hardware sources: GTETRG pin input, GTIOC3A/GTIOC3B pin input, and GTIOC3A/GTIOC3B pin internal output (output compare).

Figure 26.61 shows an example of phase shifting start operation by a hardware source and Figure 26.62 shows the setting example. In this example, the GPT3.GTCNT and GPT0.GTCNT counters simultaneously start counting and the GPT1.GTCNT and GPT2.GTCNT counters start counting by the GTIOC3A and GTIOC3B pin internal outputs (output compare).

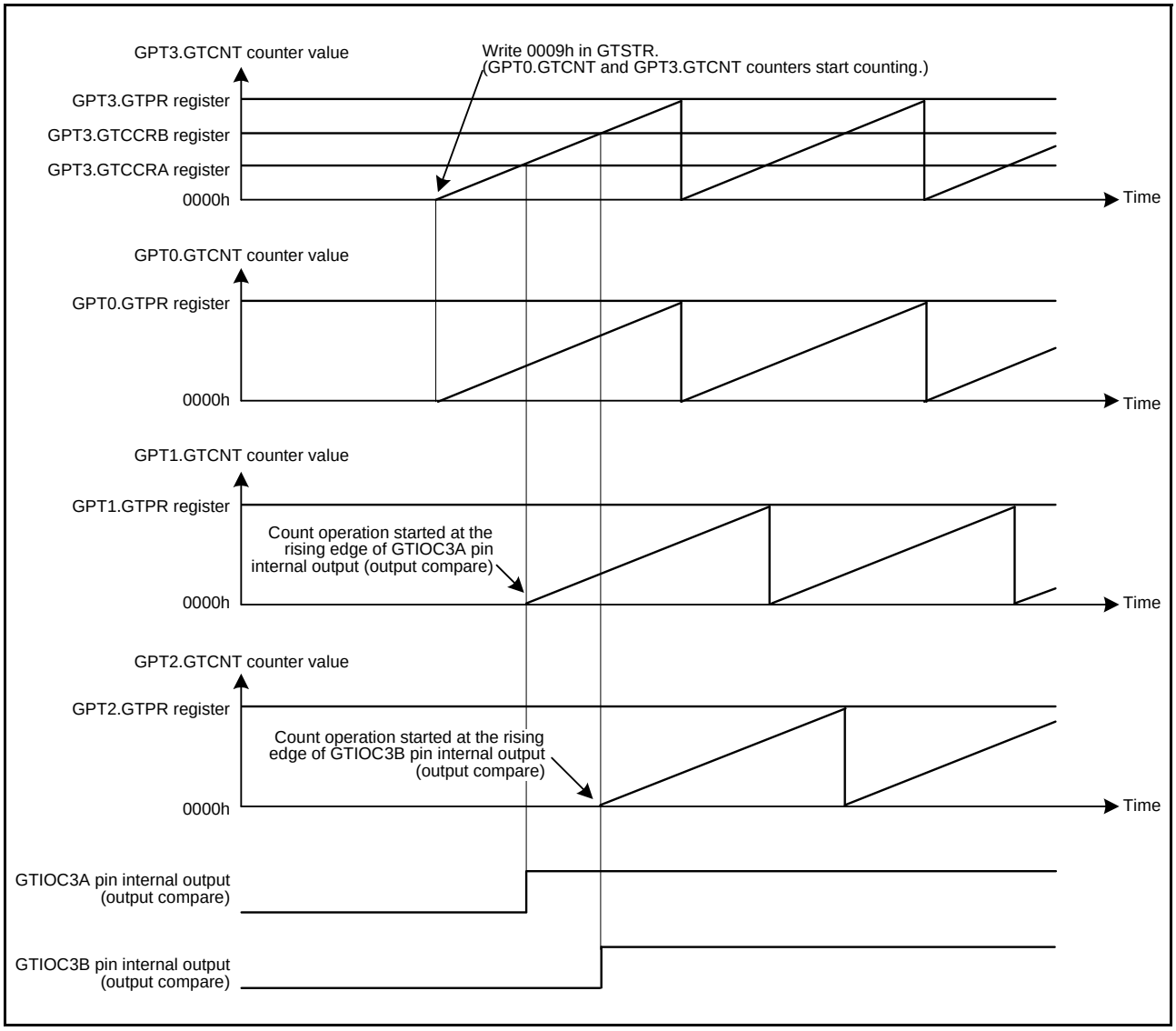


Figure 26.61 Example of Phase Shifting Start Operation by Hardware Source (with Same Count Period (GTPR Value))

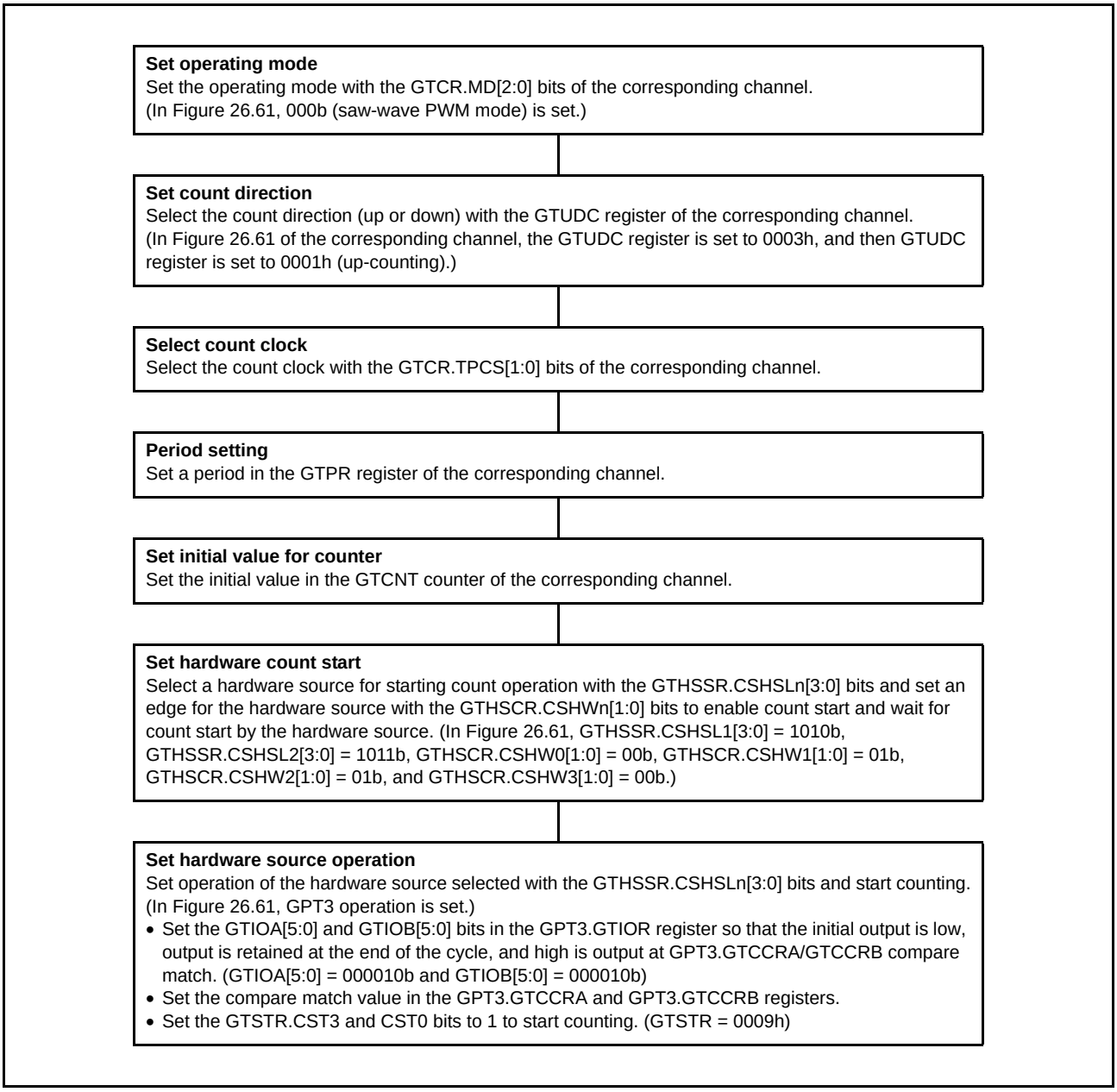


Figure 26.62 Example for Setting Phase Start by Hardware Source

26.3.8 PWM Output Operation Examples

(1) Synchronous PWM Output

The GPT can output eight phases of linked PWM waveforms for a maximum of four channels by synchronizing operation of the channels.

Figure 26.63 shows an example in which all the channels perform synchronous operation in saw-wave PWM mode and eight phases of PWM waveforms are output. The GTIOCnA pin is set so that it will output low as the initial output, high at a GTCCRA compare match, and low at the end of the cycle. The GTIOCnB pin is set so that it will output low as the initial output, high at a GTCCRB compare match, and low at the end of the cycle (n = 0 to 3).

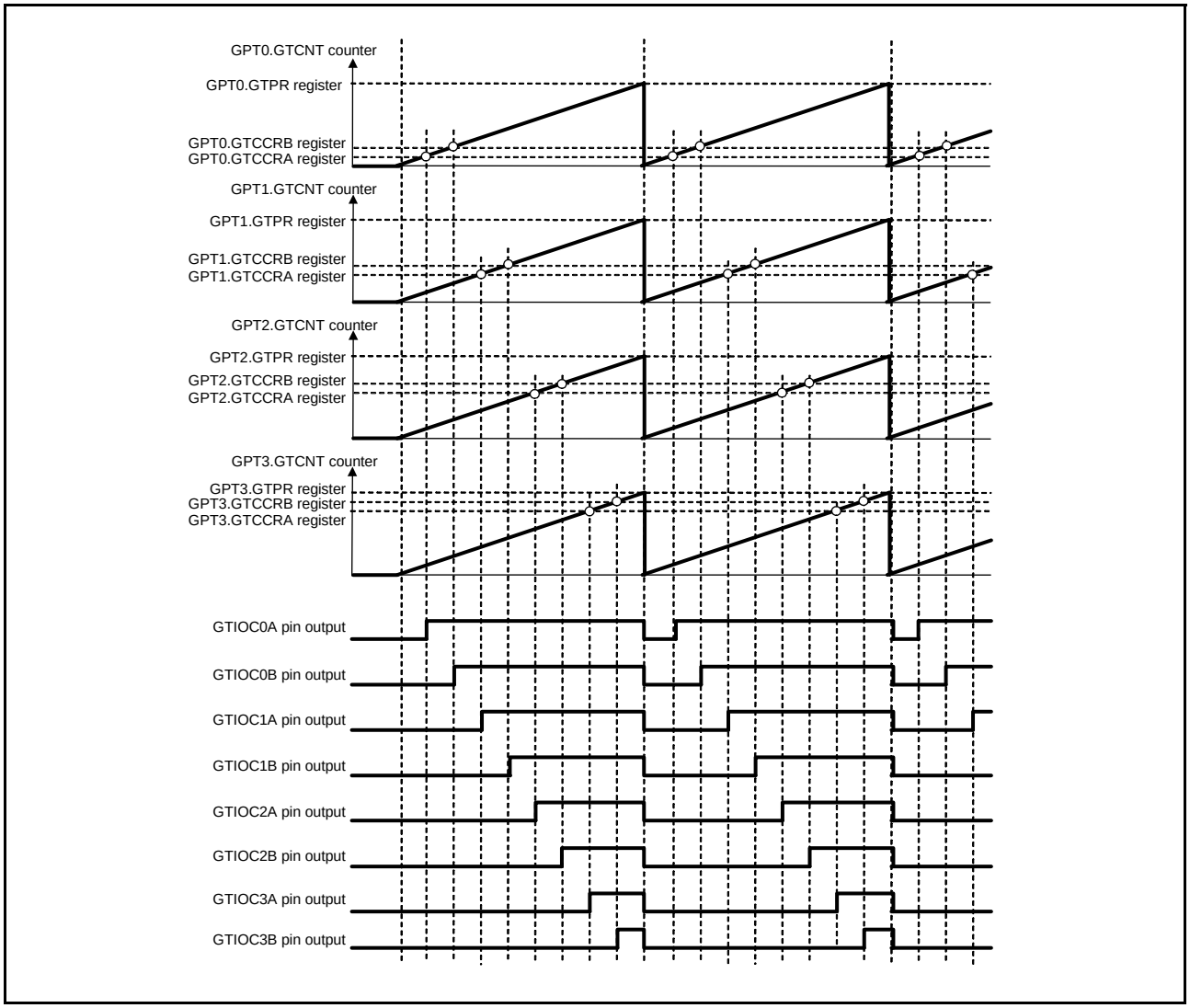


Figure 26.63 Example of Synchronous PWM Output

(2) Three-Phase Saw-Wave Complementary PWM Output

Figure 26.64 shows an example in which three channels perform synchronous operation in saw-wave PWM mode and three-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it will output low as the initial output, high at a GTCCRA compare match, and low at the end of the cycle. The GTIOCnB pin is set so that it will output high as the initial output, low at a GTCCRB compare match, and high at the end of the cycle.

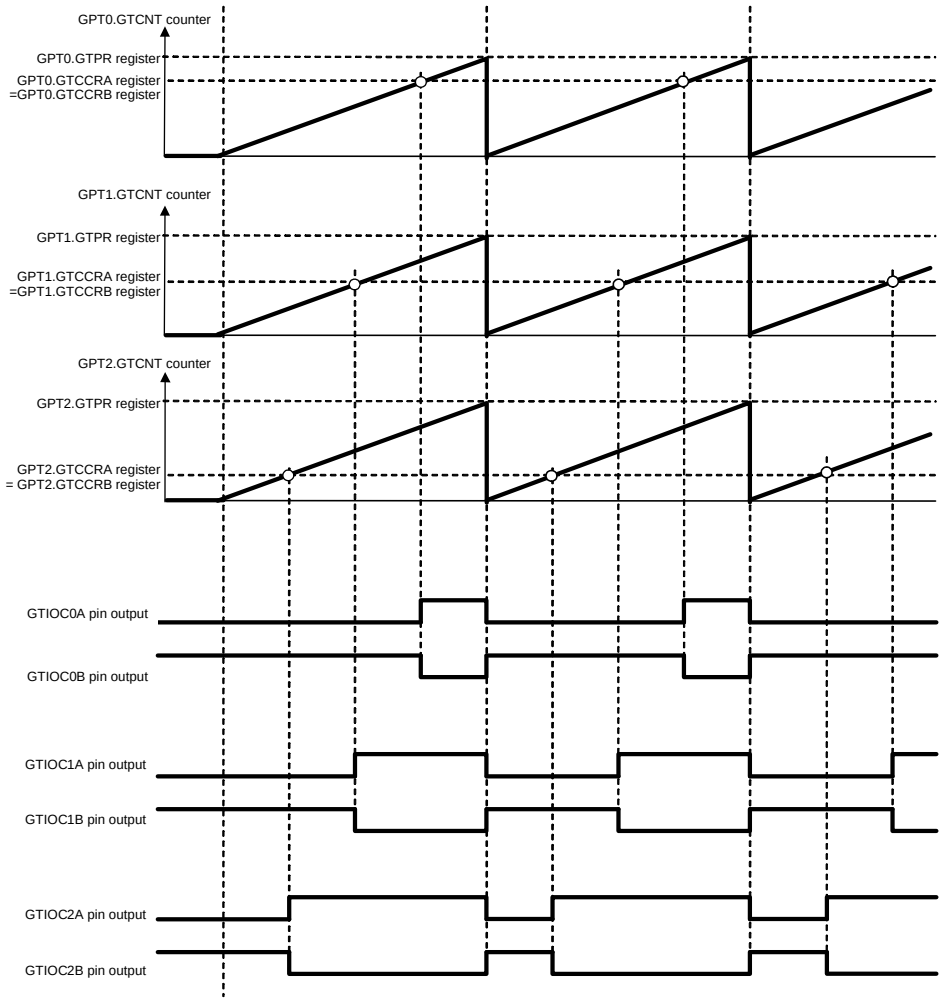


Figure 26.64 Example of Three-Phase Saw-Wave Complementary PWM Output

(3) Three-Phase Saw-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 26.65 shows an example in which three channels perform synchronous operation in saw-wave one-shot pulse mode with automatic dead time setting and three-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it will output low as the initial output, toggle the output at a GTCCRA compare match, and retain the output at the end of the cycle. The GTIOCnB pin is set so that it will output high as the initial output, toggle the output at a GTCCRB compare match, and retain the output at the end of the cycle.

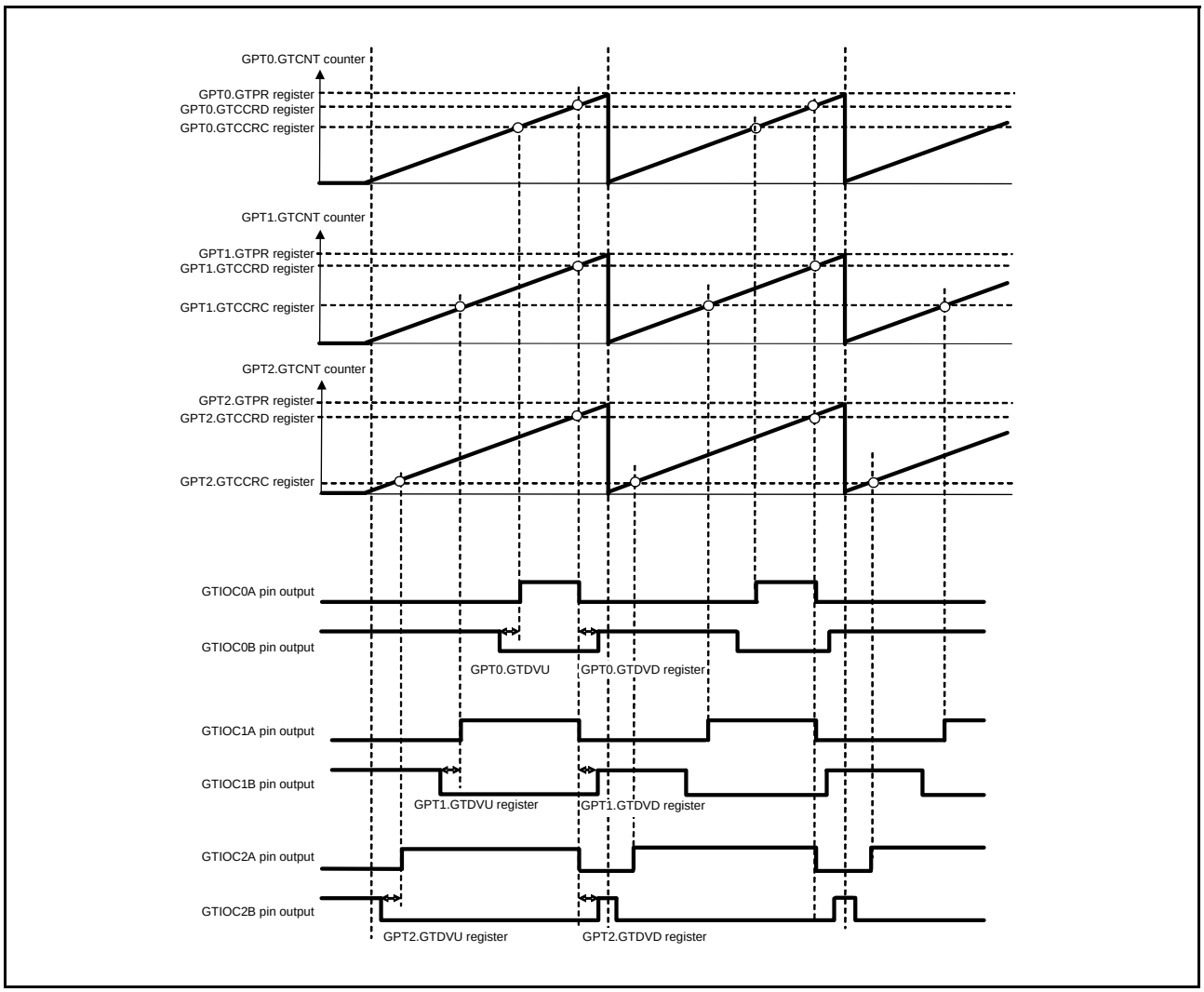


Figure 26.65 Example of Three-Phase Saw-Wave Complementary PWM Output with Automatic Dead Time Setting

(4) Three-Phase Triangle-Wave Complementary PWM Output

Figure 26.66 shows an example in which three channels perform synchronous operation in triangle-wave PWM mode 1 and three-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it will output low as the initial output, toggle the output at a GTCCRA compare match, and retain the output at the end of the cycle. The GTIOCnB pin is set so that it will output high as the initial output, toggle the output at a GTCCRB compare match, and retain the output at the end of the cycle.

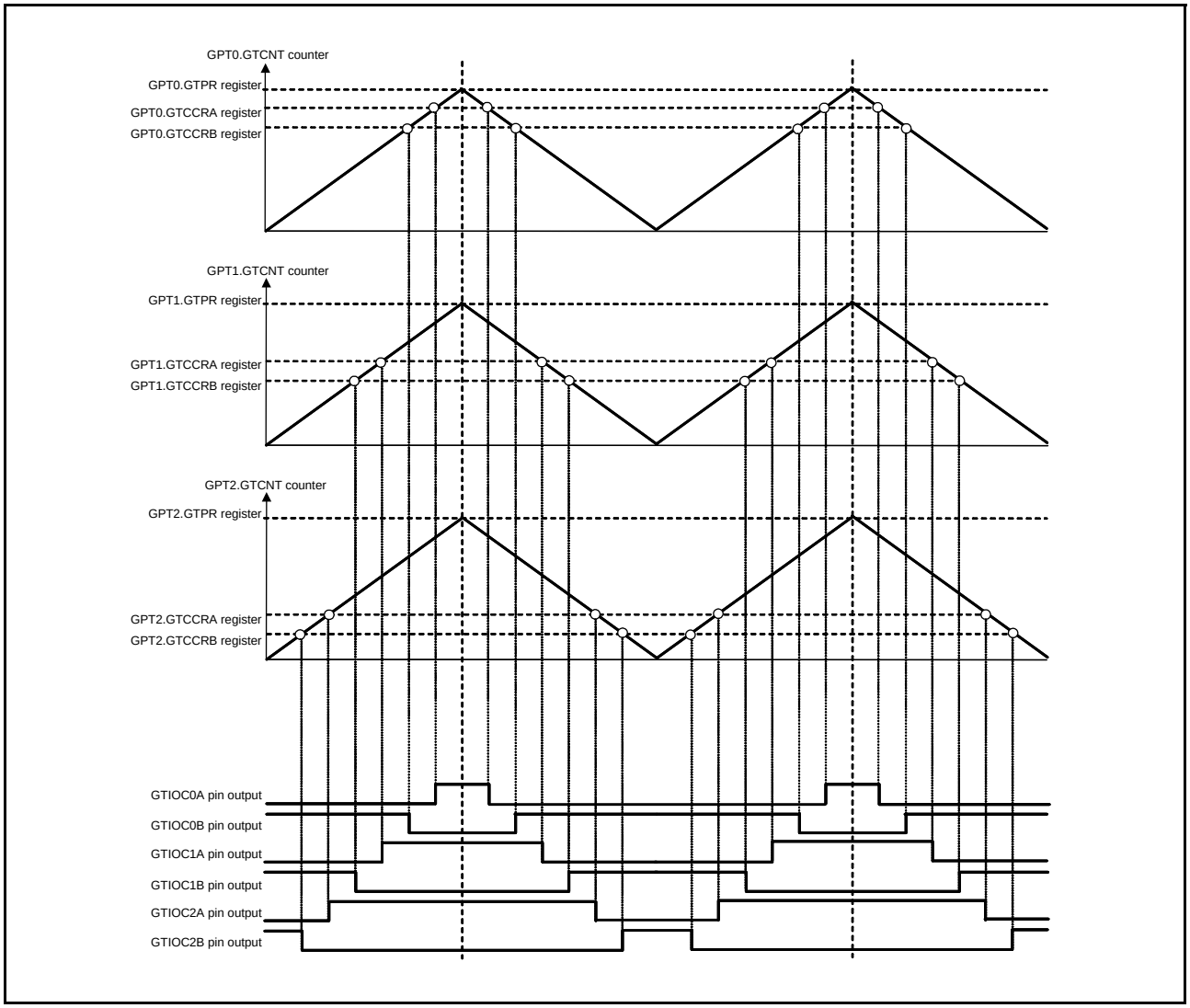


Figure 26.66 Example of Three-Phase Triangle-Wave Complementary PWM Output

(5) Three-Phase Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 26.67 shows an example in which three channels perform synchronous operation in triangle-wave PWM mode 1 with automatic dead time setting and three-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it will output low as the initial output, toggle the output at a GTCCRA compare match, and retain the output at the end of the cycle. The GTIOCnB pin is set so that it will output high as the initial output, toggle the output at a GTCCRB compare match, and retain the output at the end of the cycle.

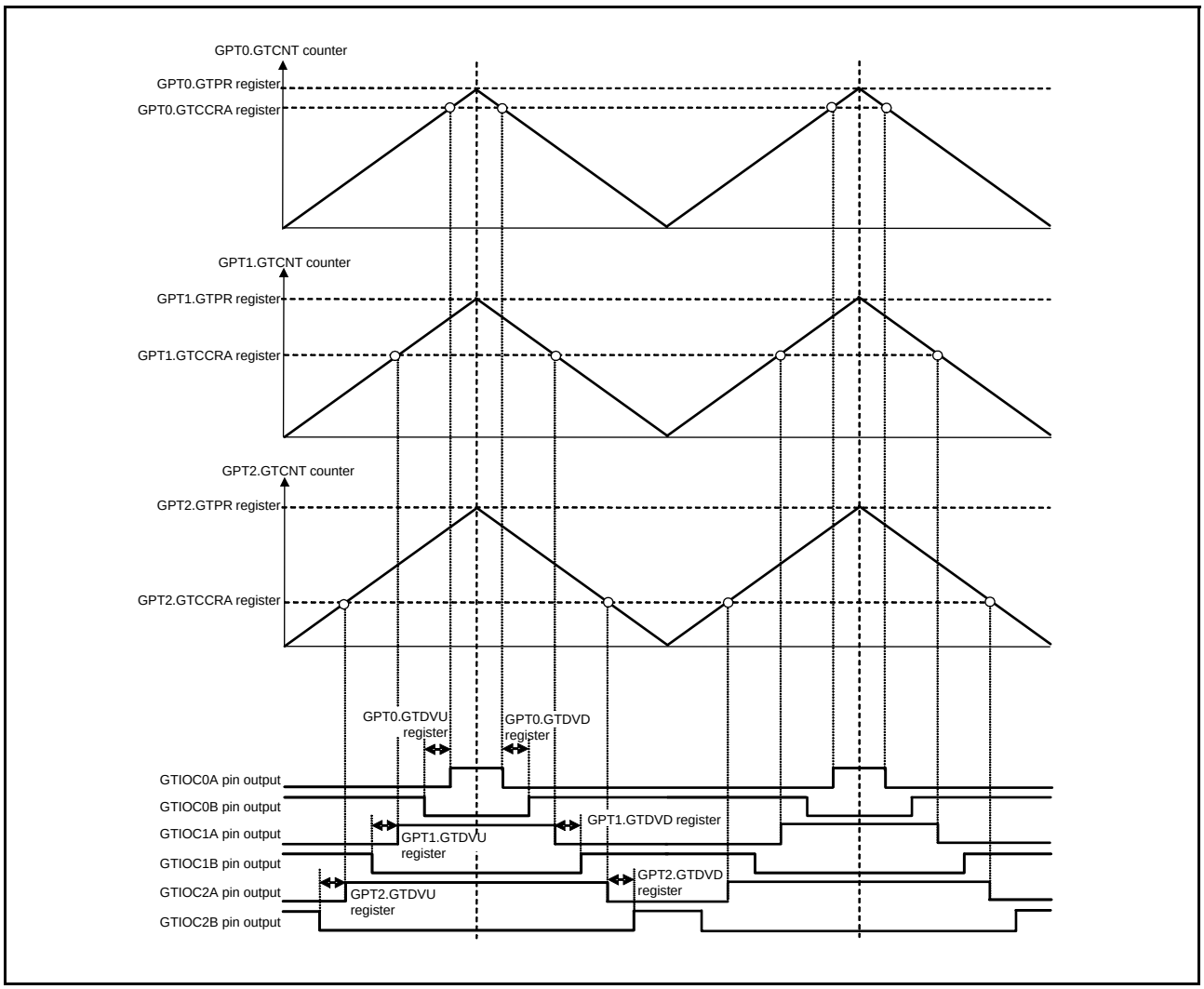


Figure 26.67 Example of Three-Phase Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

(6) Three-Phase Asymmetric Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 26.68 shows an example in which three channels perform synchronous operation in triangle-wave PWM mode 3 with automatic dead time setting and three-phase complementary PWM waveforms are output. The GTIOCnA is set so that it will output low as the initial output, toggle the output at a GTCCRA compare match, and retain the output at the end of the cycle. The GTIOCnB is set so that it will output high as the initial output, toggle the output at a GTCCRB compare match, and retain the output at the end of the cycle.

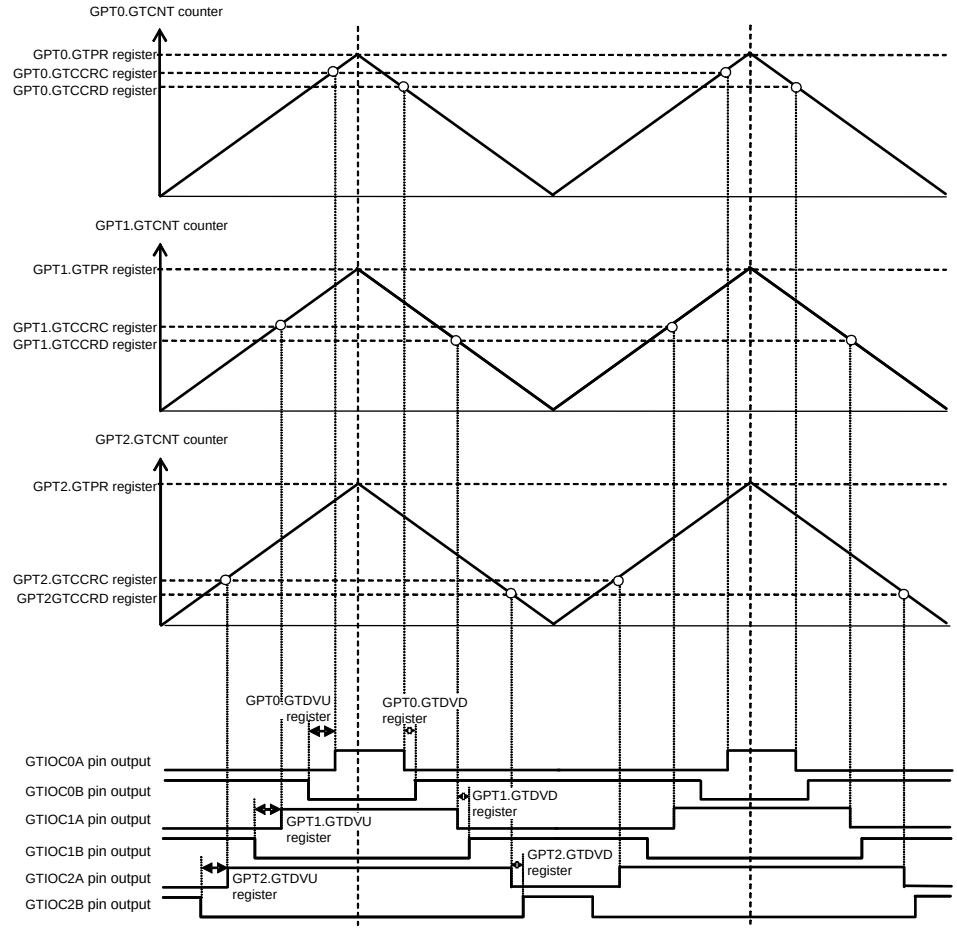


Figure 26.68 Example of Three-Phase Asymmetric Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

26.3.9 Noise Filter Function

Each pin for use in input capture and external trigger input to the GPT is equipped with a noise filter. The noise filter samples input signals at the sampling clock and removes the pulses of which length is less than three sampling periods. The noise filter functionality includes enabling and disabling of the noise filter for each pin and setting of the sampling clock for each channel. The NFCR register can be used to set the GTIOCnA and GTIOCnB pins. The GTETINT register can be used to set the GTETRG pin.

Figure 26.69 shows the timing of noise filtering.

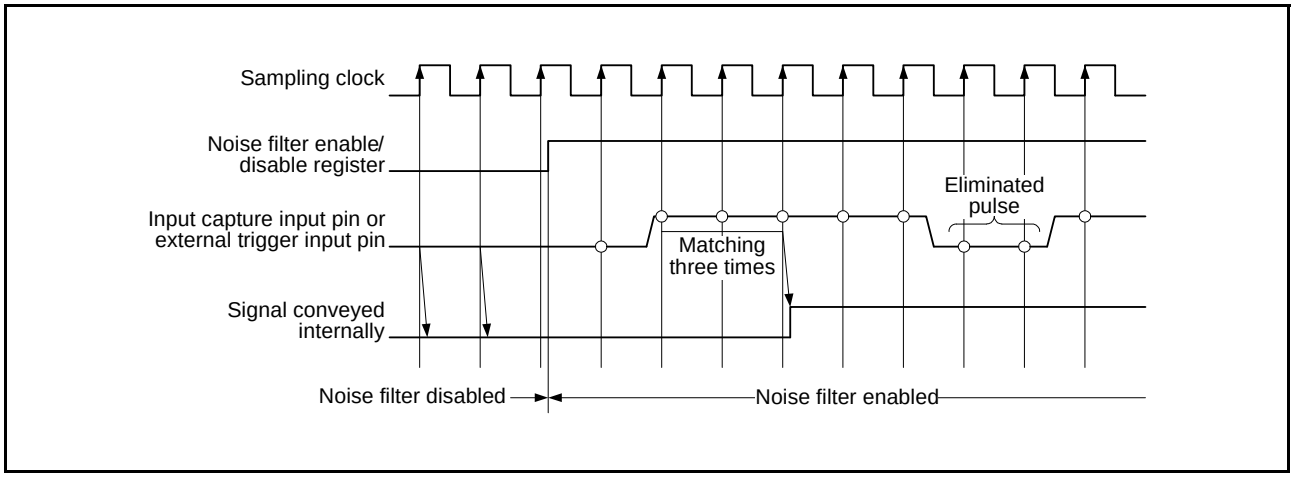


Figure 26.69 Timing of Noise Filtering

If noise filtering is enabled, input capture operation or external trigger operation is performed on the edges of noise-filtered signal after a delay of (minimum sampling interval × 2 + PCLKA) due to noise filtering for the input capture input or external trigger operation.

26.4 Interrupt Sources

26.4.1 Interrupt Sources and Priorities

Table 26.5 lists the interrupt sources.

Each interrupt source has a control bit for the respective dedicated interrupt request generation and enables or disables the generation of interrupt request independently.

The priority of channels can be changed by the interrupt controller settings. For details, see section 15, Interrupt Controller (ICUA).

Table 26.5 GPT Interrupt Sources

Channel	Name	Interrupt Source	DMAC/DTC Activation
Common	ETGIN	External trigger falling input	Possible
	ETGIP	External trigger rising input	Possible
GPT0	GTCIA0	GPT0.GTCCRA input capture/compare match	Possible
	GTCIB0	GPT0.GTCCRB input capture/compare match	Possible
	GTCIC0	GPT0.GTCCRC compare match	Possible
	GTCID0	GPT0.GTCCRD compare match	Possible
	GDTE0	Dead time error	Possible
	GTCIE0	GPT0.GTCCRE compare match	Possible
	GTCIF0	GPT0.GTCCRF compare match	Possible
	GTCIV0	GPT0.GTCNT overflow (GPT0.GTPR compare match)	Possible
	GTCIU0	GPT0.GTCNT underflow	Possible
GPT1	GTCIA1	GPT1.GTCCRA input capture/compare match	Possible
	GTCIB1	GPT1.GTCCRB input capture/compare match	Possible
	GTCIC1	GPT1.GTCCRC compare match	Possible
	GTCID1	GPT1.GTCCRD compare match	Possible
	GDTE1	Dead time error	Possible
	GTCIE1	GPT1.GTCCRE compare match	Possible
	GTCIF1	GPT1.GTCCRF compare match	Possible
	GTCIV1	GPT1.GTCNT overflow (GPT1.GTPR compare match)	Possible
	GTCIU1	GPT1.GTCNT underflow	Possible
GPT2	GTCIA2	GPT2.GTCCRA input capture/compare match	Possible
	GTCIB2	GPT2.GTCCRB input capture/compare match	Possible
	GTCIC2	GPT2.GTCCRC compare match	Possible
	GTCID2	GPT2.GTCCRD compare match	Possible
	GDTE2	Dead time error	Possible
	GTCIE2	GPT2.GTCCRE compare match	Possible
	GTCIF2	GPT2.GTCCRF compare match	Possible
	GTCIV2	GPT2.GTCNT overflow (GPT2.GTPR compare match)	Possible
	GTCIU2	GPT2.GTCNT underflow	Possible
GPT3	GTCIA3	GPT3.GTCCRA input capture/compare match	Possible
	GTCIB3	GPT3.GTCCRB input capture/compare match	Possible
	GTCIC3	GPT3.GTCCRC compare match	Possible
	GTCID3	GPT3.GTCCRD compare match	Possible
	GDTE3	Dead time error	Possible
	GTCIE3	GPT3.GTCCRE compare match	Possible
	GTCIF3	GPT3.GTCCRF compare match	Possible
	GTCIV3	GPT3.GTCNT overflow (GPT3.GTPR compare match)	Possible
	GTCIU3	GPT3.GTCNT underflow	Possible

(1) GTCIAn interrupt (n = 0 to 3)

When the GTINTAD.GTINTA bit is 1, an interrupt request is generated under the following conditions.

- When the GTCCRA register functions as a compare match register, the GTCNT counter value matches with the GTCCRA register.
- When the GTCCRA register functions as an input capture register, the input-capture signal has caused transfer of the GTCNT counter value to the GTCCRA register.

(2) GTCIBn interrupt (n = 0 to 3)

When the GTINTAD.GTINTB bit is 1, an interrupt request is generated under the following conditions.

- When the GTCCRB register functions as a compare match register, the GTCNT counter value matches with the GTCCRB register.
- When the GTCCRB register functions as an input capture register, the input-capture signal has caused transfer of the GTCNT counter value to the GTCCRB register.

(3) GTCICn interrupt (n = 0 to 3)

When the GTINTAD.GTINTC bit is 1, an interrupt request is generated under the following condition.

- When the GTCCRC register functions as a compare match register, the GTCNT counter value matches with the GTCCRC register.

A compare match is not performed and thus interrupt is not requested under the following conditions.

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 01b, 10b, 11b (buffer operation with the GTCCRC register)

(4) GTCIDn interrupt (n = 0 to 3)

When the GTINTAD.GTINTD bit is 1, an interrupt request is generated under the following condition.

- When the GTCCRD register functions as a compare match register, the GTCNT counter value matches with the GTCCRD register.

A compare match is not performed and thus interrupt is not requested under the following conditions.

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 10b, 11b (buffer operation with the GTCCRD register)

(5) GTCIEn interrupt (n = 0 to 3)

When the GTINTAD.GTINTE bit is 1, an interrupt request is generated under the following condition.

- When the GTCCRE register functions as a compare match register, the GTCNT counter value matches with the GTCCRE register.

A compare match is not performed and thus interrupt is not requested under the following conditions.

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (buffer operation with the GTCCRE register)

(6) GTCIFn interrupt (n = 0 to 3)

When the GTINTAD.GTINTF bit is 1, an interrupt request is generated under the following condition.

- When the GTCCRF register functions as a compare match register, the GTCNT counter value matches with the GTCCRF register.

A compare match is not performed and thus interrupt is not requested under the following conditions.

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 10b, 11b (buffer operation with the GTCCRF register)

(7) GTCIVn interrupt (n = 0 to 3)

When the GTINTAD.GTINTPR[0] bit is 1, an interrupt request is generated under the following conditions.

- In saw-wave mode, interrupt requests are enabled at overflows (GTCNT counter value changes from GTPR register value to 0000h during up-counting).
- In triangle-wave mode, interrupt requests are enabled at crests (GTCNT counter value changes from GTPR register value to GTPR register value minus 1).

(8) GTCIUn interrupt (n = 0 to 3)

When the GTINTAD.GTINTPR[1] bit is 1, an interrupt request is generated under the following conditions.

- In saw-wave mode, interrupt requests are enabled at underflows (GTCNT counter value changes from 0000h to GTPR register value during down-counting).
- In triangle-wave mode, interrupt requests are enabled at troughs (GTCNT counter value changes from 0000h to 0001h).

(9) ETGIP interrupt

When the GTETINT.ETIPEN bit is 1, an interrupt request is generated under the following condition.

- When the rising edge of an external trigger input is detected

(10) ETGIN interrupt

When the GTETINT.ETINEN bit is 1, an interrupt request is generated under the following condition.

- When the falling edge of an external trigger input is detected

(11) GDTEn interrupt (n = 0 to 3)

When automatic dead time setting has been made, the GTST.DTEF flag becomes 1 when the timer output toggle point with dead time added exceeds the count period. If GTINTAD.EINT is 1 at this time, a dead time error interrupt request (GDTE) is generated.
In addition, when the timer output toggle point with dead time added is back within the count period, the GTST.DTEF flag changes from 1 to 0.

26.4.2 DMAC/DTC Activation

The DMAC and DTC can be triggered by the interrupt request in each channel. For details, see section 15, Interrupt Controller (ICUA), section 18, DMA Controller (DMACAa), and section 20, Data Transfer Controller (DTCa).

26.4.3 Interrupt and A/D Conversion Request Skipping Function

By setting the GTITC register, the GTCNT counter overflow (GTPR compare match) interrupt (GTCIV) and underflow interrupt (GTCIU) can be skipped. Other interrupts and A/D converter start request signals can be skipped in coordination with the GTCIV/GTCIU skipping function. However, the dead time error interrupts cannot be linked with the GTCIV/GTCIU skipping function.

When both troughs and crests are counted and skipped in triangle-wave mode, if the number of times of skipping is odd, GTCIV/GTCIU interrupt requests cannot be generated at troughs only or at crests only depending on the skipping counter start timing. Therefore, in order to count both troughs and crests and generate the GTCIV/GTCIU interrupts at troughs only or crests only in triangle-wave mode, the number of times of skipping should be even.

Similarly, in saw-wave mode, when both overflows and underflows are counted and skipped with the count direction changed, GTCIV/GTCIU interrupt requests cannot be generated at overflows only or at underflows only. Therefore, in order to count both overflows and underflows with the count direction changed and generate the GTCIV/GTCIU interrupts at overflows only or underflows only in saw wave mode, the skipping state should be carefully checked before using.

When changing the skipping count, be sure to release the skipping count setting (GTITC.IVTC[1:0] bits = 00b).

Figure 26.70 to Figure 26.75 show examples of skipping function operation.

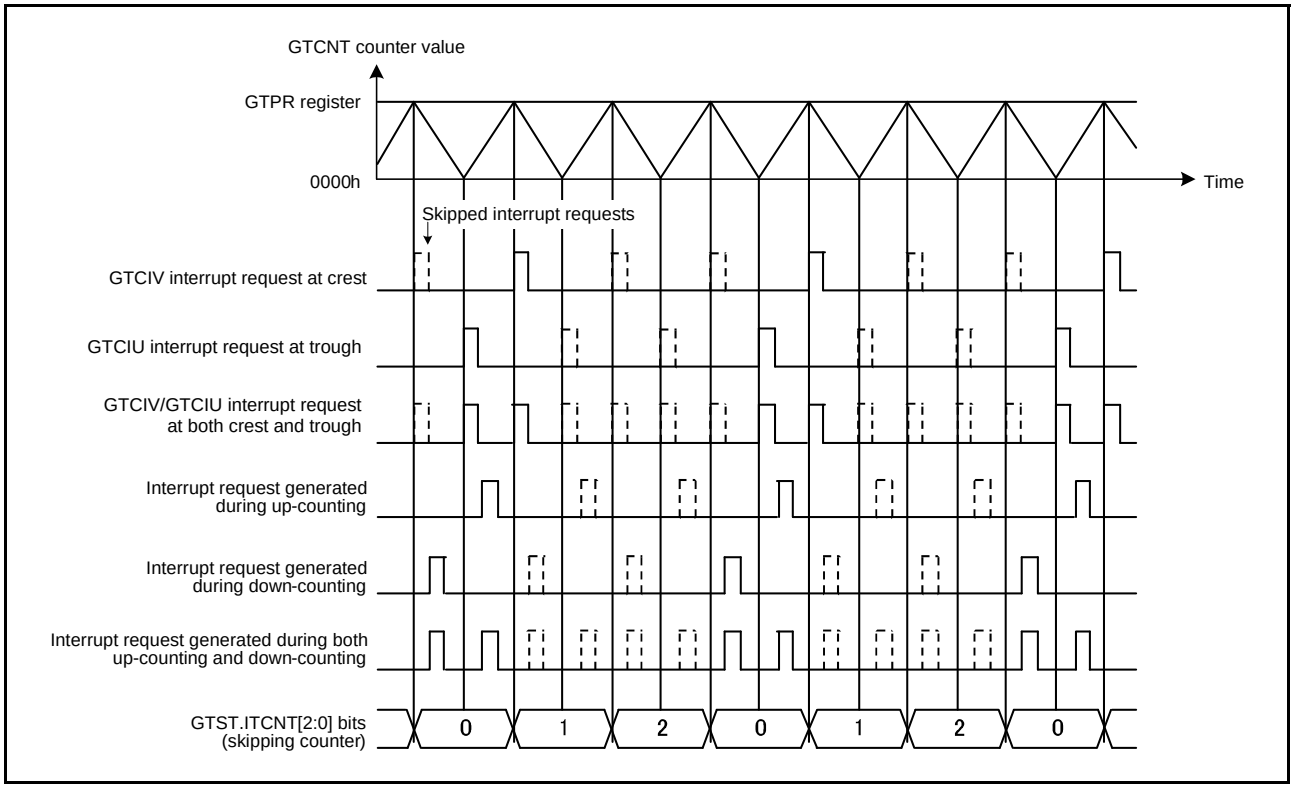


Figure 26.70 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Crests, Skipping Count: 2)

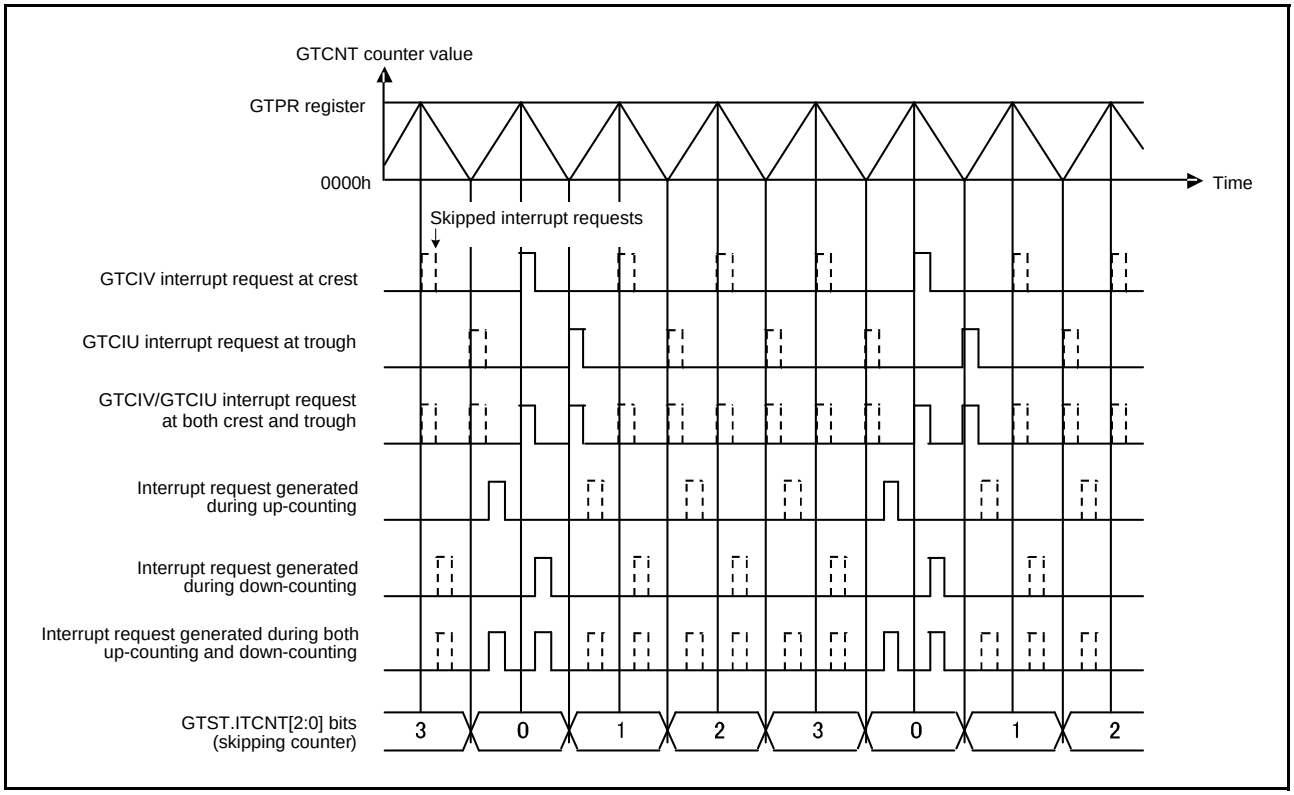


Figure 26.71 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Troughs, Skipping Count: 3)

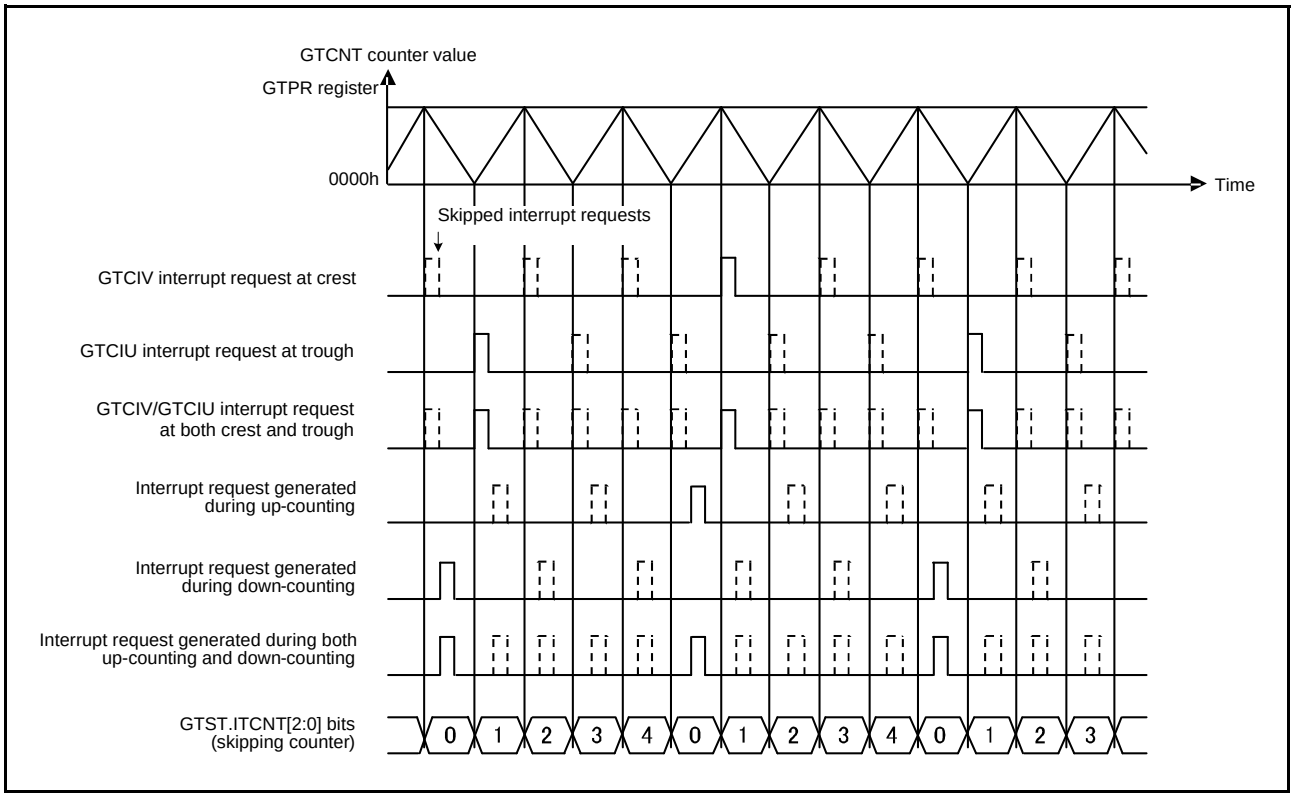


Figure 26.72 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 4)

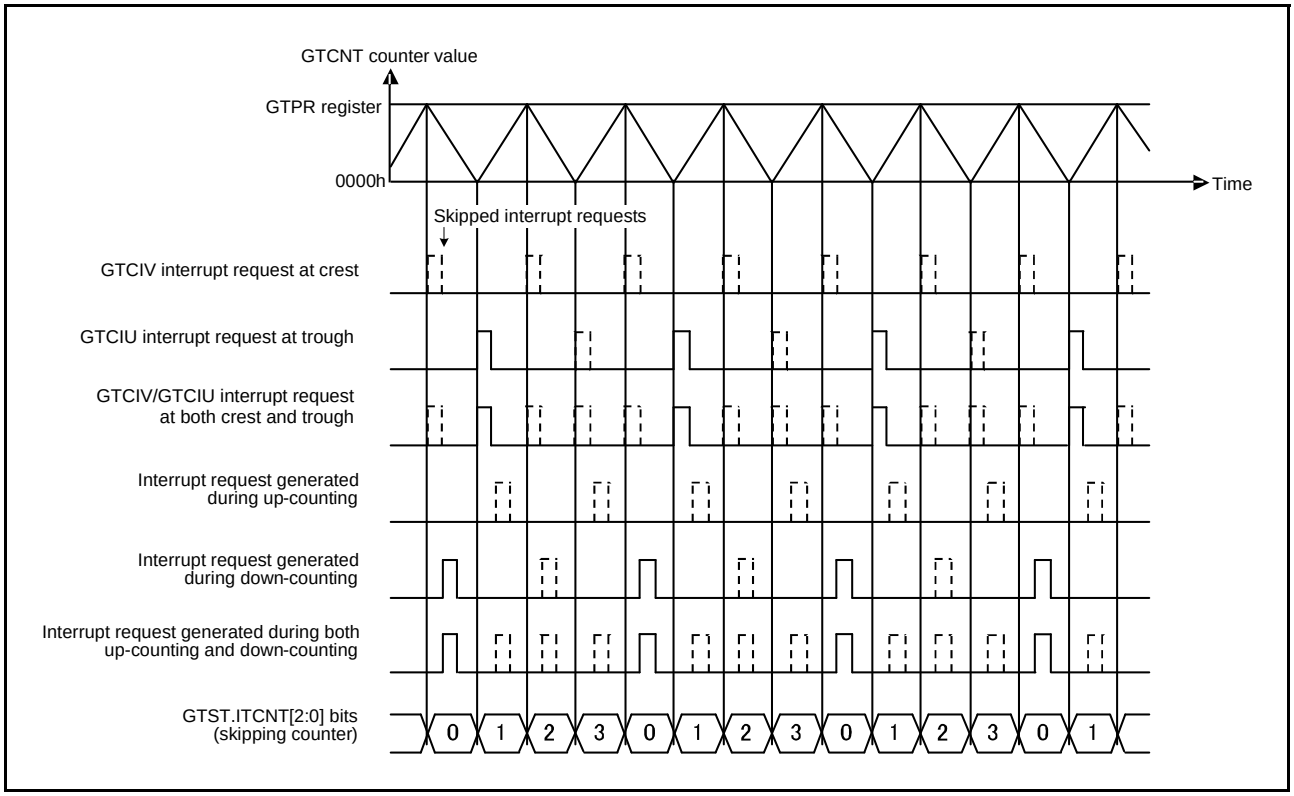


Figure 26.73 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 3, Skipping Started at Up-Counting)

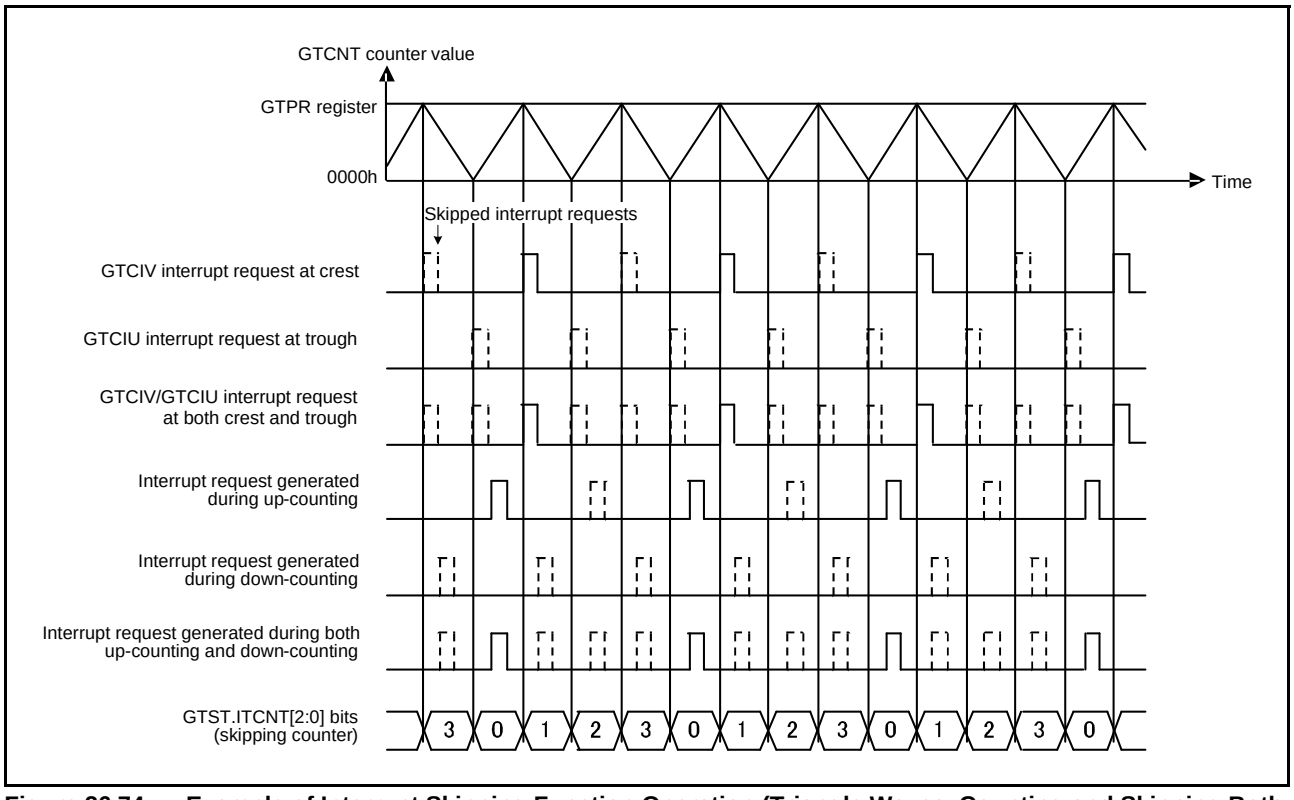


Figure 26.74 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 3, Skipping Started at Down-Counting)

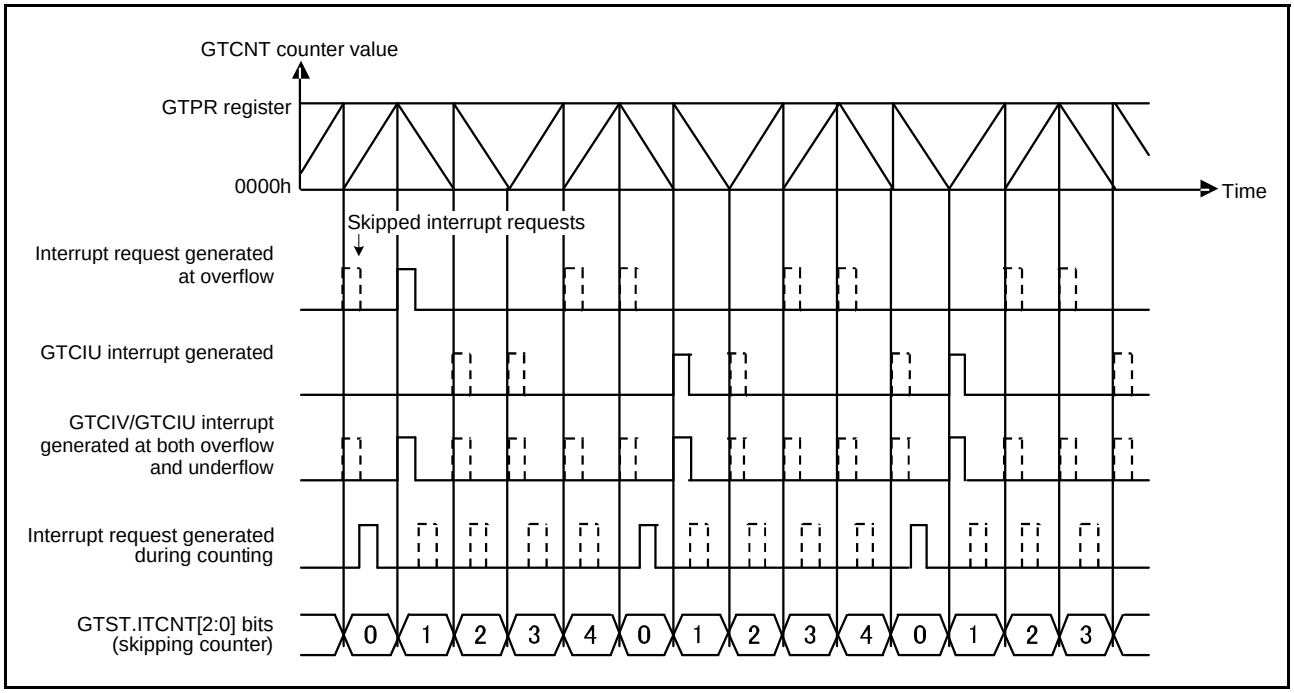


Figure 26.75 Example of Interrupt Skipping Function Operation (Saw Waves, Operation with Count Direction Changed, Counting and Skipping Both Overflows and Underflows, Skipping Count: 4)

26.5 A/D Converter Start Request

An A/D converter start request can be issued at a compare match between the GTCNT counter and the GTADTRA or GTADTRB register, up-counting only, down-counting only, or both up-counting and down-counting can be specified by setting the GTINTAD register.

The GTADTRA and GTADTRB registers each has two buffer registers. Buffer operation with the GTADTRA register used together with the GTADTBRA and GTADTDBRA registers, and buffer operation with the GTADTRB register used together with the GTADTBRB and GTADTDBRB registers can be performed.

Figure 26.76 shows an example of A/D converter start request operation, and Figure 26.77 shows an example for setting A/D converter start request operation.

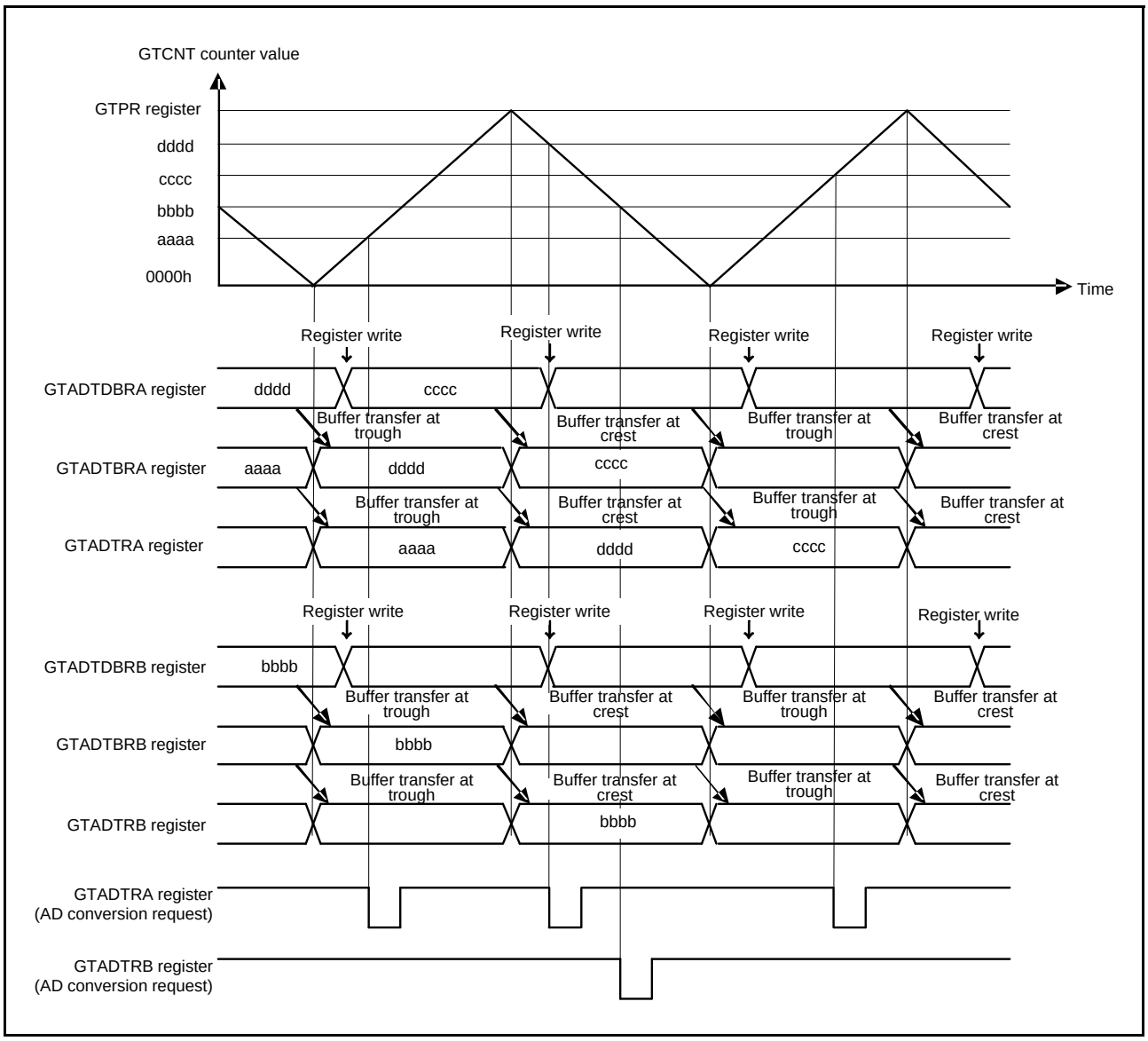


Figure 26.76 Example of A/D Converter Start Request Timing Operation (Triangle Waves, Double Buffer Operation, Buffer Transfer at Both Troughs and Crests, A/D Converter Start Requested by GTADTRA0 at Both Up-Counting and Down-Counting, A/D Converter Start Requested by GTADTRB0 at Down-Counting)

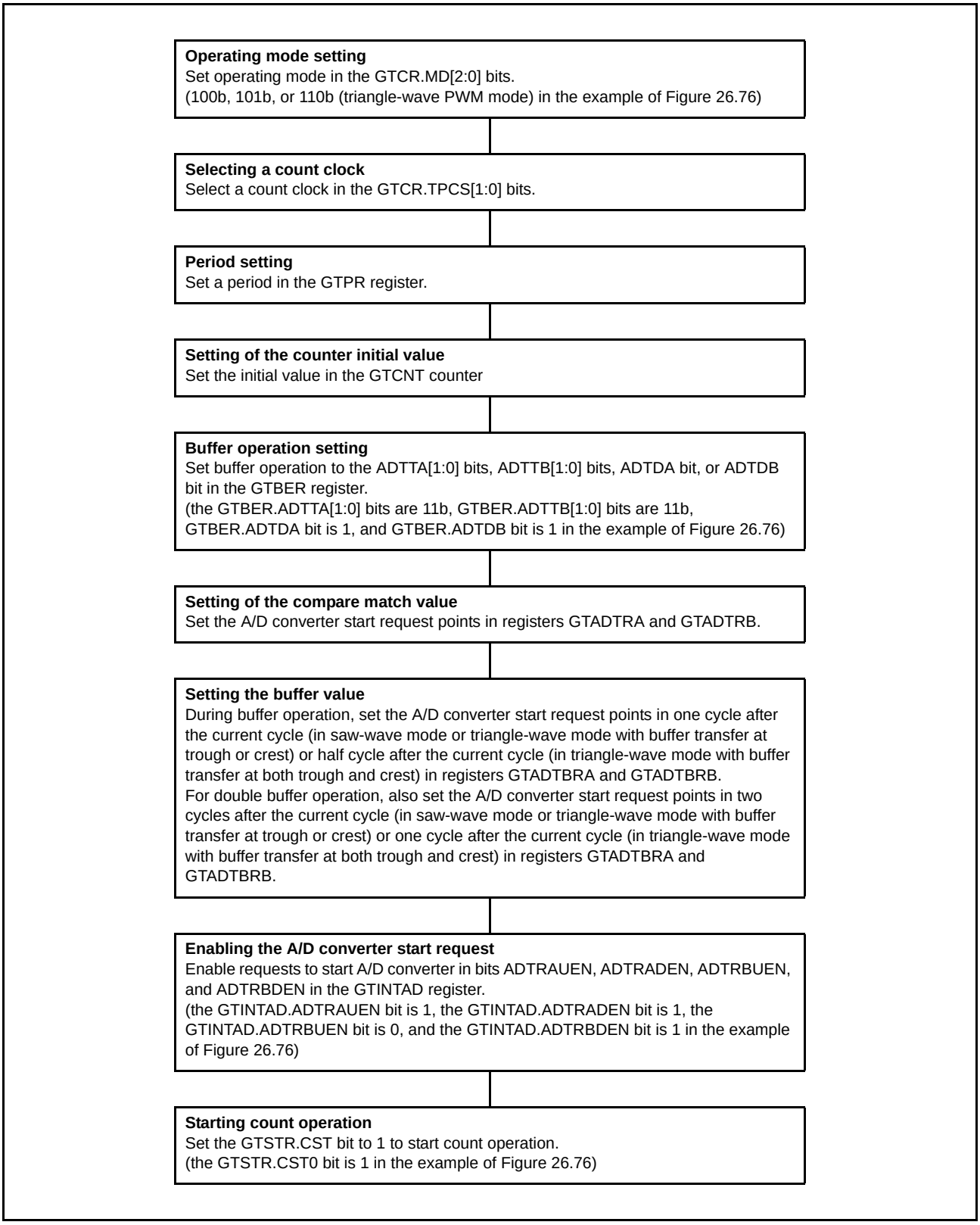


Figure 26.77 Example for Setting A/D Converter Start Request Timing Operation

26.6 Operations Linked by the ELC

26.6.1 Event Signal Output to the ELC

GPT is capable of operation linked with another module set in advance when its interrupt request signal is used as an event signal by the event link controller (ELC).
GPT outputs the following event signal: compare match A, compare match B, compare match C, compare match D, overflow, and underflow.
The event signal can be output regardless of the settings of the corresponding interrupt request enabling bits.
For details, refer to section 21, Event Link Controller (ELC).

26.6.2 GPT Operations in Response to Receiving Event Signals from the ELC

The GPT can perform the following operations in response to the event set in advance in the ELSRn register of the event link controller (ELC).

(1) Start Counting

Counting by the GPT starts in response to the event when this is selected by the setting of the ELOPI or ELOPJ register of the ELC. The ELOPI register handles control for GPT0 and GPT1, and the ELOPJ register controls for GPT2 and GPT3. When the event specified in the ELSRn register occurs, the GTSTR.CSTn bit shown in Table 26.6 is set to 1, and the GPT counter starts.
However, when the specified event is generated while the GTSTR.CSTn bit has already been set to 1, the event has no effect. Table 26.6 lists the GTSTR register bits used for each channel.

Table 26.6 General PWM Timer Software Start Registers for Linked Operation with the ELC

Channel No.	GTSTR register (timer start register)
GPT0	GTSTR.CST0 bit
GPT1	GTSTR.CST1 bit
GPT2	GTSTR.CST2 bit
GPT3	GTSTR.CST3 bit

(2) Input Capture Operation

Input capture by the GPT proceeds when this is selected by the setting of the ELOPI or ELOPJ register of the ELC. The ELOPI register handles control for GPT0 and GPT1, and the ELOPJ register controls for GPT2 and GPT3. When the event specified in the ELSRn register occurs, the GTCCRm register (general PWM timer compare capture register) captures the value of the GTCNT counter (timer counter register). When using input capture in response to an event, the corresponding bit of the GTIOR register (general PWM timer I/O control register) in the GPT should be set for input capture and the GTSTR.CSTn bit should be set to 1 to start counting by the counter.
In this case, the GTIOCnA pin (input capture pin) input has no effect.
Table 26.7 lists the GTCCRm and GTIOR registers used for each channel in input capture operation in response to the ELC.

Table 26.7 Registers Used in the Input Capture Operation by ELC

Channel No.	Timer General Register Name	Timer I/O Control Register
GPT0	GPT0.GTCCRA register	GPT0.GTIOR.GTIOA[5:0] bit
GPT1	GPT1.GTCCRA register	GPT1.GTIOR.GTIOA[5:0] bit
GPT2	GPT2.GTCCRA register	GPT2.GTIOR.GTIOA[5:0] bit
GPT3	GPT3.GTCCRA register	GPT3.GTIOR.GTIOA[5:0] bit

(3) Restart Counting

The counter is cleared in response to the event when the ELOPI or ELOPJ register of the ELC is set for “restart counting” as the operation for the GPT. The ELOPI register handles control for GPT0 and GPT1, and the ELOPJ register controls for GPT2 and GPT3. When the event specified in the ELSRn register occurs, the GTCNT counter (timer counter register) is returned to its initial value. If the corresponding GTSTR.CSTn bit is set to 1, counting will continue. For the GTSTR.CSTn bit, see Table 26.6.

(4) Stop Counting

The counter is cleared in response to the event when the ELOPI or ELOPJ register of the ELC is set for “stop counting” as the operation for the GPT. The ELOPI register handles control for GPT0 and GPT1, and the ELOPJ register controls for GPT2 and GPT3. When the event specified in the ELSRn register occurs, the corresponding GTSTR.CSTn bit is set to 0 and the GPT counter stops.
If an event occurs while the GTSTR.CSTn bit is 0, the event is ignored.

26.6.3 Usage Notes on GPT Operation by Event Signal Reception from ELC

The following notes on usage apply when the GPT is used in event link operation.

(1) Start Counting

If the event specified in the ELSRn register occurs during a cycle of writing to a GTSTR.CSTn bit, writing to the GTSTR.CSTn bit takes priority.

(2) Stop Counting

If the event specified in the ELSRn register occurs during a cycle of writing to the GTCNT counter (timer counter register), writing to the GTCNT counter takes priority.

(3) Restart Counting

When the counter is restarted, the compare match function is disabled.

26.7 Protection Function

26.7.1 Write-Protection for Registers

Registers can be write-protected per channel by setting the GTWP.WPn bit (n = 0 to 3) to prevent the values of the registers of each channel from being accidentally modified.
The write-protection can be set for the following registers:

Table 26.8 List of Write-Protected Registers

Register Symbol	Register Name
GTIOR	General PWM timer I/O control register
GTINTAD	General PWM timer interrupt output register
GTCR	General PWM timer control register
GTBER	General PWM timer buffer enable register
GTUDC	General PWM timer count direction register
GTITC	General PWM timer interrupt and A/D converter start request skipping setting register
GTST	General PWM timer status register
GTCNT	General PWM timer counter
GTCCRA to GTCCRF	General PWM timer compare capture registers A to F
GTPR	General PWM timer period setting register
GTPBR	General PWM timer period setting buffer register
GTPDBR	General PWM timer period setting double-buffer register
GTADTRA, GTADTRB	A/D converter start request timing registers A and B
GTADTBRA, GTADTBRB	A/D converter start request timing buffer registers A and B
GTADTDBRA, GTADTDBRB	A/D converter start request timing double-buffer registers A and B
GTONCR	General PWM timer output negate control register
GTDTCR	General PWM timer dead time control register
GTDVU, GTDVD	General PWM timer dead time value registers U and D
GTDBU, GTDBD	General PWM timer dead time buffer registers U and D
GTSOTR	General PWM timer output protection function temporary release register

26.7.2 Disabling of Buffer Operation

If the timing of buffer register write is too late for the buffer transfer timing, buffer operation can be suspended with the GTBDR setting. Specifically, buffer transfer can be temporarily disabled, even though a buffer transfer condition is generated during buffer register write, by setting the corresponding GTBDR bit to 1 (buffer operation disabled) before buffer register write and setting the bit to 0 (buffer operation enabled) after completion of writing to all the buffer registers.
Figure 26.78 shows an example of operation for disabling buffer operation

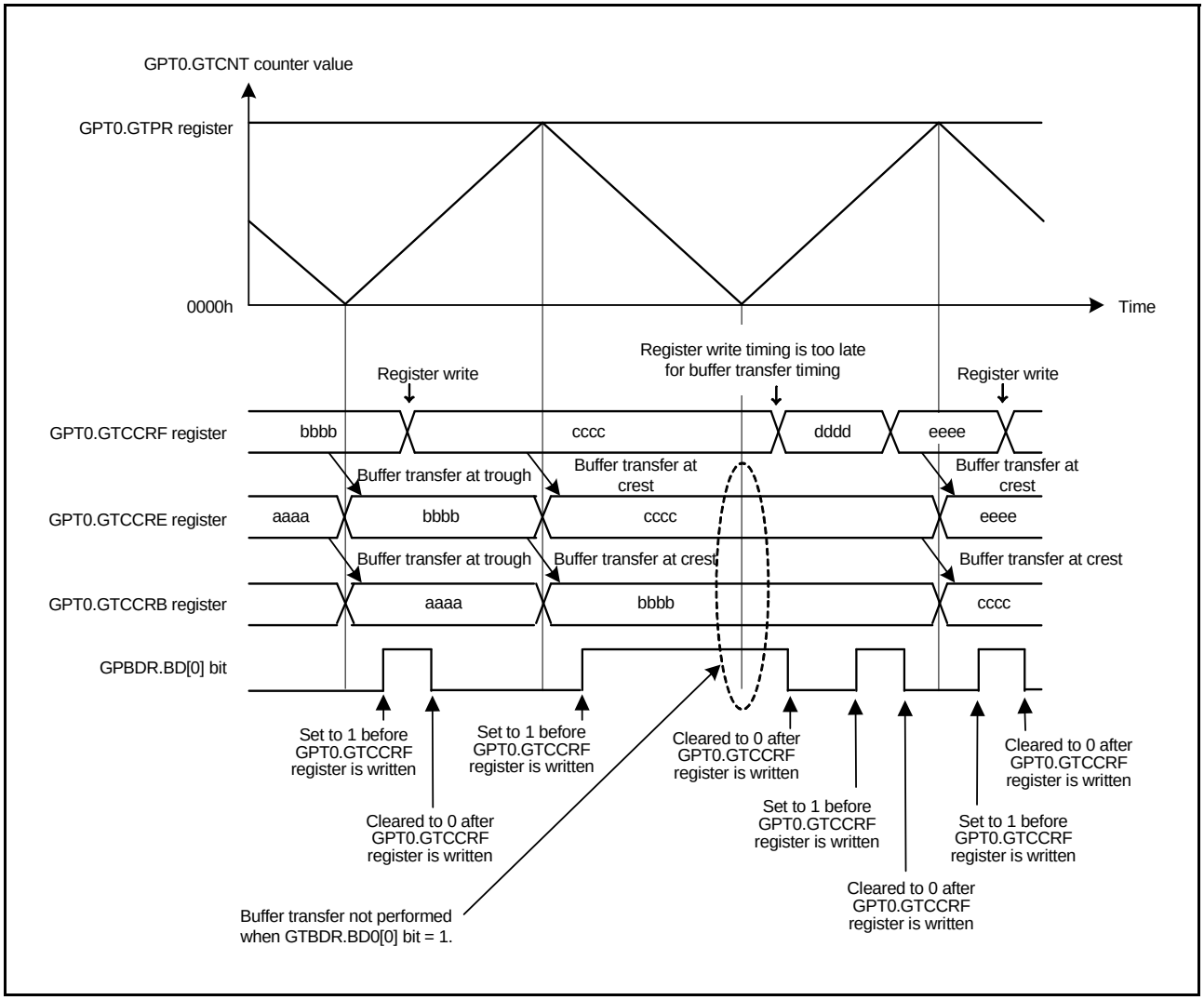


Figure 26.78 Example of Operation for Disabling Buffer Operation (Triangle Waves, Double Buffer Operation, Buffer Transfer at Both Troughs and Crests)

26.7.3 GTIOC Pin Output Negate Control

For protection from system failure, the negate control (deactivating the output level) is provided for GTIOC pin output. There are two negate control sources:GTETRG pin input, and writing to the GTONCR.SWN bit.

The negate control source can be selected by the GTONCR.NFS[3:0] bits, and the polarity of the negate source by the GTONCR.NFV bits. The inactive level to be output can be selected by the GTONCR.NVA and NAB bits.

When negate control is enabled by the GTONCR.NEA and NAB NEB bits, the inactive level selected by the GTONCR.NVA and NAB bits is output from the GTIOC pin.

Figure 26.79 shows an example of the GTIOC pin output negate control operation (when the GTONCR.NFV bit is set to 1 and the GTONCR.NVA bit is set to 0).

Note that once the negate control is performed, the negate control will not be released in the same cycle if the negate condition is no longer satisfied. The negate control is released in the next cycle.

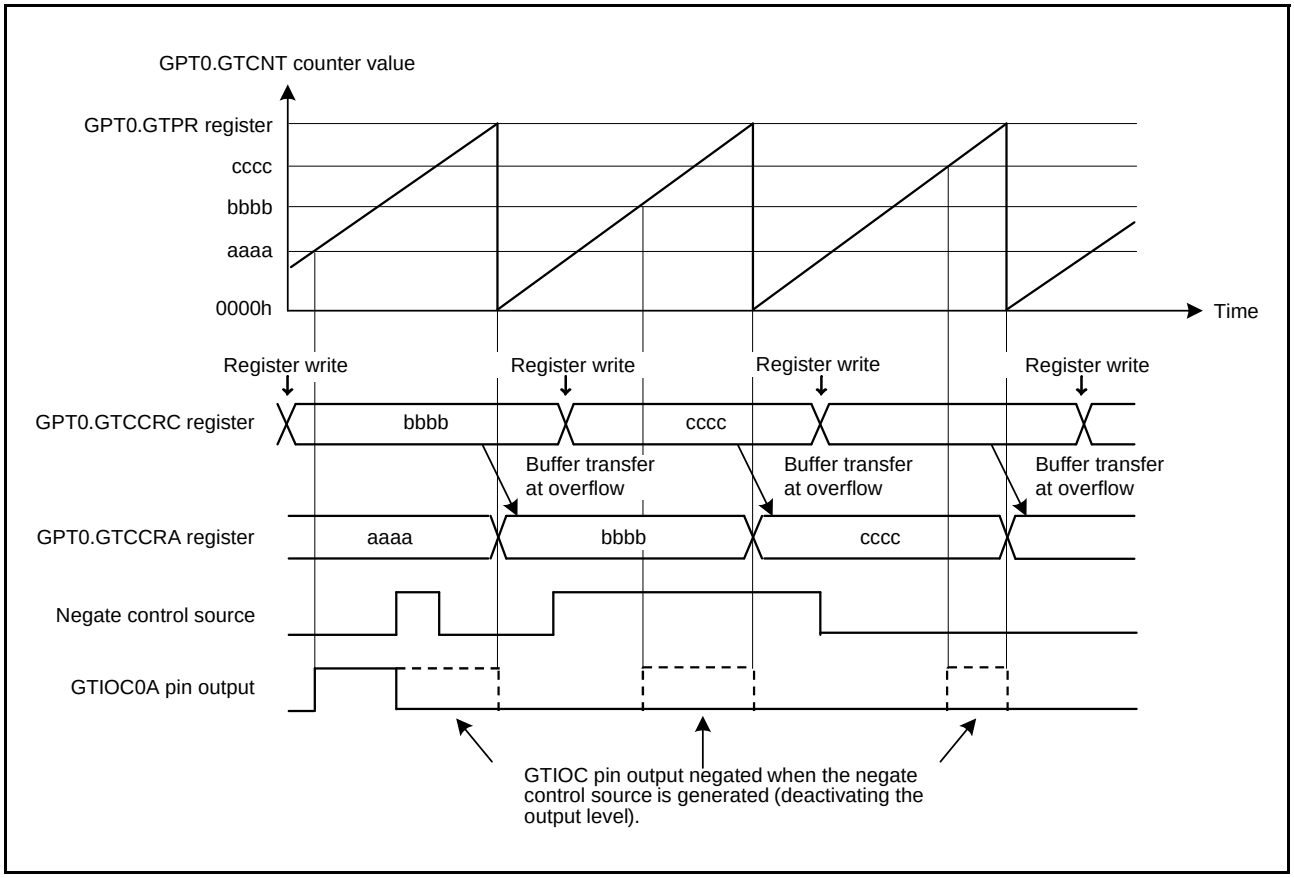


Figure 26.79 Example of GTIOC Pin Output Negate Control Operation (Saw-Wave Up-Counting, Buffer Operation, Active Level: 1, High Output at GTCCRA Compare Match, Low Output at the End of the Cycle)

26.7.4 Output Protection Function for GTIOC Pin Output

Under the automatic dead time setting (GTDTCR.TDE bit = 1) in triangle-wave PWM mode, if an incorrect value (0000h or a value greater than or equal to the GTPR value) is set in the GTCCRA register, the output protection function for the GTIOC pin output (disabling function) is activated.

The status of the output protection function can be read from the GTSOS.SOS[1:0] bits.

The output protection for the GTIOCnB pin can temporarily be released by setting the GTSOTR.SOTR bit to 1 only when an incorrect value (greater than or equal to the GTPR setting value) during buffer transfer at troughs is transferred to the GTCCRA register and output protection (disabling) is activated (the GTSOS.SOS[1:0] bits are 10b). After setting the GTSOTR.SOTR bit to 1, the output protection function is released from the first trough. When setting the GTSOTR.SOTR bit to 0 while the output protection function is released, the output protection function operates from the first trough.

Figure 26.80 shows the output protection function state transition.

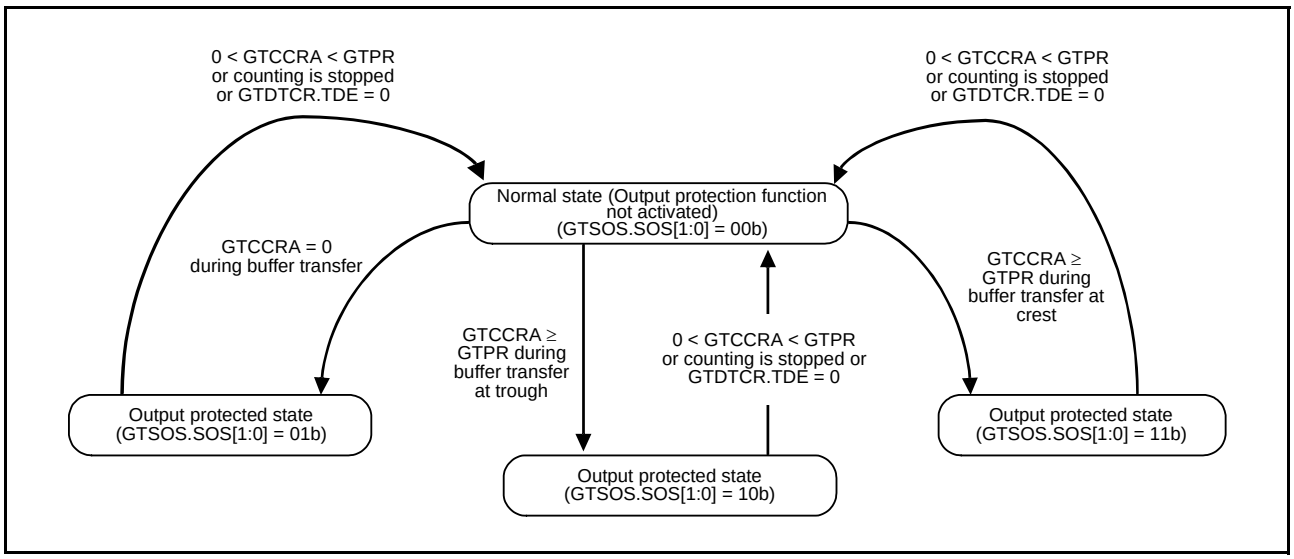
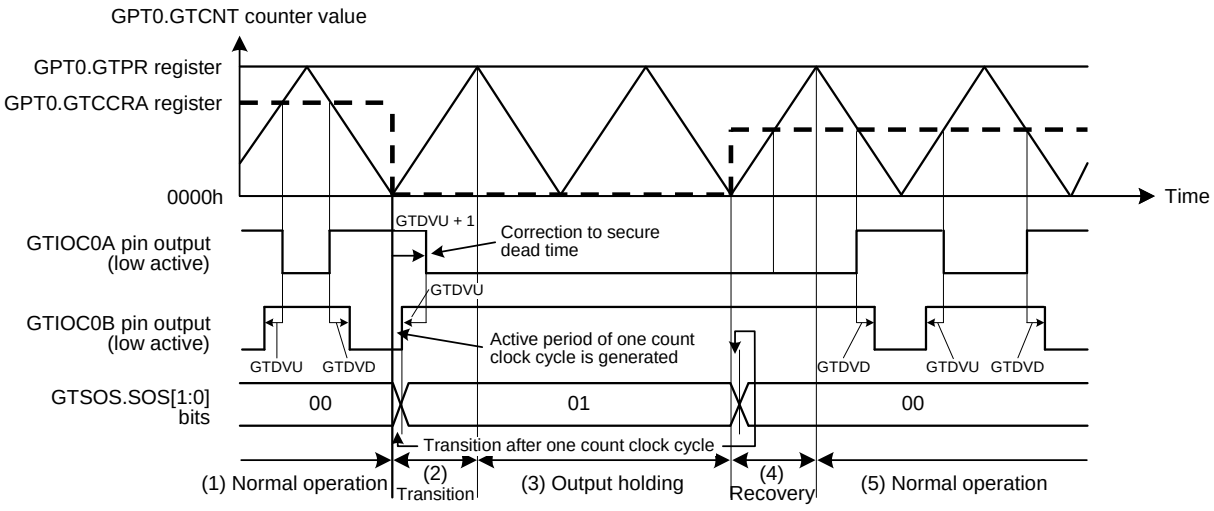


Figure 26.80 Output Protection Function

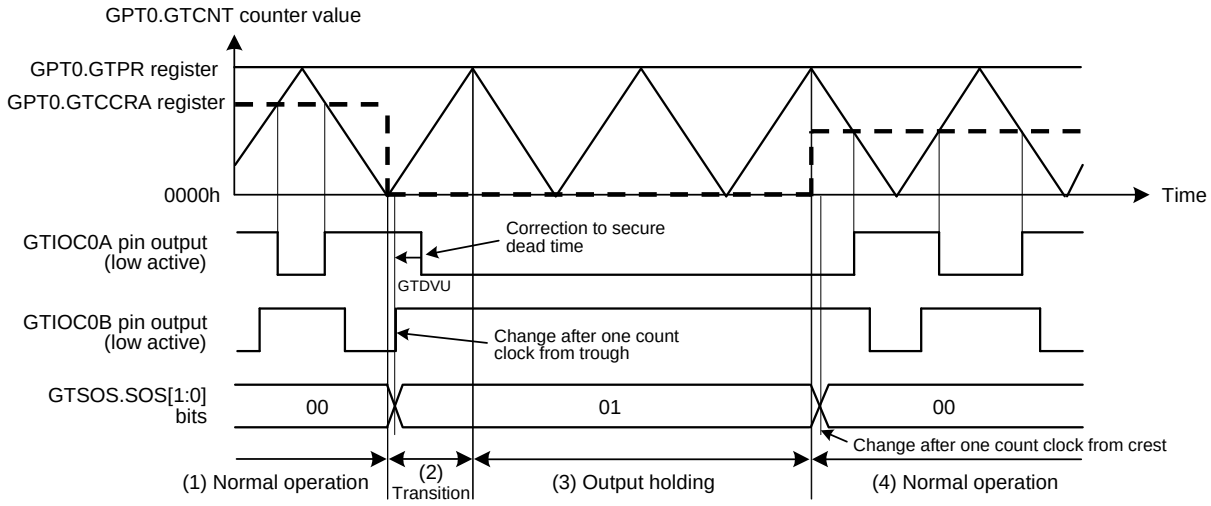
(1) Output Protection Function When the GTCCRA Register is Set to 0000h during Buffer Transfer

Figure 26.81 and Figure 26.82 show examples of output protection function operation when the GTCCRA register is set to 0000h during buffer transfer at troughs, and Figure 26.83 and Figure 26.84 show examples when the GTCCRA register is set to 0000h during buffer transfer at crests.



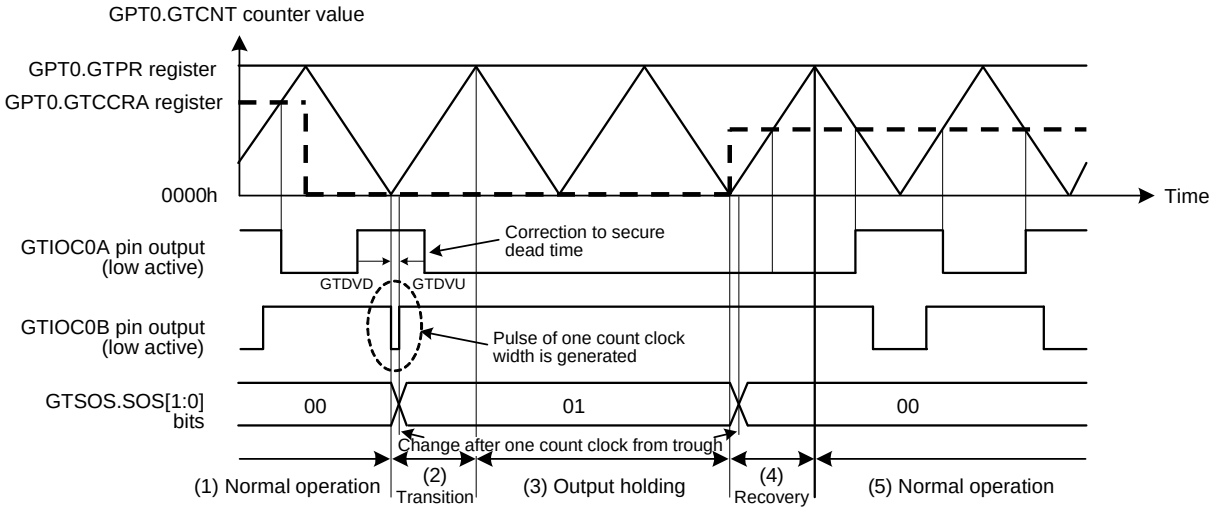
(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Transition period: When GTCCRA register = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest.
(3) Output holding period: The output is held while GTCCRA register = 0.
(4) Recovery period: When 0 < GTCCRA register < GTPR register is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and holding output is maintained up to the next crest.
(5) Normal operation period: Toggle output due to a compare match is performed normally.

Figure 26.81 Example of Output Protection Function Operation When the GTCCRA Register is Set to 0000h during Buffer Transfer at Troughs (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Troughs, Active Level: Low)



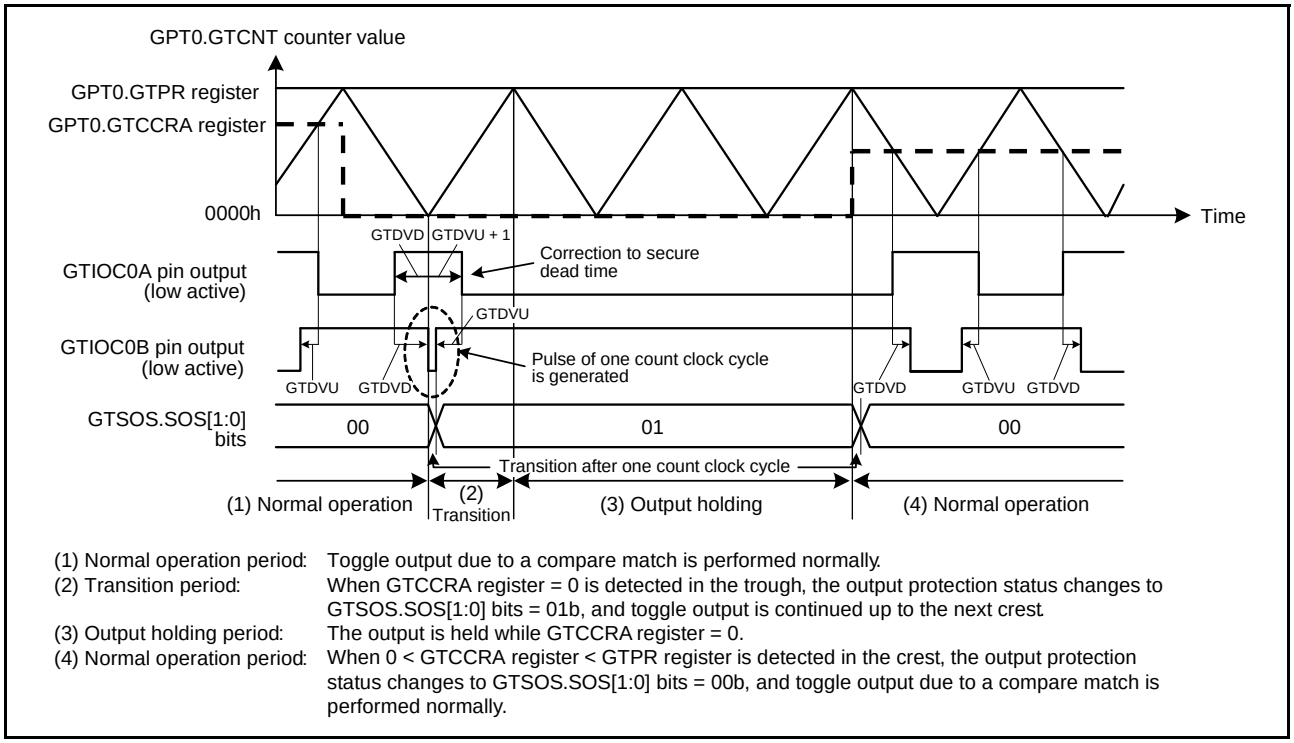
(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Transition period: When GTCCRA register = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest.
(3) Output holding period: The output is held while GTCCRA register = 0.
(4) Normal operation period: When 0 < GTCCRA register < GTPR register is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 26.82 Example of Output Protection Function Operation When the GTCCRA Register is Set to 0000h during Buffer Transfer at Troughs (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Crests, Active Level: Low)



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Transition period: When GTCCRA register = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest.
(3) Output holding period: The output is held while GTCCRA register = 0.
(4) Recovery period: When 0 < GTCCRA register < GTPR register is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and holding output is maintained up to the next crest.
(5) Normal operation period: Toggle output due to a compare match is performed normally.

Figure 26.83 Example of Output Protection Function Operation When the GTCCRA Register is Set to 0000h during Buffer Transfer at Crests (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Troughs, Active Level: Low)

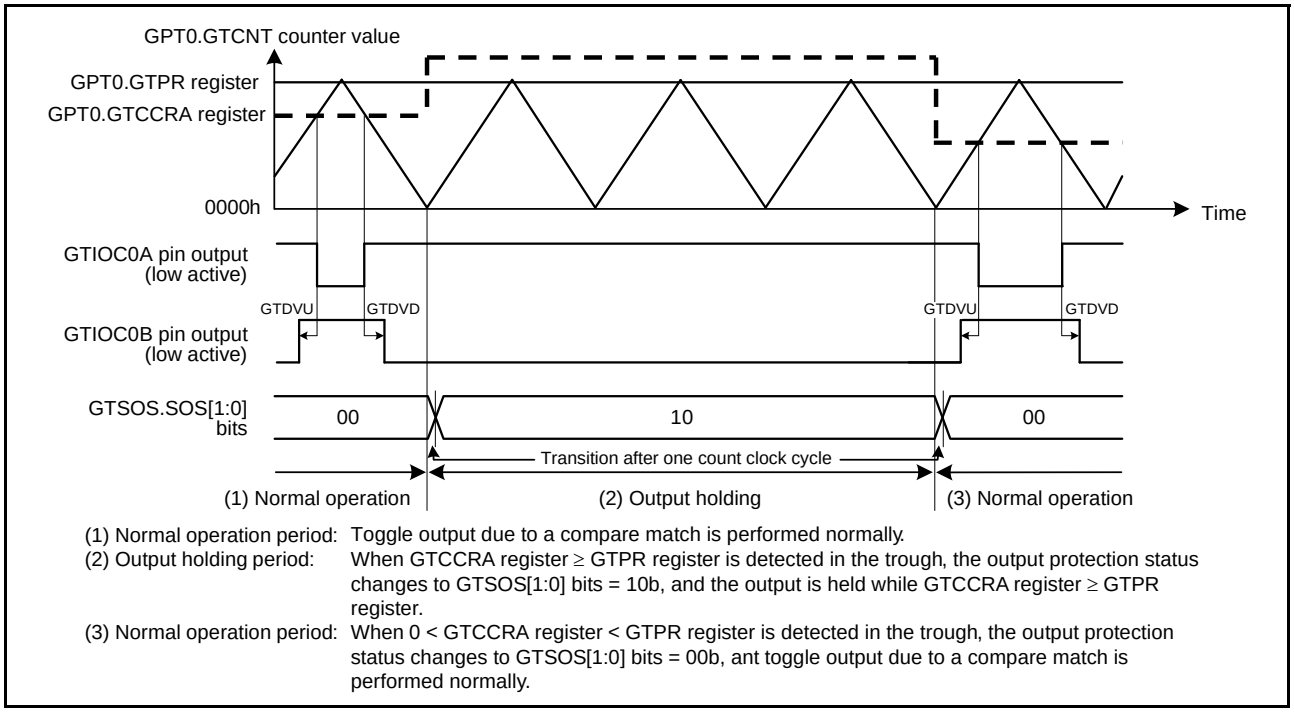


(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Transition period: When GTCCRA register = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest.
(3) Output holding period: The output is held while GTCCRA register = 0.
(4) Normal operation period: When 0 < GTCCRA register < GTPR register is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 26.84 Example of Output Protection Function Operation When the GTCCRA Register is Set to 0000h during Buffer Transfer at Crests (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Crests, Active Level: Low)

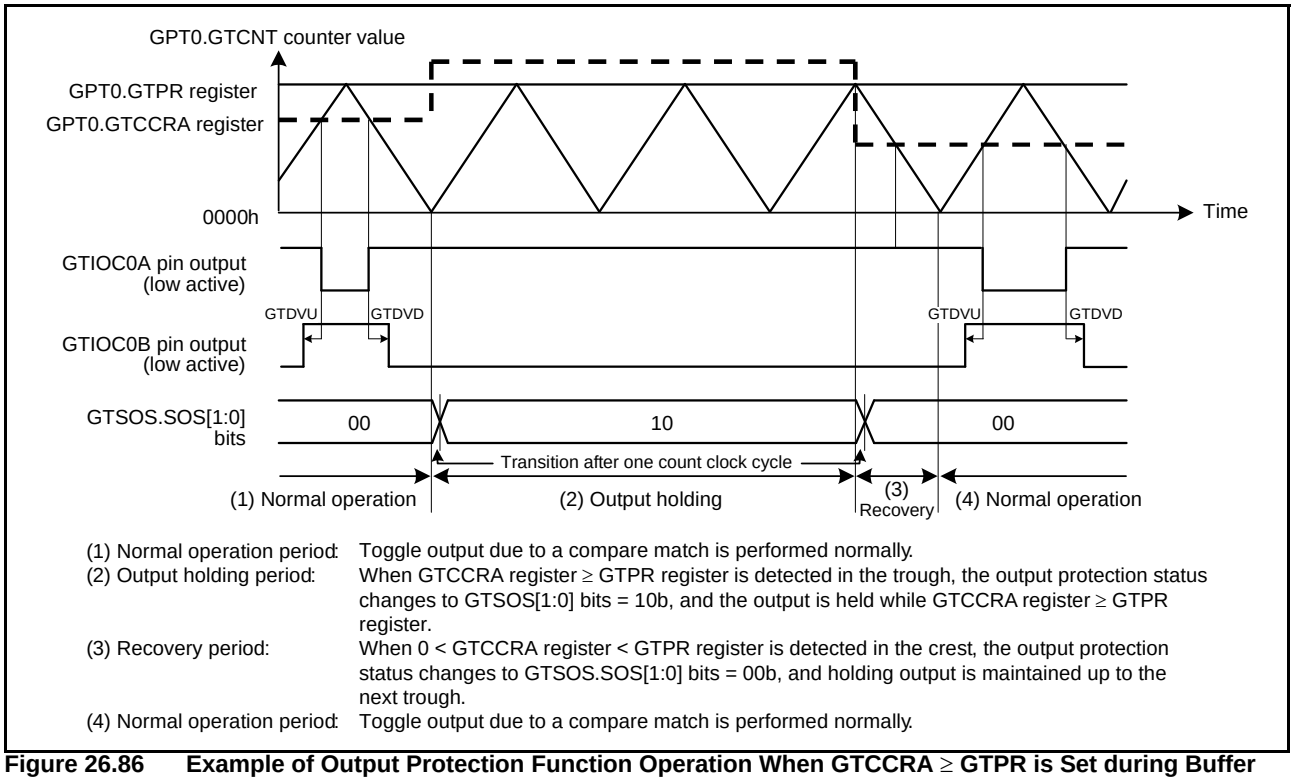
(2) Output Protection Function When GTCCRA ≥ GTPR is Set during Buffer Transfer at Troughs

Figure 26.85 and Figure 26.86 show examples of output protection function operation when GTCCRA ≥ GTPR is set during buffer transfer at troughs.



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA register ≥ GTPR register is detected in the trough, the output protection status changes to GTSOS[1:0] bits = 10b, and the output is held while GTCCRA register ≥ GTPR register.
(3) Normal operation period: When 0 < GTCCRA register < GTPR register is detected in the trough, the output protection status changes to GTSOS[1:0] bits = 00b, ant toggle output due to a compare match is performed normally.

Figure 26.85 Example of Output Protection Function Operation When GTCCRA ≥ GTPR is set during Buffer Transfer at Troughs (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Troughs, Active Level: Low)

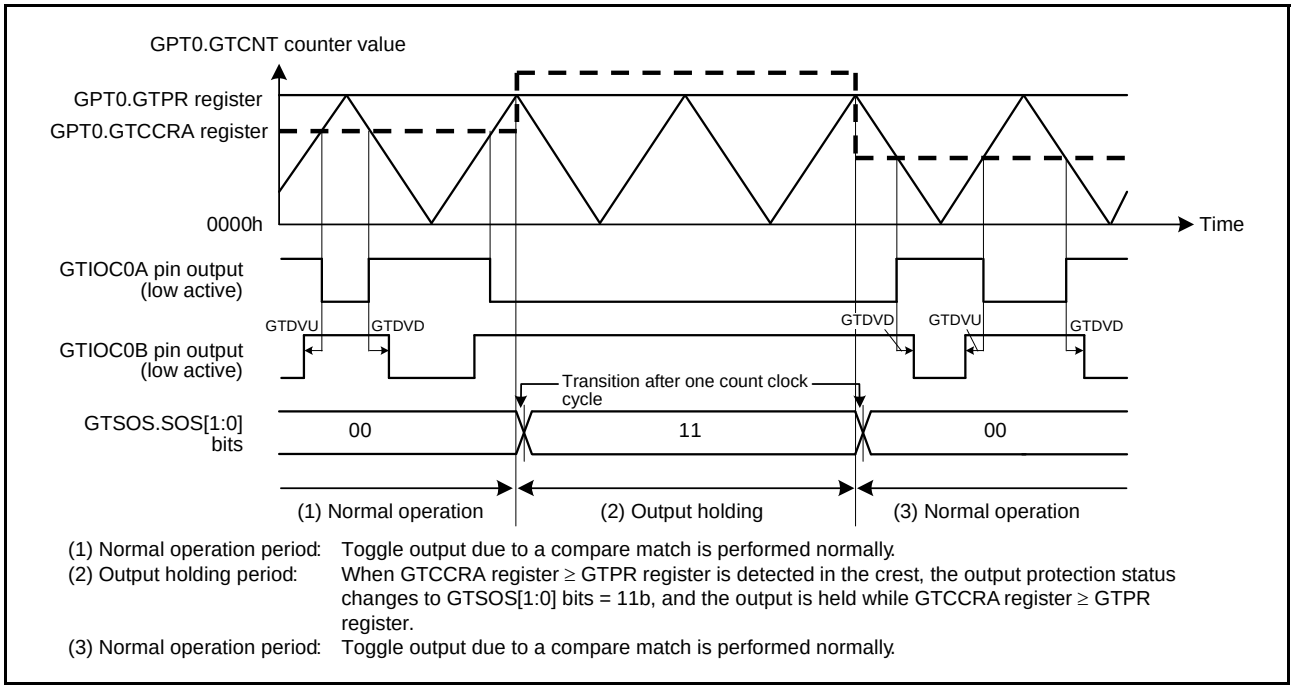


(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA register ≥ GTPR register is detected in the trough, the output protection status changes to GTSOS[1:0] bits = 10b, and the output is held while GTCCRA register ≥ GTPR register.
(3) Recovery period: When 0 < GTCCRA register < GTPR register is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and holding output is maintained up to the next trough.
(4) Normal operation period: Toggle output due to a compare match is performed normally.

Figure 26.86 Example of Output Protection Function Operation When GTCCRA ≥ GTPR is Set during Buffer Transfer at Troughs (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Crests, Active Level: Low)

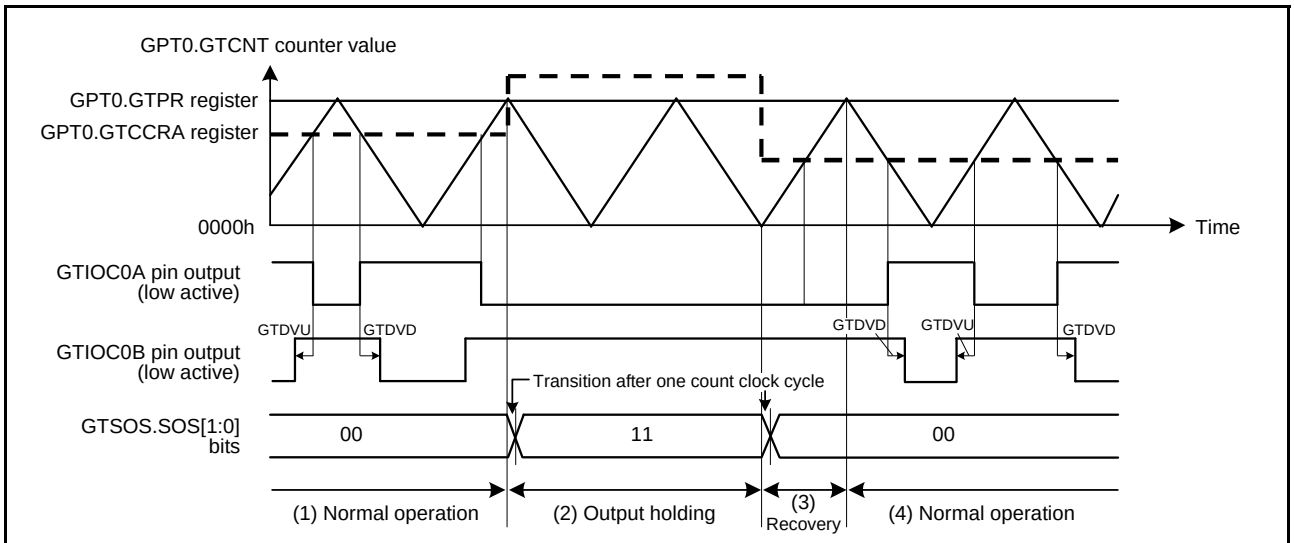
(3) Output Protection Function When GTCCRA ≥ GTPR is Set during Buffer Transfer at Crests

Figure 26.87 and Figure 26.88 show examples of output protection function operation when GTCCRA ≥ GTPR is set during buffer transfer at crests.



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA register ≥ GTPR register is detected in the crest, the output protection status changes to GTSOS[1:0] bits = 11b, and the output is held while GTCCRA register ≥ GTPR register.
(3) Normal operation period: Toggle output due to a compare match is performed normally.

Figure 26.87 Example of Output Protection Function Operation When GTCCRA ≥ GTPR is Set during Buffer Transfer at Crests (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Crests, Active Level: Low)



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA register ≥ GTPR register is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 11b, and the output is held while GTCCRA register ≥ GTPR register.
(3) Recovery period: When 0 < GTCCRA register < GTPR register is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and holding output is maintained up to the next crest.
(4) Normal operation period: Toggle output due to a compare match is performed normally.

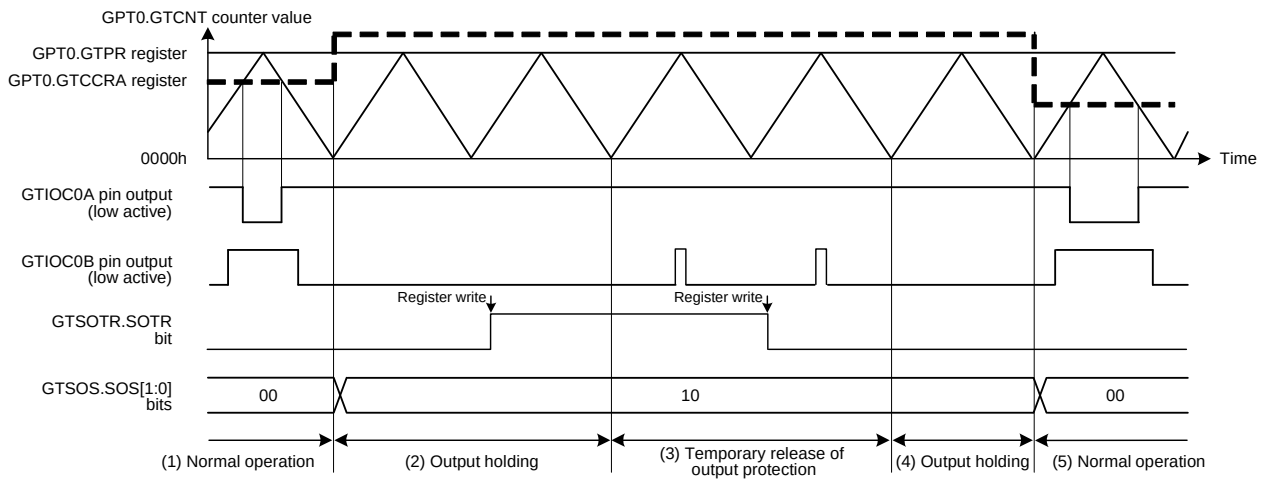
Figure 26.88 Example of Output Protection Function Operation When GTCCRA ≥ GTPR is Set during Buffer Transfer at Crests (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Troughs, Active Level: Low)

(4) Restricted Specification of Output Protection Function

When the GTCCRA value is greater than 0000h, and less than the setting value of the GTPR register, start count operation.
Even if an incorrect value (0000h or a value greater than or equal to the GTPR value) is set in the GTCCRA register during count operation, the output protection functions in a specific way such that one of the positive- and negative-phase outputs becomes non-active. However, when count operation is started while an incorrect value is set in the GTCCRA register, the output protection does not operate normally.

(5) Temporary Release of Output Protection Function

When the GTSOS.SOS[1:0] bits = 10b (protected state in which GTCCRA ≥ GTPR has occurred during transfer at trough), the protected state of the GTIOCnBF pin output can be temporarily released by setting the GTSOTR.SOTR bit to 1. The GTSOS.SOS[1:0] bits retain 10b even if the output protection function is released.
When the SOTR bit is set to 0, the GTIOCnB pin output protection can be restarted.
Figure 26.89 shows an example of output protection function operation when GTCCRA ≥ GTPR is set during buffer transfer at troughs.



(1) Normal operation period:	Toggle output due to a compare match is performed normally.
(2) Output holding period:	When GTCCRA register ≥ GTPR register is detected in the trough, the output protection status changes to GTSOS[1:0] bits = 10b, and the output is held while GTCCRA register ≥ GTPR register.
(3) Output protection temporary release period:	Even when the GTSOTR.SOTR bit is set to 1, the GTIOCOB pin output is held until the first trough.
(4) Output holding period:	After the GTSOTR.SOTR bit is set to 1, the GTIOCOB pin output is performed normally from the first trough. The output is held until 0 < GTCCRA register < GTPR register is detected in the tough.
(5) Normal operation period:	When 0 < GTCCRA register < GTPR register is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 26.89 Example of Output Protection Function Operation When GTCCRA ≥ GTPR is Set during Buffer Transfer at Troughs (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Troughs, Active Level: Low)

26.7.5 High-Impedance Control of GTIOC Pin Output by POE Function

For protection from system failure, the high-impedance state of the GTIOC pin output can be controlled by the port output enable (POE) function.
For details, see section 25, Port Output Enable 3 (POE3a).

26.8 Initialization Method of Output Pins

26.8.1 Pin Settings after Reset

The GPT registers are initialized at a reset. Start counting after selecting the port mode (PMR and PmnPFS), setting the GTIOR register and the OAE and OBE bits in the GTONCR register and outputting the GPT function to external pins.

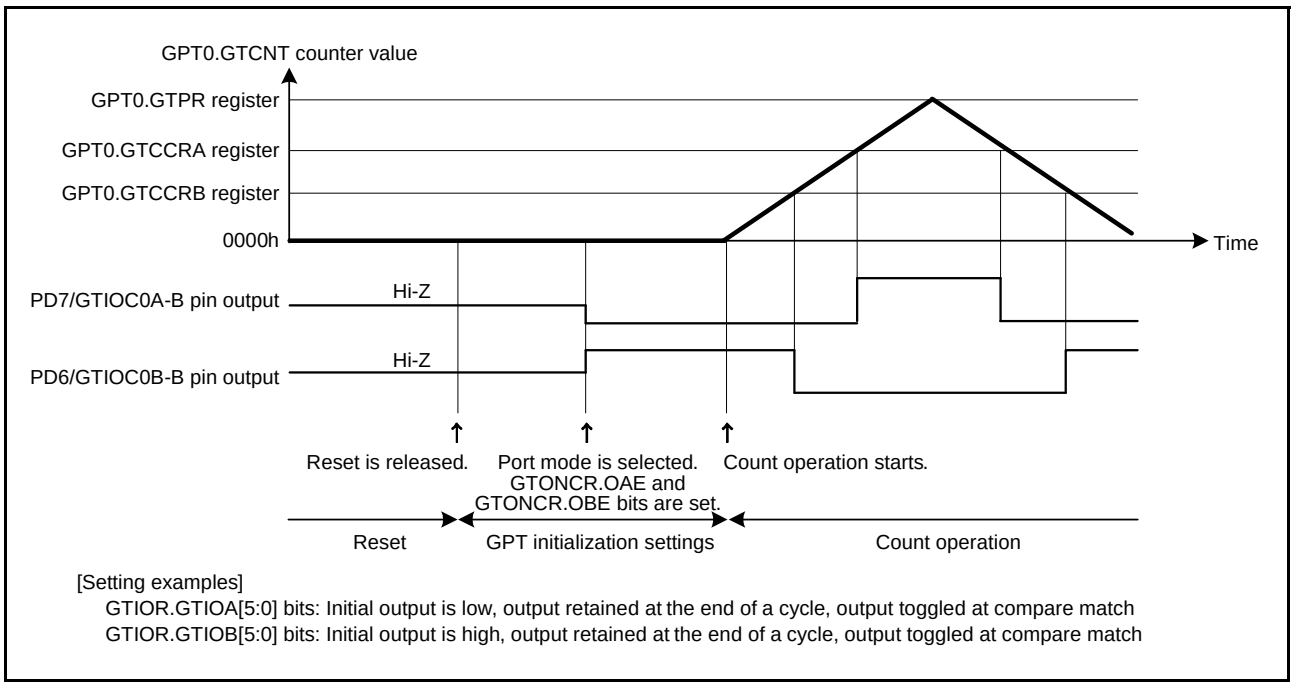


Figure 26.90 Example of Pin Settings after Reset

26.8.2 Pin Initialization Due to Error during Operation

If an error occurs during GPT operation, the following four types of pin processing can be performed before pin initialization.

(1) Set OAHLD and OBHLD bits in the GTIOR register to 1 and retain the outputs at count stop.

(2) Set OAHLD and OBHLD bits in the GTIOR register to 0, specify arbitrary output values at OADFLT and OBDFLT in the GTIOR register, and output the arbitrary values at count stop.

(3) Set the pin to output an arbitrary value as a general output port by setting the PDR, PODR, and PMR registers of the I/O port in advance. Set the OAE and OBE bits in the GTONCR register to 0 and the control bit in PMR that corresponds to the pin to 0 to allow the arbitrary values to be output from the pin set as a general output port when an error occurs.

(4) Drive the output to a high impedance state using the POE function.

When automatic dead time setting has been made, set the GTDTCR.TDE bit to 0 once after counting is stopped.

When counting is stopped, only the values of registers that are changed by a GPT external source will change. If counting is resumed, operation will carry on from where it was stopped.
If counting was stopped, registers should be initialized before counting is started.

26.9 Usage Notes

26.9.1 Module Stop Function Setting

Operation of the GPT can be disabled or enabled by the module stop control register. The initial setting is for operation of the GPT to be halted. Register access is enabled by clearing module stop state. For details, see section 11, Low Power Consumption.

26.9.2 Settings of the GTCCRm Register during Compare Match Operation (m = A to F)

(1) When automatic dead time setting has been made in triangle-wave PWM mode

The GTCCRA register should satisfy the following conditions:

GTCCRA > GTDVU,
GTCCRA > GTDVD, and
GTCCRA < GTPR.

When the GTCCRA register is set to 0000h or a value greater than or equal to the GTPR value during count operation, the output protection function is activated.
When the GTCCRA value is greater than 0000h, and less than the setting value of the GTPR register, start count operation. If count operation is started while 0000h or a value greater than or equal to the GTPR value is set in the GTCCRA register, the output protection does not operate normally.
For details, refer to section 26.7.4, Output Protection Function for GTIOC Pin Output.

(2) When automatic dead time setting has not been made in triangle-wave PWM mode

Set a value greater than 0000h, and less than the setting value of the GTPR register in the GTCCRA register. When 0000h or the same value as that of the GTPR register is set in the GTCCRA register, compare match is generated in one cycle only when [GTCCRA = 0000h] or [GTCCRA = GTPR] is met. Furthermore, a value exceeding the setting value of the GTPR register is set in the GTCCRA register, compare match does not occur.
Similarly, set a value greater than 0000h, and less than the setting value of the GTPR register in the GTCCRB register. When 0000h or the same value as that of the GTPR register is set in the GTCCRB register, compare match is generated in one cycle only when [GTCCRB = 0000h] or [GTCCRB = GTPR] is met. Furthermore, a value exceeding the setting value of the GTPR register is set in the GTCCRB register, compare match does not occur.

(3) When automatic dead time setting has been made in saw-wave one-shot pulse mode

The GTCCRC and GTCCRD registers should be set to satisfy the following restrictions. If the restrictions are not satisfied, correct output waveforms with secured dead time may not be obtained.

- In up-counting:
  GTCCRC < GTCCRD
  GTCCRC > GTDVU
  GTCCRD < GTPR – GTDVD
- In down-counting:
  GTCCRC > GTCCRD
  GTCCRC < GTPR – GTDVU
  GTCCRD > GTDVD

(4) When automatic dead time setting has not been made in saw-wave one-shot pulse mode

The GTCCRC and GTCCRD registers should be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRC < GTCCRD < GTPR
- In down-counting: GTPR > GTCCRC > GTCCRD > 0

Similarly, GTCCRE and GTCCRF should be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRE < GTCCRF < GTPR
- In down-counting: GTPR > GTCCRE > GTCCRF > 0

(5) In saw-wave PWM mode

The GTCCRA register should be set with the range of 0 < GTCCRA < GTPR. If GTCCRA = 0000h or GTCCRA = GTPR is set, a compare match occurs within the cycle only when GTCCRA = 0000h or GTCCRA = GTPR is satisfied. If GTCCRA > GTPR is set, no compare match occurs.
Similarly, GTCCRB should be set with the range of 0 < GTCCRB < GTPR. If GTCCRB = 0000h or GTCCRB = GTPR is set, a compare match occurs within the cycle only when GTCCRB = 0000h or GTCCRB = GTPR is satisfied. If GTCCRB > GTPR is set, no compare match occurs.

26.9.3 Stopping the Timer in the Safe Way

When the timer stopping by the GTSTR register writing and the GPT compare match interrupt conflict, an interrupt may be generated after the GTSTR register writing.
Therefore, stop the timer in the following order. Then, a compare match interrupt is not generated after the timer has been stopped, and the timer can be stopped in the safe way.

(1) Disable the GPT interrupt request by the IER register of the ICU.
(2) Disable the interrupt request by the GTINTAD register.
(3) Set the GTSTR.CSTn bit to 0.

26.9.4 Order of Priority in Event Counter Operation

(1) Contention between writing to and clearing of the GTCNT counter

If the counter clearing signal is generated during a cycle of writing to the GTCNT counter, writing to the GTCNT counter takes priority over clearing of the GTCNT counter.

(2) Contention between writing to the GTSTR.CSTn bit and a hardware source/event signal

If a hardware source or event signal from the ELC is generated in a cycle of writing to the GTSTR.CSTn bit (n = 0 to 3), writing to the CSTn bit takes priority over an automatic set or reset of the CSTn bit due to the hardware source.

(3) Contention between a set and a reset by a hardware source to the GTSTR.CSTn bit or event signal

When a set and a reset by a hardware source to the GTSTR.CSTn bit (n = 0 to 3) or event signal from the ELC occurs at the same time, an automatic set takes priority.

(4) Contention between access to the GTCCRm register and input capture/buffer transfer

In case of contention between the writing to the GTCCRm register (m = A to F) and input capture or buffer transfer, writing to the GTCCRm register takes priority over input capture or buffer transfer.
If reading from the GTCCRm register and input capture or buffer transfer are in contention, the data before an update is read.

(5) Contention between access to the GTPR register and buffer transfer

If writing to the GTPR register and buffer transfer are in contention, writing to the GTPR register takes priority over buffer transfer.

If reading from the GTPR register and buffer transfer are in contention, the data before an update is read.

(6) Contention between access to the GTADTRm register and buffer transfer

If writing to the GTADTRm register (m = A, B) and buffer transfer are in contention, writing to the GTADTRm register takes priority over the buffer transfer.
If reading from the GTADTRm register and buffer transfer are in contention, the data before an update is read.

(7) Contention between access to the GTDVm register and buffer transfer

If writing to the GTDVm register (m = U, D) and buffer transfer are in contention, writing to the GTDVm register takes priority over the buffer transfer.
If reading from the GTDVm register and buffer transfer are in contention, the data before an update is read.

27. 16-Bit Timer Pulse Unit (TPUa)

This MCU has on-chip 16-bit timer pulse units (TPU) comprising six-channel 16-bit timers.
In this section, “PCLK” is used to refer to PCLKB.

27.1 Overview

Specifications of the TPU are listed in Table 27.1. Functions of TPU are listed in Table 27.2.
Figure 27.1 shows a block diagram of TPU.

Table 27.1 Specifications of TPU

Item	Description
Pulse input/output	Maximum 16
Count clocks	Seven or eight types are provided for each channel.
Settable operations	• Waveform output at compare match • Input capture function (noise filters can be set) • Counter clear operation • Simultaneous writing to multiple timer counters (TCNT) • Simultaneous clearing by compare match and input capture • Synchronous input/output for registers by counter synchronous operation • Maximum of 15-phase PWM output by combination with synchronous operation • Cascaded operation
TPU0 and TPU3	Buffer operation can be set.
TPU1, TPU2, TPU4, and TPU5	Phase counting mode can be set.
Interrupt sources	26 sources
Buffer operation	Automatic transfer of register data
Generation of trigger	Programmable pulse generator (PPG) output trigger can be generated.
	Conversion start trigger for the A/D converter can be generated.
Event linking (output)	• Six types of event signal can be output to the ELC. Compare match A (TPU0 to TPU3) Compare match B (TPU0 to TPU3) Compare match C (TPU0, TPU3) Compare match D (TPU0, TPU3) Overflow (TPU0 to TPU3) Underflow (TPU1, TPU2)
Event linking (input)	• Any of the three operations in response to event input is possible. Starting counts (TPU0 to TPU3) Restarting counts (TPU0 to TPU3) Input capture operation (TPU0 to TPU3)
Low power consumption function	Module stop state can be set.

Table 27.2 TPU Functions (1/2)

Item		TPU0	TPU1	TPU2	TPU3	TPU4	TPU5
Count clocks		PCLK/1 PCLK/4 PCLK/16 PCLK/64 TCLKA TCLKB TCLKC TCLKD	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 TCLKA TCLKB	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 TCLKA TCLKB TCLKC	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 PCLK/1024 PCLK/4096 TCLKA	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 TCLKA TCLKC	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 TCLKA TCLKC TCLKD
External clocks for phase counting mode		Not possible	TCLKA TCLKB	TCLKC TCLKD	Not possible	TCLKC TCLKD	TCLKA TCLKB
Timer general registers		TGRA TGRB TGRC*1 TGRD*1	TGRA TGRB	TGRA TGRB	TGRA TGRB TGRC*1 TGRD*1	TGRA TGRB	TGRA TGRB
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4	TIOCA5 TIOCB5
Counter clear function (y = A to D)		TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture
Compare match output	Low output	Possible	Possible	Possible	Possible	Possible	Possible
	High output	Possible	Possible	Possible	Possible	Possible	Possible
	Toggle output	Possible	Possible	Possible	Possible	Possible	Possible
Input capture function		Possible	Possible	Possible	Possible	Possible	Possible
Synchronous operation		Possible	Possible	Possible	Possible	Possible	Possible
PWM mode		Possible	Possible	Possible	Possible	Possible	Possible
Phase counting mode		Not possible	Possible	Possible	Not possible	Possible	Possible
Buffer operation		Possible	Not possible	Not possible	Possible	Not possible	Not possible
DTC activation (y = A to D)		TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture
DMAC activation		TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture
A/D conversion start trigger		TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	Not possible
PPG trigger		TGRA/TGRB compare match or input capture	TGRA/TGRB compare match or input capture	TGRA/TGRB compare match or input capture	TGRA/TGRB compare match or input capture	Not possible	Not possible
Interrupt sources		5 sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	5 sources • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	4 sources • Compare match or input capture 4A • Compare match or input capture 4B • Overflow • Underflow	4 sources • Compare match or input capture 5A • Compare match or input capture 5B • Overflow • Underflow

Table 27.2 TPU Functions (2/2)

Item	TPU0	TPU1	TPU2	TPU3	TPU4	TPU5
Event linking (output)	5 sources • Compare match 0A • Compare match 0B • Compare match 0C • Compare match 0D • Overflow	4 sources • Compare match 1A • Compare match 1B • Overflow • Underflow	4 sources • Compare match 2A • Compare match 2B • Overflow • Underflow	5 sources • Compare match 3A • Compare match 3B • Compare match 3C • Compare match 3D • Overflow	Not possible	Not possible
Event linking (input)	• Starting counts • Restarting counts • Input capture operation (data is captured in TGRA)	• Starting counts • Restarting counts • Input capture operation (data is captured in TGRA)	• Starting counts • Restarting counts • Input capture operation (data is captured in TGRA)	• Starting counts • Restarting counts • Input capture operation (data is captured in TGRA)	Not possible	Not possible
Module stop setting*2	MSTPCRA.MSTPA13 bit					

Note 1. TGRC and TGRD can be set as a buffer register.
Note 2. For details, see section 11, Low Power Consumption.

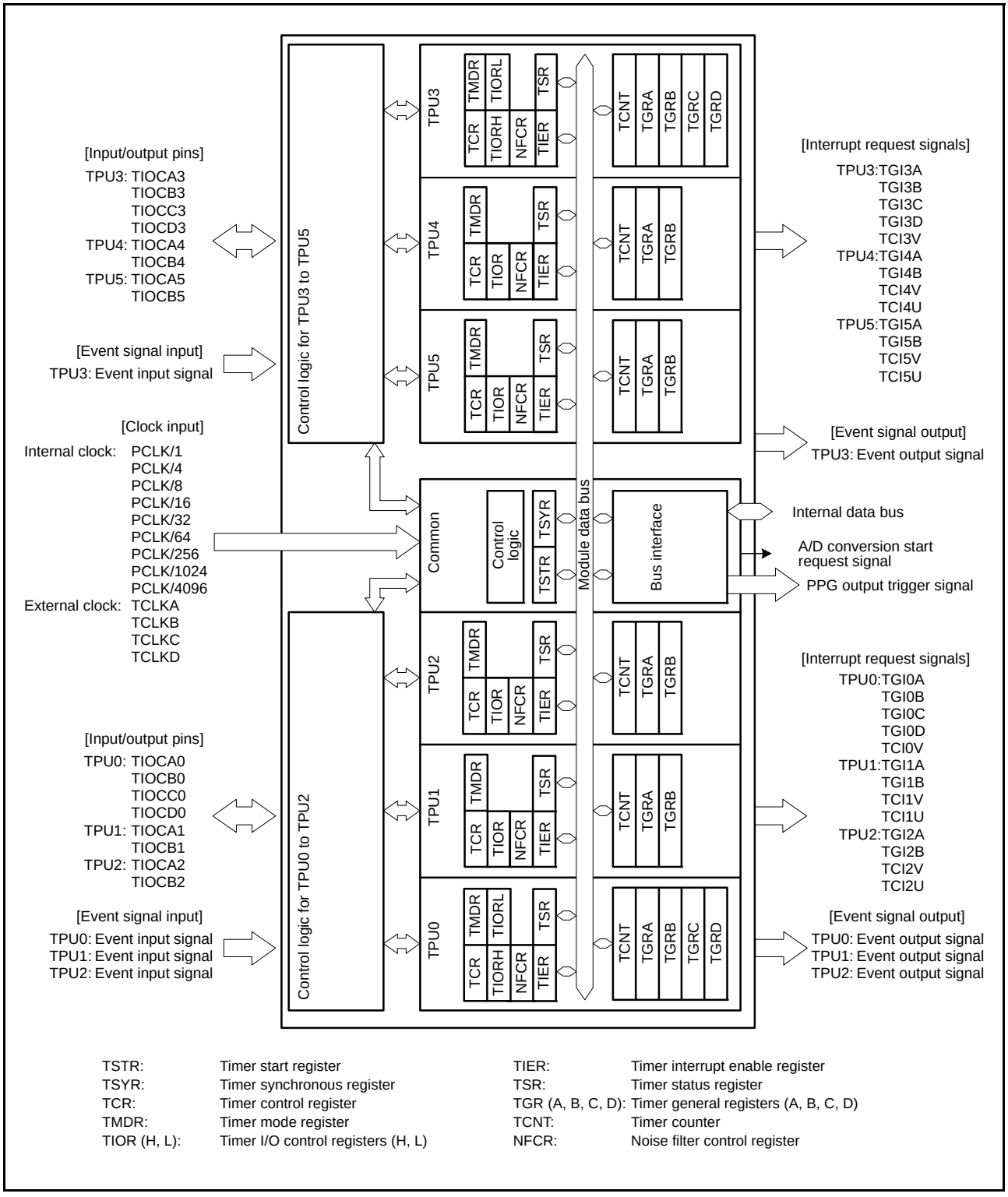


Figure 27.1 Block Diagram of TPU

Table 27.3 lists the input/output pins of the TPU.

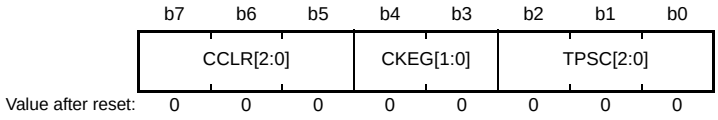
Table 27.3 Pin Configuration of TPU

Channel	Pin Name	I/O	Description
Common	TCLKA	Input	External clock A input pin (TPU1 and TPU5 phase counting mode A phase input)
	TCLKB	Input	External clock B input pin (TPU1 and TPU5 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (TPU2 and TPU4 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (TPU2 and TPU4 phase counting mode B phase input)
TPU0	TIOCA0	I/O	TPU0.TGRA input capture input/output compare output/PWM output pin
	TIOCB0	I/O	TPU0.TGRB input capture input/output compare output/PWM output pin
	TIOCC0	I/O	TPU0.TGRC input capture input/output compare output/PWM output pin
	TIOCD0	I/O	TPU0.TGRD input capture input/output compare output/PWM output pin
TPU1	TIOCA1	I/O	TPU1.TGRA input capture input/output compare output/PWM output pin
	TIOCB1	I/O	TPU1.TGRB input capture input/output compare output/PWM output pin
TPU2	TIOCA2	I/O	TPU2.TGRA input capture input/output compare output/PWM output pin
	TIOCB2	I/O	TPU2.TGRB input capture input/output compare output/PWM output pin
TPU3	TIOCA3	I/O	TPU3.TGRA input capture input/output compare output/PWM output pin
	TIOCB3	I/O	TPU3.TGRB input capture input/output compare output/PWM output pin
	TIOCC3	I/O	TPU3.TGRC input capture input/output compare output/PWM output pin
	TIOCD3	I/O	TPU3.TGRD input capture input/output compare output/PWM output pin
TPU4	TIOCA4	I/O	TPU4.TGRA input capture input/output compare output/PWM output pin
	TIOCB4	I/O	TPU4.TGRB input capture input/output compare output/PWM output pin
TPU5	TIOCA5	I/O	TPU5.TGRA input capture input/output compare output/PWM output pin
	TIOCB5	I/O	TPU5.TGRB input capture input/output compare output/PWM output pin

27.2 Register Descriptions

27.2.1 Timer Control Register (TCR)

Address(es): TPU0.TCR 0008 8110h, TPU1.TCR 0008 8120h, TPU2.TCR 0008 8130h, TPU3.TCR 0008 8140h, TPU4.TCR 0008 8150h, TPU5.TCR 0008 8160h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC[2:0]	Timer Prescaler Select	See Table 27.4 to Table 27.9.	R/W
b4, b3	CKEG[1:0]	Input Clock Edge Select	See Table 27.10.	R/W
b7 to b5	CCLR[2:0]*1	Counter Clear Source Select	See Table 27.11 and Table 27.12.	R/W

Note 1. Bit 7 in TCR of TPU1, TPU2, TPU4, and TPU5 are reserved. These bits are read as 0. The write value should be 0.

TPUm.TCR settings should be made while TPUm.TCNT operation is stopped.

TPSC[2:0] Bits (Timer Prescaler Select)

These bits select the TCNT clock. The clock source can be selected independently for each channel.
To select the external clock as the clock source, set the bit in the port direction register (PDR) for the corresponding pin to 0 (input port), and set the bit in the port mode register (PMR) to 1 (uses the pin as an I/O port for peripheral functions). For details, see section 22, I/O Ports.

CKEG[1:0] Bits (Input Clock Edge Select)

These bits select the input clock edge.
When the internal clock is counted using both edges, the input clock period is halved (e.g. Both edges of PCLK/4 = PCLK/2 rising edge).
Internal clock edge selection is valid when the input clock is PCLK/4 or slower. This setting is ignored if the input clock is PCLK/1, or when overflow/underflow of another channel is selected.

Table 27.4 Bits TPSC[2:0] (TPU0)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU0	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	External clock: counts on TCLKB pin input
	1	1	0	External clock: counts on TCLKC pin input
	1	1	1	External clock: counts on TCLKD pin input

Table 27.5 Bits TPSC[2:0] (TPU1)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU1	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	External clock: counts on TCLKB pin input
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	Counts on TPU2.TCNT overflow/underflow

Note: This setting is invalid when TPU1 is in phase counting mode.

Table 27.6 Bits TPSC[2:0] (TPU2)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU2	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	External clock: counts on TCLKB pin input
	1	1	0	External clock: counts on TCLKC pin input
	1	1	1	Internal clock: counts on PCLK/1024

Note: This setting is invalid when TPU2 is in phase counting mode.

Table 27.7 Bits TPSC[2:0] (TPU3)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU3	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	Internal clock: counts on PCLK/1024
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	Internal clock: counts on PCLK/4096

Table 27.8 Bits TPSC[2:0] (TPU4)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU4	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	External clock: counts on TCLKC pin input
	1	1	0	Internal clock: counts on PCLK/1024
	1	1	1	Counts on TPU5.TCNT overflow/underflow

Note: This setting is invalid when TPU4 is in phase counting mode.

Table 27.9 Bits TPSC[2:0] (TPU5)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU5	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	External clock: counts on TCLKC pin input
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	External clock: counts on TCLKD pin input

Note: This setting is invalid when TPU5 is in phase counting mode.

Table 27.10 Bits CKEG[1:0]

Bits CKEG[1:0]		Input Clock	
b4	b3	Internal Clock	External clock
0	0	Counted at falling edge	Counted at rising edge
0	1	Counted at rising edge	Counted at falling edge
1	0	Counted at both edges	Counted at both edges
1	1	Counted at both edges	Counted at both edges

Table 27.11 Bits CCLR[2:0] (TPU0, TPU3)

Channel	Bits CCLR[2:0]			Description
	b7	b6	b5	
TPU0, TPU3	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*2
	1	0	0	TCNT clearing disabled
	1	0	1	TCNT cleared by TGRC compare match/input capture*1
	1	1	0	TCNT cleared by TGRD compare match/input capture*1
	1	1	1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*2

Note 1. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.
Note 2. Synchronous operation is selected by setting the TPUA.TSYR.SYNCj bit (j = 0, 3) to 1.

Table 27.12 Bits CCLR[2:0] (TPU1, TPU2, TPU4, TPU5)

Channel	Bits CCLR[2:0]*1			Description
	b7	b6	b5	
TPU1, TPU2, TPU4, TPU5	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*2
	1	0	0	Setting prohibited
	1	0	1	Setting prohibited
	1	1	0	Setting prohibited
	1	1	1	Setting prohibited

Note 1. Bit 7 in TCR of TPU1, TPU2, TPU4, and TPU5 are reserved. These bits are read as 0. The write value should be 0.
Note 2. Synchronous operation is selected by setting the TPUA.TSYR.SYNCj bit (j = 1, 2, 4, 5) to 1.

27.2.2 Timer Mode Register (TMDR)

Address(es): TPU0.TMDR 0008 8111h, TPU1.TMDR 0008 8121h, TPU2.TMDR 0008 8131h, TPU3.TMDR 0008 8141h, TPU4.TMDR 0008 8151h, TPU5.TMDR 0008 8161h

	b7	b6	b5	b4	b3	b2	b1	b0
	ICSELD	ICSELB	BFB	BFA	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MD[3:0]	Mode Select	b3 b0 0 0 0 0: Normal operation 0 0 0 1: Setting prohibited 0 0 1 0: PWM mode 1 0 0 1 1: PWM mode 2 0 1 0 0: Phase counting mode 1*1 0 1 0 1: Phase counting mode 2*1 0 1 1 0: Phase counting mode 3*1 0 1 1 1: Phase counting mode 4*1 Settings other than above are prohibited.	R/W
b4	BFA*2	Buffer Operation A	0: TPUm.TGRA operates normally 1: TPUm.TGRA and TPUm.TGRC used together for buffer operation (m = 0, 3)	R/W
b5	BFB*3	Buffer Operation B	0: TPUm.TGRB operates normally 1: TPUm.TGRB and TPUm.TGRD used together for buffer operation (m = 0, 3)	R/W
b6	ICSELB	TGRB Input Capture Input Select	0: Input capture input source is TIOCBn pin 1: Input capture input source is TIOCAn pin (n = 0 to 5)	R/W
b7	ICSELD*3	TGRD Input Capture Input Select	0: Input capture input source is TIOCDn pin 1: Input capture input source is TIOCCn pin (n = 0, 3)	R/W

Note 1. Phase counting mode cannot be set for TPU0 and TPU3. A 0 should be written to bit 2 for them.
Note 2. Bit 4 of TPU1, TPU2, TPU4, and TPU5 that do not have TGRC is reserved. This bit is read as 0. The write value should be 0.
Note 3. Bits 5 and 7 of TPU1, TPU2, TPU4, and TPU5 that do not have TGRD are reserved. These bits are read as 0. The write value should be 0.

TPUm.TMDR settings should be made while TPUm.TCNT operation is stopped.

BFA Bit (Buffer Operation A)

Specifies whether TPUm.TGRA (m = 0, 3) is to normally operate, or TPUm.TGRA and TPUm.TGRC (m = 0, 3) are to be used together for buffer operation.

When TGRC is used as a buffer register, TGRC input capture/output compare is not generated.

BFB Bit (Buffer Operation B)

Specifies whether TPUm.TGRB (m = 0, 3) is to normally operate, or TPUm.TGRB and TPUm.TGRD (m = 0, 3) are to be used together for buffer operation.

When TGRD is used as a buffer register, TGRD input capture/output compare is not generated.

ICSELB Bit (TGRB Input Capture Input Select)

Selects the input capture input for TPUm.TGRB (m = 0 to 5).

This function allows measurement of high-level width and period of the input pulse on a TIOCAn input pin.

ICSELD Bit (TGRD Input Capture Input Select)

Selects the input capture input for TPUm.TGRD (m = 0, 3).

This function allows measurement of high-level width and period of the input pulse on a TIOCCn input pin.

27.2.3 Timer I/O Control Register (TIORH, TIORL, TIOR)

- TPU0.TIORH, TPU1.TIOR, TPU2.TIOR, TPU3.TIORH, TPU4.TIOR, TPU5.TIOR

Address(es): TPU0.TIORH 0008 8112h, TPU1.TIOR 0008 8122h, TPU2.TIOR 0008 8132h, TPU3.TIORH 0008 8142h, TPU4.TIOR 0008 8152h, TPU5.TIOR 0008 8162h

	b7	b6	b5	b4	b3	b2	b1	b0
	IOB[3:0]				IOA[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOA[3:0]	TGRA Control	See Table 27.13 to Table 27.18.*1	R/W
b7 to b4	IOB[3:0]	TGRB Control	See Table 27.13 to Table 27.18.*1	R/W

Note 1. If the IOn[3:0] bit (n = A, B) values are changed to output disabled (0000b or 0100b) during low/high/toggle output on compare match, the TIOCAn/TIOCBn pin (n = 0 to 5) is placed in high impedance state.

- TPU0.TIORL, TPU3.TIORL

Address(es): TPU0.TIORL 0008 8113h, TPU3.TIORL 0008 8143h

	b7	b6	b5	b4	b3	b2	b1	b0
	IOD[3:0]				IOC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOC[3:0]	TGRC Control	See Table 27.19 and Table 27.20.*1	R/W
b7 to b4	IOD[3:0]	TGRD Control	See Table 27.19 and Table 27.20.*1	R/W

Note 1. If the IOn[3:0] bit (n = C, D) values are changed to output disabled (0000b or 0100b) during low/high/toggle output on compare match, the TIOCCn/TIOCDn pin (n = 0, 3) is placed in high impedance state.

TPU has two TIORH registers, one for TPU0 and TPU3, and two TIORL registers, one for TPU0 and TPU3, and also has four TIOR registers, one for TPU1, TPU2, TPU4, and TPU5. Thus the TPU has eight timer I/O control registers in total.

TIORH, TIORL, and TIOR control registers TGRA, TGRB, TGRC, and TGRD.

Note that TIORH, TIORL, and TIOR are affected by the TMDR setting. For details, see Table 27.13 to Table 27.20.

The initial output specified by TIORH, TIORL, and TIOR is valid when the counter is stopped (the TPUA.TSTR.CSTj bit (j = 0 to 5) is cleared to 0). In PWM mode 2, the output at the time when the TCNT is cleared to 0 is specified as the initial output.

When buffer operation has been selected for register TGRC or TGRD, the settings of the IOC[3:0] or IOD[3:0] bits become ineffective, and the TGRC or TGRD register simply operates as a buffer.

To specify the input capture pin in TIORH, TIORL, or TIOR, set the bit in the port direction register (PDR) for the corresponding pin to 0 (input port), and set the bit in the port mode register (PMR) to 1 (uses the pin as an I/O port for peripheral functions). For details, see section 22, I/O Ports.

IOA[3:0] Bits (TGRA Control)

Select the function of TPUm.TGRA (m = 0 to 5).

IOB[3:0] Bits (TGRB Control)

Select the function of TPUm.TGRB (m = 0 to 5).

IOC[3:0] Bits (TGRC Control)

Select the function of TPUm.TGRC (m = 0, 3).

IOD[3:0] Bits (TGRD Control)

Select the function of TPUm.TGRD (m = 0, 3).

Table 27.13 TPU0.TIORH

Bits IOA[3:0]				Description	
b3	b2	b1	b0	TPU0.TGRA Function	TIOCA0 Pin Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA0 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCA0 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCA0 pin; input capture at both edges
1	1	x	x		Capture input source is TPU1 count clock; input capture at TPU1.TCNT count-up/ count-down*1

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPU0.TGRB Function	TIOCB0 Pin Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB0 or TIOCA0 pin*2; input capture at rising edge
1	0	0	1		Capture input source is TIOCB0 or TIOCA0 pin*2; input capture at falling edge
1	0	1	x		Capture input source is TIOCB0 or TIOCA0 pin*2; input capture at both edges
1	1	x	x		Capture input source is TPU1 count clock; input capture at TPU1.TCNT count-up/ count-down*1

x: Don't care

Note 1. When the TPSC[2:0] bits in TPU1.TCR are set to 000b and PCLK/1 is used as the TPU1.TCNT count clock, this setting is invalid and input capture is not generated.

Note 2. Selected by the ICSELB bit in TPU0.TMDR.

Table 27.14 TPU1.TIOR

Bits IOA[3:0]				Description	
b3	**b2**	**b1**	**b0**	**TPU1.TGRA Function**	**TIOCA1 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA1 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCA1 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCA1 pin; input capture at both edges
1	1	x	x		Capture input source is TPU0.TGRA compare match/input capture; input capture at generation of TPU0.TGRA compare match/input capture

Bits IOB[3:0]				Description	
b7	**b6**	**b5**	**b4**	**TPU1.TGRB Function**	**TIOCB1 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB1 or TIOCA1 pin*1; input capture at rising edge
1	0	0	1		Capture input source is TIOCB1 or TIOCA1 pin*1; input capture at falling edge
1	0	1	x		Capture input source is TIOCB1 or TIOCA1 pin*1; input capture at both edges
1	1	x	x		Capture input source is TPU0.TGRC compare match/input capture; input capture at generation of TPU0.TGRC compare match/input capture

x: Don’t care

Note 1. Selected by the ICSELB bit in TPU1.TMDR.

Table 27.15 TPU2.TIOR

Bits IOA[3:0]				Description	
b3	**b2**	**b1**	**b0**	**TPU2.TGRA Function**	**TIOCA2 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCA2 pin; input capture at rising edge
1	x	0	1		Capture input source is TIOCA2 pin; input capture at falling edge
1	x	1	x		Capture input source is TIOCA2 pin; input capture at both edges

Bits IOB[3:0]				Description	
b7	**b6**	**b5**	**b4**	**TPU2.TGRB Function**	**TIOCB2 Pin (Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCB2 or TIOCA2 pin*1; input capture at rising edge
1	x	0	1		Capture input source is TIOCB2 or TIOCA2 pin*1; input capture at falling edge
1	x	1	x		Capture input source is TIOCB2 or TIOCA2 pin*1; input capture at both edges

x: Don’t care

Note 1. Selected by the ICSELB bit in TPU2.TMDR.

Table 27.16 TPU3.TIORH

Bits IOA[3:0]				Description	
b3	**b2**	**b1**	**b0**	**TPU3.TGRA Function**	**TIOCA3 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA3 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCA3 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCA3 pin; input capture at both edges
1	1	x	x		Capture input source is TPU4 count clock; input capture at TPU4.TCNT count-up/ count-down*1

Bits IOB[3:0]				Description	
b7	**b6**	**b5**	**b4**	**TPU3.TGRB Function**	**TIOCB3 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB3 or TIOCA3 pin*2; input capture at rising edge
1	0	0	1		Capture input source is TIOCB3 or TIOCA3 pin*2; input capture at falling edge
1	0	1	x		Capture input source is TIOCB3 or TIOCA3 pin*2; input capture at both edges
1	1	x	x		Capture input source is TPU4 count clock; input capture at TPU4.TCNT count-up/ count-down*1

x: Don’t care

Note 1. When the TPSC[2:0] bits in TPU4.TCR are set to 000b and PCLK/1 is used as the TPU4.TCNT count clock, this setting is invalid and input capture is not generated.

Note 2. Selected by the ICSELB bit in TPU3.TMDR.

Table 27.17 TPU4.TIOR

Bits IOA[3:0]				Description	
b3	**b2**	**b1**	**b0**	**TPU4.TGRA Function**	**TIOCA4 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA4 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCA4 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCA4 pin; input capture at both edges
1	1	x	x		Capture input source is TPU3.TGRA compare match/input capture; input capture at generation of TPU3.TGRA compare match/input capture

Bits IOB[3:0]				Description	
b7	**b6**	**b5**	**b4**	**TPU4.TGRB Function**	**TIOCB4 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB4 or TIOCA4 pin*1; input capture at rising edge
1	0	0	1		Capture input source is TIOCB4 or TIOCA4 pin*1; input capture at falling edge
1	0	1	x		Capture input source is TIOCB4 or TIOCA4 pin*1; input capture at both edges
1	1	x	x		Capture input source is TPU3.TGRC compare match/input capture; input capture at generation of TPU3.TGRC compare match/input capture

x: Don’t care

Note 1. Selected by the ICSELB bit in TPU4.TMDR.

Table 27.18 TPU5.TIOR

Bits IOA[3:0]				Description	
b3	**b2**	**b1**	**b0**	**TPU5.TGRA Function**	**TIOCA5 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCA5 pin; input capture at rising edge
1	x	0	1		Capture input source is TIOCA5 pin; input capture at falling edge
1	x	1	x		Capture input source is TIOCA5 pin; input capture at both edges

Bits IOB[3:0]				Description	
b7	**b6**	**b5**	**b4**	**TPU5.TGRB Function**	**TIOCB5 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCB5 or TIOCA5 pin*1; input capture at rising edge
1	x	0	1		Capture input source is TIOCB5 or TIOCA5 pin*1; input capture at falling edge
1	x	1	x		Capture input source is TIOCB5 or TIOCA5 pin*1; input capture at both edges

x: Don't care

Note 1. Selected by the ICSELB bit in TPU5.TMDR.

Table 27.19 TPU0.TIORL

Bits IOC[3:0]				Description	
b3	b2	b1	b0	TPU0.TGRC Function	TIOCC0 Pin Function and Related Issue
0	0	0	0	Output compare register*1	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register*1	Capture input source is TIOCC0 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCC0 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCC0 pin; input capture at both edges
1	1	x	x		Capture input source is TPU1 count clock; input capture at TPU1.TCNT count-up/ count-down*3

Bits IOD[3:0]				Description	
b7	b6	b5	b4	TPU0.TGRD Function	TIOCD0 Pin Function and Related Issue
0	0	0	0	Output compare register*2	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD0 or TIOCC0 pin*4; input capture at rising edge
1	0	0	1		Capture input source is TIOCD0 or TIOCC0 pin*4; input capture at falling edge
1	0	1	x		Capture input source is TIOCD0 or TIOCC0 pin*4; input capture at both edges
1	1	x	x		Capture input source is TPU1 count clock; input capture at TPU1.TCNT count-up/ count-down*3

x: Don't care

Note 1. When the BFA bit in TPU0.TMDR is set to 1 (TPU0.TGRA and TPU0.TGRC are used for buffer operation) and TPU0.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 2. When the BFB bit in TPU0.TMDR is set to 1 (TPU0.TGRB and TPU0.TGRD are used for buffer operation) and TPU0.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 3. When the TPSC[2:0] bits in TPU1.TCR are set to 000b and PCLK/1 is used as the TPU0.TCNT count clock, this setting is invalid and input capture is not generated.

Note 4. Selected by the ICSELD bit in TPU0.TMDR.

Table 27.20 TPU3.TIORL

Bits IOC[3:0]				Description	
b3	b2	b1	b0	TPU3.TGRC Function	TIOCC3 Pin Function and Related Issue
0	0	0	0	Output compare register*1	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register*1	Capture input source is TIOCC3 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCC3 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCC3 pin; input capture at both edges
1	1	x	x		Capture input source is TPU4 count clock; input capture at TPU4.TCNT count-up/ count-down*3

Bits IOD[3:0]				Description	
b7	b6	b5	b4	TPU3.TGRD Function	TIOCD3 Pin Function and Related Issue
0	0	0	0	Output compare register*2	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD3 or TIOCC3 pin*4; input capture at rising edge
1	0	0	1		Capture input source is TIOCD3 or TIOCC3 pin*4; input capture at falling edge
1	0	1	x		Capture input source is TIOCD3 or TIOCC3 pin*4; input capture at both edges
1	1	X	x		Capture input source is TPU4 count clock; input capture at TPU4.TCNT count-up/ count-down*3

x: Don't care

Note 1. When the BFA bit in TPU3.TMDR is set to 1 (TPU3.TGRA and TPU3.TGRC are used for buffer operation) and TPU3.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 2. When the BFB bit in TPU3.TMDR is set to 1 (TPU3.TGRB and TPU3.TGRD are used for buffer operation) and TPU3.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 3. When the TPSC[2:0] bits in TPU4.TCR are set to 000b and PCLK/1 is used as the TPU3.TCNT count clock, this setting is invalid and input capture is not generated.

Note 4. Selected by the ICSELD bit in TPU3.TMDR.

27.2.4 Timer Interrupt Enable Register (TIER)

Address(es): TPU0.TIER 0008 8114h, TPU1.TIER 0008 8124h, TPU2.TIER 0008 8134h, TPU3.TIER 0008 8144h, TPU4.TIER 0008 8154h, TPU5.TIER 0008 8164h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	—	TCIEU	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIEA	TGRA Interrupt Enable	0: Interrupt requests (TGImA) disabled 1: Interrupt requests (TGImA) enabled (m = 0 to 5)	R/W
b1	TGIEB	TGRB Interrupt Enable	0: Interrupt requests (TGImB) disabled 1: Interrupt requests (TGImB) enabled (m = 0 to 5)	R/W
b2	TGIEC*1	TGRC Interrupt Enable	0: Interrupt requests (TGImC) disabled 1: Interrupt requests (TGImC) enabled (m = 0, 3)	R/W
b3	TGIED*1	TGRD Interrupt Enable	0: Interrupt requests (TGImD) disabled 1: Interrupt requests (TGImD) enabled (m = 0, 3)	R/W
b4	TCIEV	Overflow Interrupt Enable	0: Interrupt requests (TCImV) disabled 1: Interrupt requests (TCImV) enabled (m = 0 to 5)	R/W
b5	TCIEU*2	Underflow Interrupt Enable	0: Interrupt requests (TCImU) disabled 1: Interrupt requests (TCImU) enabled (m = 1, 2, 4, 5)	R/W
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	TTGE*3	A/D Conversion Start Request Enable	0: A/D conversion start request generation disabled 1: A/D conversion start request generation enabled	R/W

Note 1. Bits 3 and 2 in TIER of TPU1, TPU2, TPU4, and TPU5 are reserved. These bits are read as 0. The write value should be 0.
Note 2. Bit 5 in TIER of TPU0 and TPU3 is reserved. This bit is read as 0. The write value should be 0.
Note 3. Bit 7 in TIER of TPU5 is reserved. This bit is read as 0. The write value should be 0.

TTGE Bit (A/D Conversion Start Request Enable)

Enables/disables generation of A/D conversion start requests by TPUm.TGRA (m = 0 to 4) input capture/compare match.

27.2.5 Timer Status Register (TSR)

Address(es): TPU0.TSR 0008 8115h, TPU1.TSR 0008 8125h, TPU2.TSR 0008 8135h, TPU3.TSR 0008 8145h, TPU4.TSR 0008 8155h, TPU5.TSR 0008 8165h

	b7	b6	b5	b4	b3	b2	b1	b0
	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGFA	Input Capture/Output Compare Flag A	0: Input capture to TPUm.TGRA or compare match with TPUm.TGRA has not occurred. 1: Input capture to TPUm.TGRA or compare match with TPUm.TGRA has occurred. (m = 0 to 5)	R/W*2
b1	TGFB	Input Capture/Output Compare Flag B	0: Input capture to TPUm.TGRB or compare match with TPUm.TGRB has not occurred. 1: Input capture to TPUm.TGRB or compare match with TPUm.TGRB has occurred. (m = 0 to 5)	R/W*2
b2	TGFC*4	Input Capture/Output Compare Flag C	0: Input capture to TPUm.TGRC or compare match with TPUm.TGRC has not occurred. 1: Input capture to TPUm.TGRC or compare match with TPUm.TGRC has occurred. (m = 0, 3)	R/W*2
b3	TGFD*4	Input Capture/Output Compare Flag D	0: Input capture to TPUm.TGRD or compare match with TPUm.TGRD has not occurred. 1: Input capture to TPUm.TGRD or compare match with TPUm.TGRD has occurred. (m = 0, 3)	R/W*2
b4	TCFV	Overflow Flag	0: TPUm.TCNT has not overflowed. 1: TPUm.TCNT has overflowed. (m = 0 to 5)	R/W*2
b5	TCFU*3	Underflow Flag	0: TPUm.TCNT has not underflowed. 1: TPUm.TCNT has underflowed. (m = 1, 2, 4, 5)	R/W*2
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	TCFD*1	Counting Direction Flag	0: TPUm.TCNT counts down. 1: TPUm.TCNT counts up. (m = 1, 2, 4, 5)	R

Note 1. Bit 7 of registers TPU0.TSR and TPU3.TSR is reserved. The bit is read as 1. The write value should be 1.
Note 2. Only writing 0 to this bit is possible; this clears the flag.
Note 3. Bit 5 of registers TPU0.TSR and TPU3.TSR is reserved. The bit is read as 0. The write value should be 0.
Note 4. Bits 2 and 3 of registers TPU1.TSR, TPU2.TSR, TPU4.TSR and TPU5.TSR are reserved. The bits are read as 0. The write value should be 0.

TGFA Flag (Input Capture/Output Compare Flag A)

This status flag indicates that input capture to TPUm.TGRA or compare match with TPUm.TGRA (m = 0 to 5) has occurred.

[Setting conditions]

- When TPUm.TGRA holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRA.
- When TPUm.TGRA is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRA.

[Clearing conditions]

- Activation of the DTC by the TGImA interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFA after reading its value as 1.

TGFB Flag (Input Capture/Output Compare Flag B)

This status flag indicates that input capture to TPUm.TGRB or compare match with TPUm.TGRB (m = 0 to 5) has occurred.

[Setting conditions]

- When TPUm.TGRB holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRB.
- When TPUm.TGRB is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRB.

[Clearing conditions]

- Activation of the DTC by the TGImB interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFB after reading its value as 1.

TGFC Flag (Input Capture/Output Compare Flag C)

This status flag indicates that input capture to TPUm.TGRC or compare match with TPUm.TGRC (m = 0, 3) has occurred.

[Setting conditions]

- When TPUm.TGRC holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRC.
- When TPUm.TGRC is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRC.

[Clearing conditions]

- Activation of the DTC by the TGImC interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFC after reading its value as 1.

TGFD Flag (Input Capture/Output Compare Flag D)

This status flag indicates that input capture to TPUm.TGRD or compare match with TPUm.TGRD (m = 0, 3) has occurred.

[Setting conditions]

- When TPUm.TGRD holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRD.
- When TPUm.TGRD is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRD.

[Clearing conditions]

- Activation of the DTC by the TGImD interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFD after reading its value as 1.

TCFV Flag (Overflow Flag)

This status flag indicates an overflow of TPUm.TCNT (m = 0 to 5).

[Setting condition]

- Overflow of the value in TPUm.TCNT (TCNT counted from FFFFh to 0000h).

[Clearing condition]

- Writing 0 to TCFV after reading its value as 1.

TCFU Flag (Underflow Flag)

This status flag indicates an underflow of TPUm.TCNT (m = 1, 2, 4, 5).

[Setting condition]

- Underflow of the value in TPUm.TCNT (TCNT counted from 0000h to FFFFh).

[Clearing condition]

- Writing 0 to TCFU after reading its value as 1.

27.2.6 Timer Counter (TCNT)

Address(es): TPU0.TCNT 0008 8116h, TPU1.TCNT 0008 8126h, TPU2.TCNT 0008 8136h, TPU3.TCNT 0008 8146h, TPU4.TCNT 0008 8156h, TPU5.TCNT 0008 8166h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TPUm.TCNT is a readable/writable counter that counts the internal clock or external events.

27.2.7 Timer General Register A (TGRA)
Timer General Register B (TGRB)
Timer General Register C (TGRC)
Timer General Register D (TGRD)

Address(es): TPU0.TGRA 0008 8118h, TPU0.TGRB 0008 811Ah, TPU0.TGRC 0008 811Ch, TPU0.TGRD 0008 811Eh, TPU1.TGRA 0008 8128h, TPU1.TGRB 0008 812Ah, TPU2.TGRA 0008 8138h, TPU2.TGRB 0008 813Ah, TPU3.TGRA 0008 8148h, TPU3.TGRB 0008 814Ah, TPU3.TGRC 0008 814Ch, TPU3.TGRD 0008 814Eh, TPU4.TGRA 0008 8158h, TPU4.TGRB 0008 815Ah, TPU5.TGRA 0008 8168h, TPU5.TGRB 0008 816Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TPU has 16 TGR registers in total, four each for TPU0 and TPU3, and two each for TPU1, TPU2, TPU4, and TPU5. TPUm.TGRA (m = 0 to 5), TPUm.TGRB (m = 0 to 5), TPUm.TGRC (m = 0, 3), and TPUm.TGRD (m = 0, 3) are readable/writable registers with a dual function as output compare and input capture registers.
TPUm.TGRC and TPUm.TGRD can also be specified for operation as buffer registers. Register combinations during buffer operations are TPUm.TGRA—TPUm.TGRC and TPUm.TGRB—TPUm.TGRD.

27.2.8 Timer Start Register (TSTR)

Address(es): TPUA.TSTR 0008 8100h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CST5	CST4	CST3	CST2	CST1	CST0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CST0	Counter Start 0	0: TCNT count operation is stopped 1: TCNT performs count operation	R/W
b1	CST1	Counter Start 1		R/W
b2	CST2	Counter Start 2		R/W
b3	CST3	Counter Start 3		R/W
b4	CST4	Counter Start 4		R/W
b5	CST5	Counter Start 5		R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TPUA.TSTR starts or stops TCNT operation for TPU0 to TPU5.
Before setting the operating mode in TPUm.TMDR or setting the TPUm.TCNT count clock in TPUm.TCR, stop the TPUm.TCNT operation.

CSTn Bit (Counter Start n) (n = 0 to 5)

This bit starts or stop the TCNT.
When the CSTn bit is cleared to 0 with CSTn = 1 and the corresponding TIOCyn pin (y = A to D; n = 0 to 5) specified for output, count operation stops but the output compare output level of the corresponding TIOCyn pin is retained.
If TIORH, TIORL, or TIOR is written to when the CSTn bit is 0, the pin output level will be changed to the set initial output value.

27.2.9 Timer Synchronous Register (TSYR)

Address(es): TPUA.TSYR 0008 8101h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SYNC0	Timer Synchronization 0	0: TCNT operates independently (TCNT setting/clearing is unrelated to other channels) 1: TCNT performs synchronous operation*1 (TCNT synchronous setting/synchronous clearing is possible)	R/W
b1	SYNC1	Timer Synchronization 1		R/W
b2	SYNC2	Timer Synchronization 2		R/W
b3	SYNC3	Timer Synchronization 3		R/W
b4	SYNC4	Timer Synchronization 4		R/W
b5	SYNC5	Timer Synchronization 5		R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To set synchronous operation, the SYNCn bit (n = 0 to 5) for at least two channels must be set to 1. To set synchronous clearing, the TCNT clearing source must also be set by the TCR.CCLR[2:0] bits in addition to the SYNCn bit.

TPUA.TSYR selects independent operation or synchronous operation for TCNT of TPU0 to TPU5.

SYNCn Bit (Timer Synchronization n) (n = 0 to 5)

This bit selects whether the TCNT operation is independent of or synchronized with TCNT of other channels. When synchronous operation is selected, synchronous setting of multiple TCNT and synchronous clearing through counter clearing on another channel are possible.

27.2.10 Noise Filter Control Register (NFCR)

Address(es): TPU0.NFCR 0008 8108h, TPU1.NFCR 0008 8109h, TPU2.NFCR 0008 810Ah, TPU3.NFCR 0008 810Bh, TPU4.NFCR 0008 810Ch, TPU5.NFCR 0008 810Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFCS[1:0]		NFDEN	NFCEN	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFAEN	Noise Filter Enable A	0: The noise filter for TIOCAm is disabled. 1: The noise filter for TIOCAm is enabled. (m = 0 to 5)	R/W
b1	NFBEN	Noise Filter Enable B	0: The noise filter for TIOCBm is disabled. 1: The noise filter for TIOCBm is enabled. (m = 0 to 5)	R/W
b2	NFCEN*1	Noise Filter Enable C	0: The noise filter for TIOCCm is disabled. 1: The noise filter for TIOCCm is enabled. (m = 0, 3)	R/W
b3	NFDEN*1	Noise Filter Enable D	0: The noise filter for TIOCDm is disabled. 1: The noise filter for TIOCDm is enabled. (m = 0, 3)	R/W
b5, b4	NFCS[1:0]	Noise Filter Clock Select	00: PCLK/1 01: PCLK/8 10: PCLK/32 11: Clock source that drives counting	R/W
b7, b6	—	Reserved	These bits are read as 0. Writing to these bits is not possible.	R

Note 1. Bits 2 and 3 of TPU1.NFCR, TPU2.NFCR, TPU4.NFCR, and TPU5.NFCR are reserved. The bits are read as 0. Writing to these bits is not possible.

Only set the TPUm.NFCR registers while the TPUm.TCNT is stopped.

NFAEN Bit (Noise Filter Enable A)

This bit disables or enables the noise filter for the TIOCAm pin (m = 0 to 5).

Since unexpected edges may be internally generated when the value of NFAEN is changed, select the output compare function in the timer I/O control register before changing the NFAEN value.

NFBEN Bit (Noise Filter Enable B)

This bit disables or enables the noise filter for the TIOCBm pin (m = 0 to 5).

Since unexpected edges may be internally generated when the value of NFBEN is changed, select the output compare function in the timer I/O control register before changing the NFBEN value.

NFCEN Bit (Noise Filter Enable C)

This bit disables or enables the noise filter for the TIOCCm pin (m = 0, 3).

Since unexpected edges may be internally generated when the value of NFCEN is changed, select the output compare function in the timer I/O control register before changing the NFCEN value.

NFDEN Bit (Noise Filter Enable D)

This bit disables or enables the noise filter for the TIOCDm pin (m = 0, 3).

Since unexpected edges may be internally generated when the value of NFDEN is changed, select the output compare function in the timer I/O control register before changing the NFDEN value.

NFCS[1:0] Bits (Noise Filter Clock Select)

These bits select the sampling clock for the noise filter.

When the count source is selected with NFCS[1:0] bits set to 11b, the clock that can be used as sampling clock are the internal clocks other than PCLK/1 specified with the TPSC[2:0] bits and the external clock. To select the PCLK/1 as both the count clock and the sampling clock, set the NFCS[1:0] bits to 00b.

The input-capture signal is sampled on rising edges of the selected clock signal. If the sampled levels match three times in a row, the given level is passed through as the input-capture signal. If the levels do not match, the existing value is retained.

After setting the NFCS[1:0] bits, wait for two selected sampling periods before setting the input capture function.

27.3 Operation

27.3.1 Basic Functions

Each channel has a TPUm.TCNT and a TPUm.TGRy register (y = A to D).
TCNT is a 16-bit up-counter, which can function as a free-running counter, periodic counter, or event counter.
TGRy can be used as an input capture register or output compare register.

(1) Counter Operation

When the CSTj bit (j = 0 to 5) in TPUA.TSTR is set to 1, the TCNT for the corresponding channel starts counting.

(a) Example of count operation setting procedure

Figure 27.2 shows an example of the count operation setting procedure.

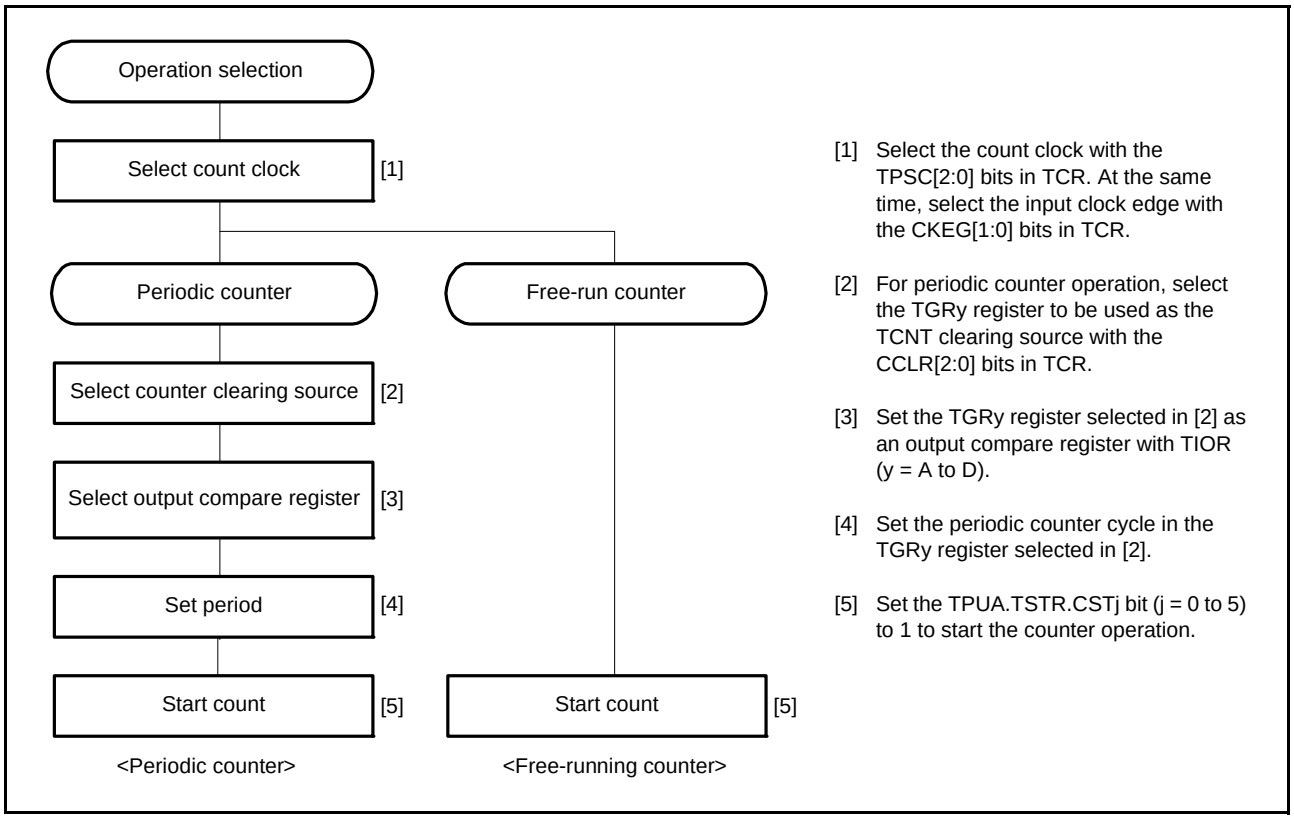


Figure 27.2 Example of Counter Operation Setting Procedure

(b) Free-running count operation and periodic count operation

Immediately after a reset, TPUm.TCNT are all set as free-running counters. When the relevant bit in TPUA.TSTR is set to 1, the corresponding TCNT starts up-count operation as a free-running counter. When TCNT overflows (changes from FFFFh to 0000h), the TPU requests an interrupt. After an overflow, TCNT restarts counting up from 0000h.
Figure 27.3 shows free-running counter operation.

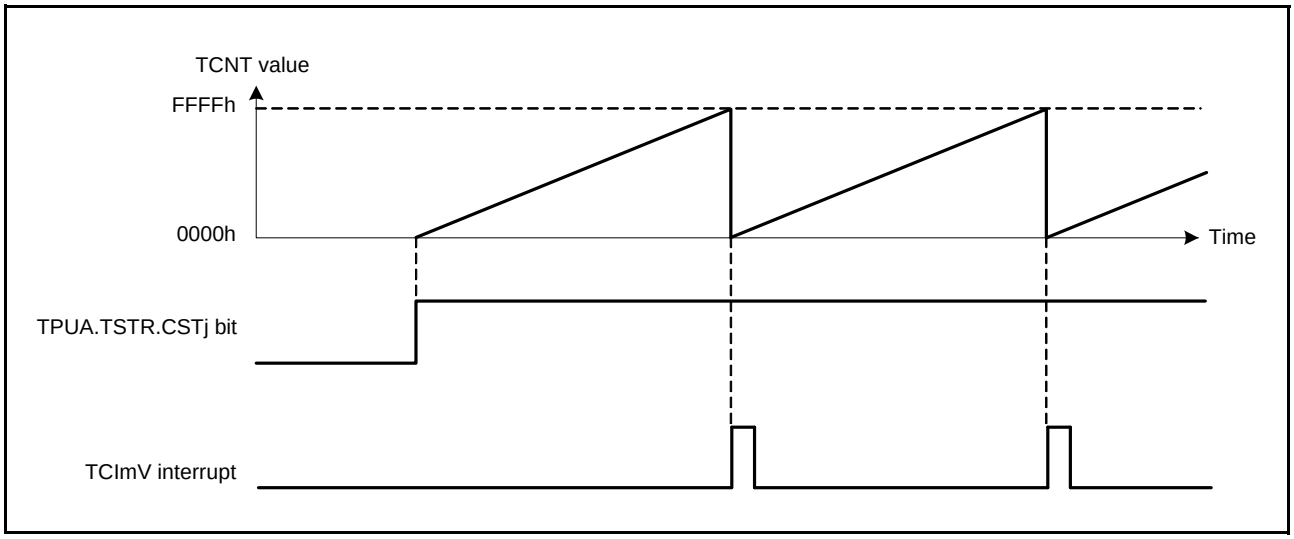


Figure 27.3 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT for the relevant channel performs periodic count operation. The TPUm.TGRy for setting the period is set as an output compare register, and counter clearing by compare match is selected by the TPUm.TCR.CCLR[2:0] bits. After the settings have been made, TCNT starts count-up operation as a periodic counter when the corresponding bit in TPUA.TSTR is set to 1. When the count value matches the TGRy value, TCNT is cleared to 0000h.
At this time, the TPU requests an interrupt. After a compare match, TCNT restarts counting up from 0000h.
Figure 27.4 shows periodic counter operation.

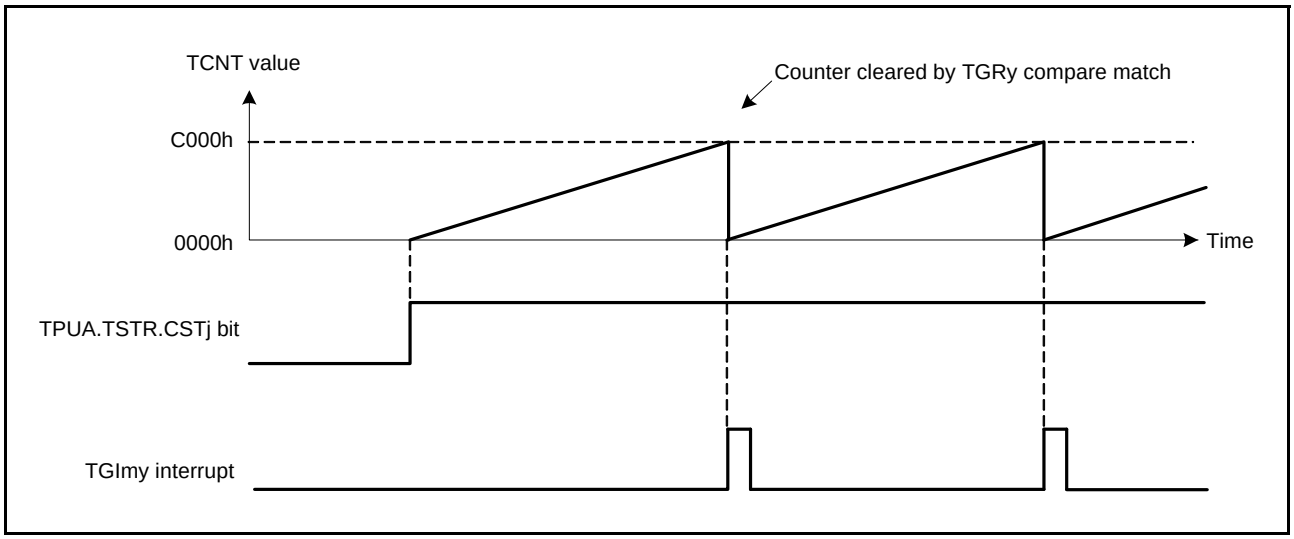


Figure 27.4 Periodic Counter Operation

(2) Waveform Output by Compare Match

The TPU can perform low, high, or toggle output from the corresponding output pin using a compare match.

(a) Example of setting procedure for waveform output by compare match

Figure 27.5 shows an example of the setting procedure for waveform output by a compare match.

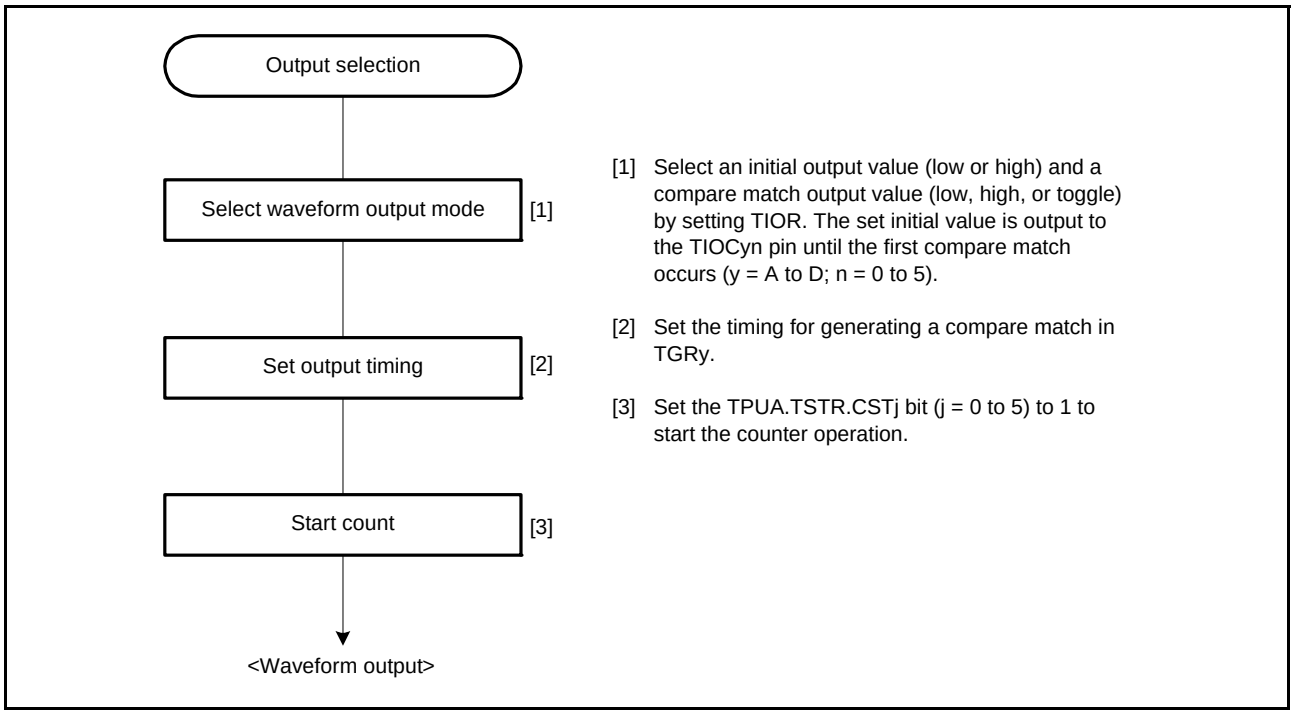


Figure 27.5 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of waveform output operation

Figure 27.6 shows an example of low output/high output.

In this example, TPUm.TCNT has been set as a free-running counter, and settings have been made so that high is output by compare match A and low is output by compare match B. When the set level and the pin level match, the pin level does not change.

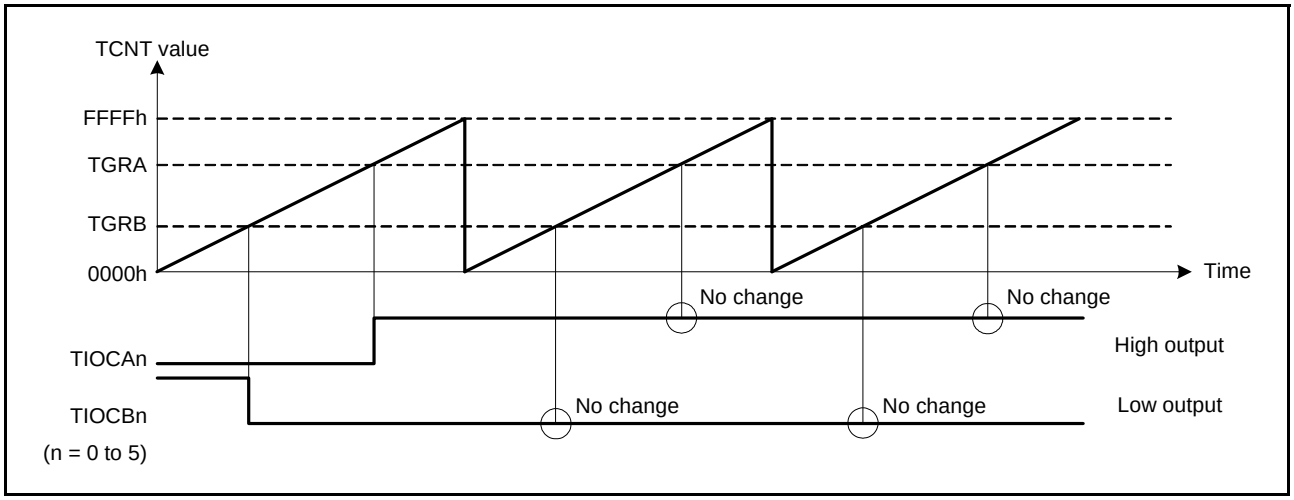


Figure 27.6 Example of Low-Output/High-Output Operation

Figure 27.7 shows an example of toggle output.
In this example, TPUm.TCNT has been set as a periodic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by both compare match A and compare match B.

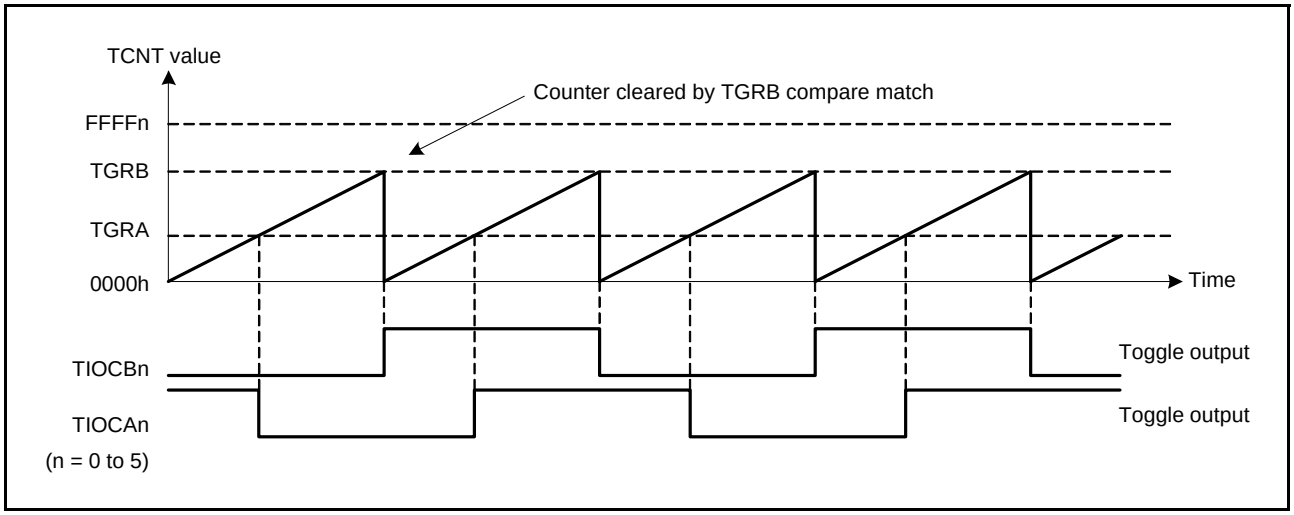


Figure 27.7 Example of Toggle Output Operation

(3) Input Capture Function

The TPUm.TCNT value can be transferred to TPUm.TGRy on detection of the TIOCyn pin (y = A to D; n = 0 to 5) input edge.
The rising edge, the falling edge, or both edges can be selected as the detection edge. It is also possible to specify the count clock or compare match signal of TPU0, TPU1, TPU3, and TPU4 as the input capture source. Noise filtering can be applied to the input capture input.

Note: Even if the counter is halted, an input capture is generated, and flag and interrupt signals are generated.
Note: When another channel's count clock is used as the input capture input for TPU0 and TPU3, PCLK/1 should not be selected as the count clock used for input capture input. Input capture will not be generated if PCLK/1 is selected.

(a) Example of setting procedure for input capture operation

Figure 27.8 shows an example of the setting procedure for input capture operation.

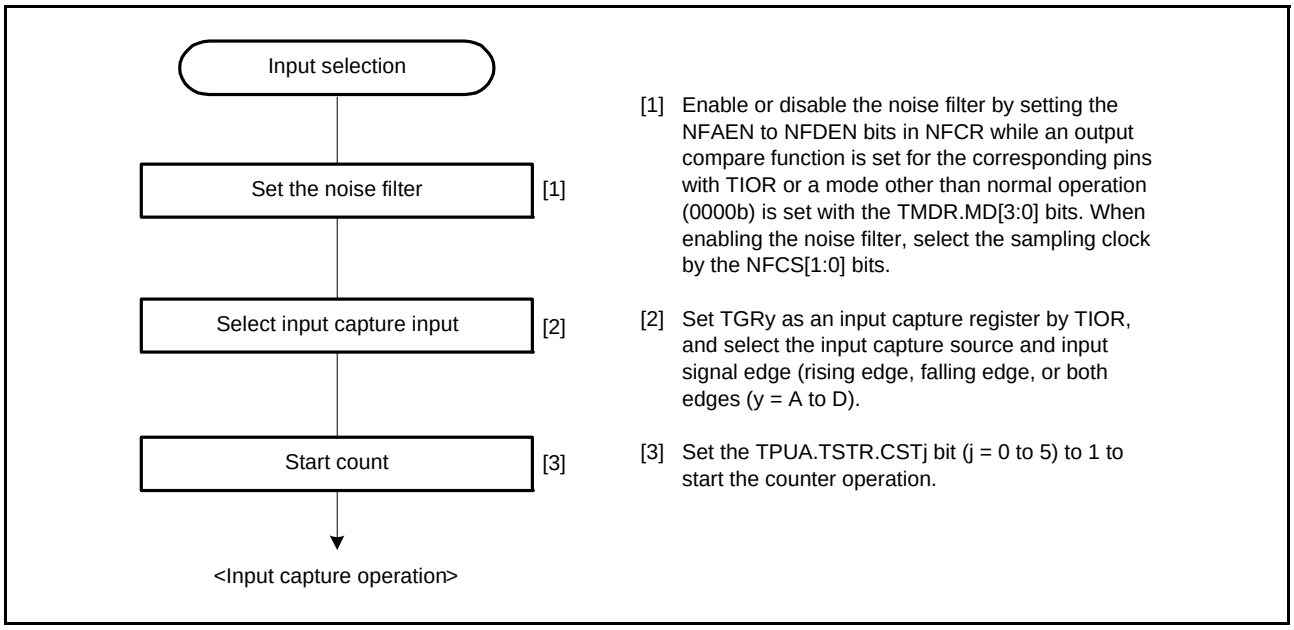


Figure 27.8 Example of Setting Procedure for Input Capture Operation

(b) Example of input capture operation

Figure 27.9 shows an example of input capture operation when the noise filter is stopped.
In this example, both rising and falling edges have been selected as the TIOCAn pin input capture input edge, the falling edge has been selected as the TIOCBn pin input capture input edge, and counter clearing by TPUm.TGRB input capture has been set for TPUm.TCNT.

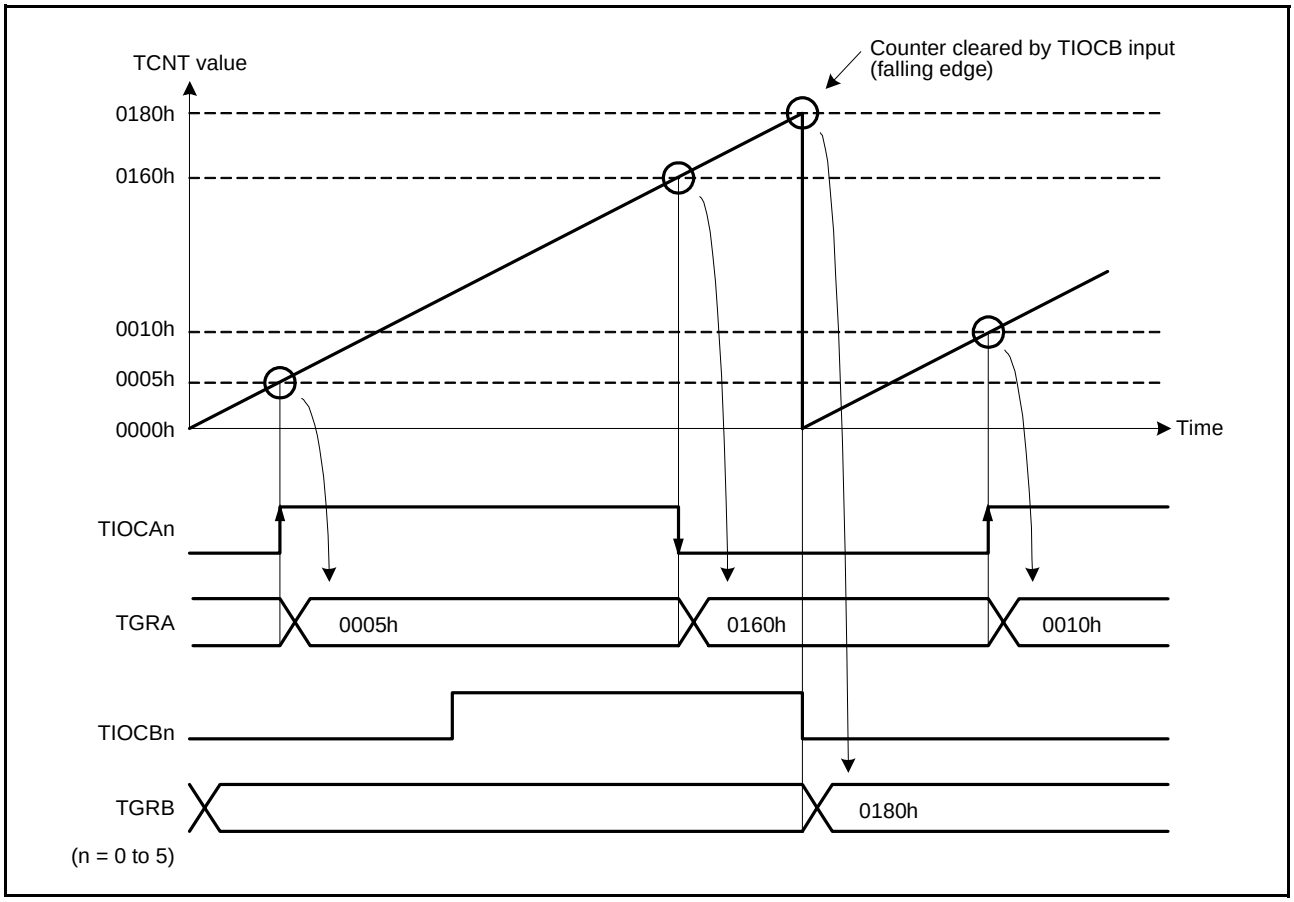


Figure 27.9 Example of Input Capture Operation (with Noise Filter Stopped)

When noise filtering is enabled, see Figure 27.30.

27.3.2 Synchronous Operation

In synchronous operation, the values in multiple TPUm.TCNT can be rewritten simultaneously (synchronous setting). Also, multiple TCNT can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TPUm.TCR.
Synchronous operation enables TPUm.TGRy to be incremented with respect to a single time base.
TPU0 to TPU5 can all be set for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 27.10 shows an example of the synchronous operation setting procedure.

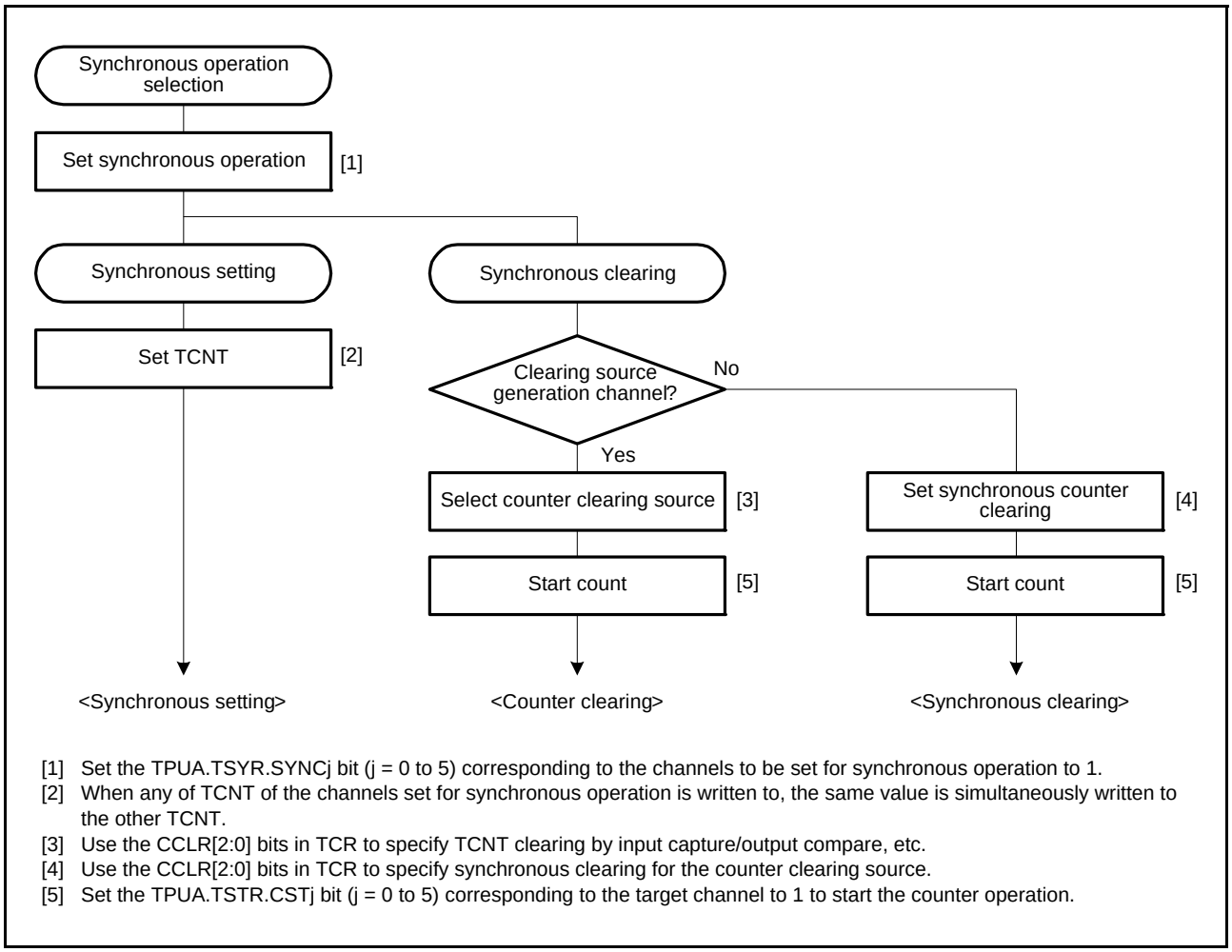


Figure 27.10 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 27.11 shows an example of synchronous operation.
In this example, synchronous operation and PWM mode 1 have been set for TPU0 to TPU2, TPU0.TGRB compare match has been set as the TPU0 counter clearing source, and synchronous clearing has been set for the TPU1 and TPU2 counter clearing source.
Three-phase PWM waveforms are output from pins TIOCA0, TIOCA1, and TIOCA2. At this time, synchronous setting and synchronous clearing by TPU0.TGRB compare match are performed for TPUm.TCNT of TPU0 to TPU2, and the data set in TPU0.TGRB is used as the PWM cycle.
For details on PWM modes, see section 27.3.5, PWM Modes.

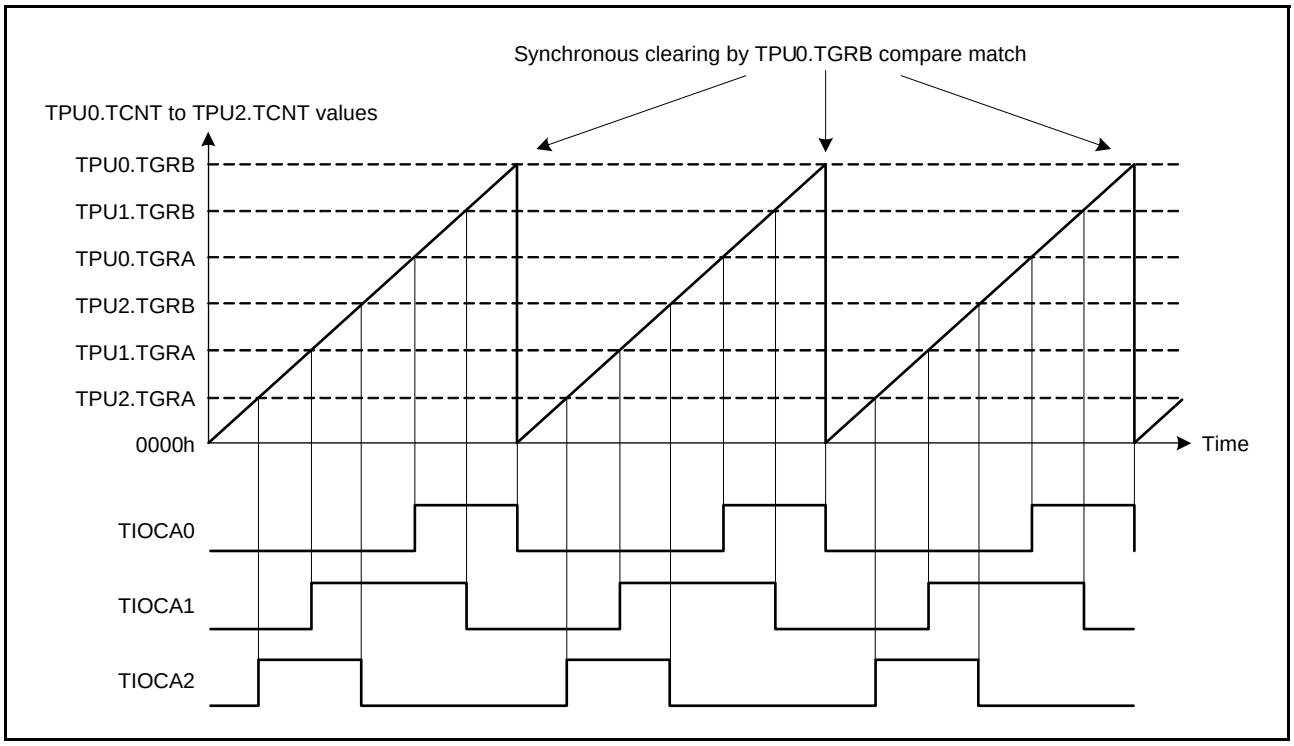


Figure 27.11 Example of Synchronous Operation

27.3.3 Buffer Operation

Buffer operation, provided for TPU0 and TPU3, enables TPUm.TGRC and TPUm.TGRD to be used as buffer registers. Buffer operation differs depending on whether TPUm.TGRy has been set as an input capture register or a compare match register.

Table 27.21 lists the register combinations used in buffer operation.

Table 27.21 Register Combinations

Channel	Timer General Register	Buffer Register
TPU0	TPU0.TGRA	TPU0.TGRC
	TPU0.TGRB	TPU0.TGRD
TPU3	TPU3.TGRA	TPU3.TGRC
	TPU3.TGRB	TPU3.TGRD

- When TPUm.TGRy is an output compare register

When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.

This operation is shown in Figure 27.12.

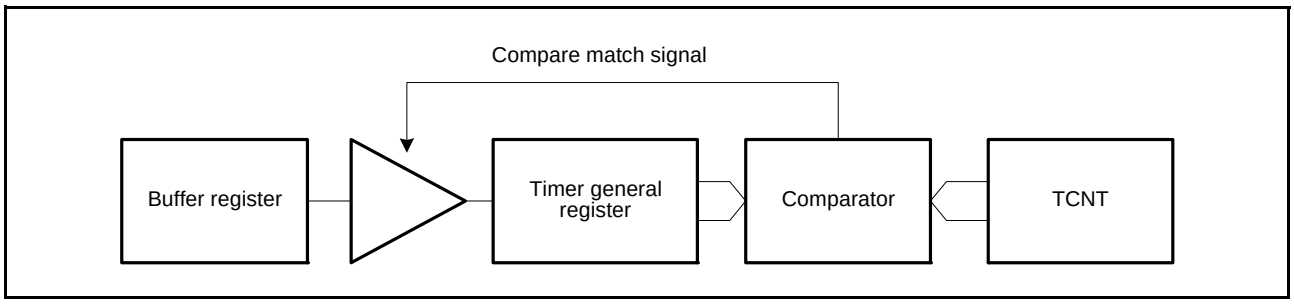


Figure 27.12 Compare Match Buffer Operation

- When TPUm.TGRy is an input capture register

When input capture occurs, the value in TPUm.TCNT is transferred to TGRy and the value previously held in TGRy is simultaneously transferred to the buffer register.

This operation is shown in Figure 27.13.

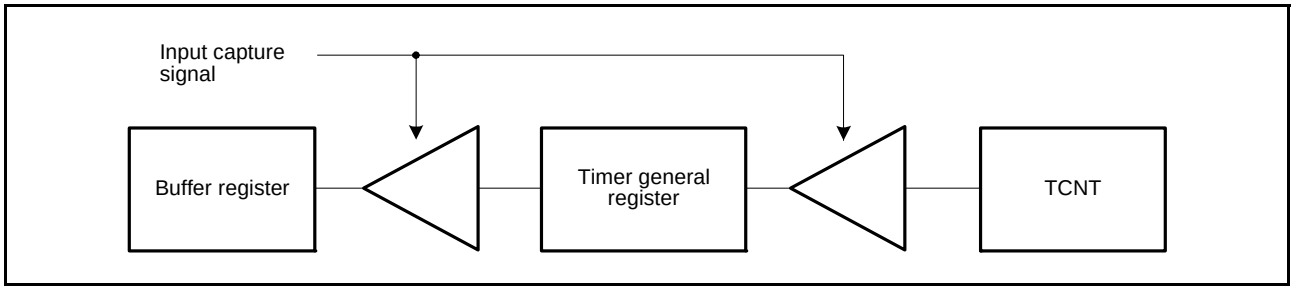


Figure 27.13 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 27.14 shows an example of the buffer operation setting procedure.

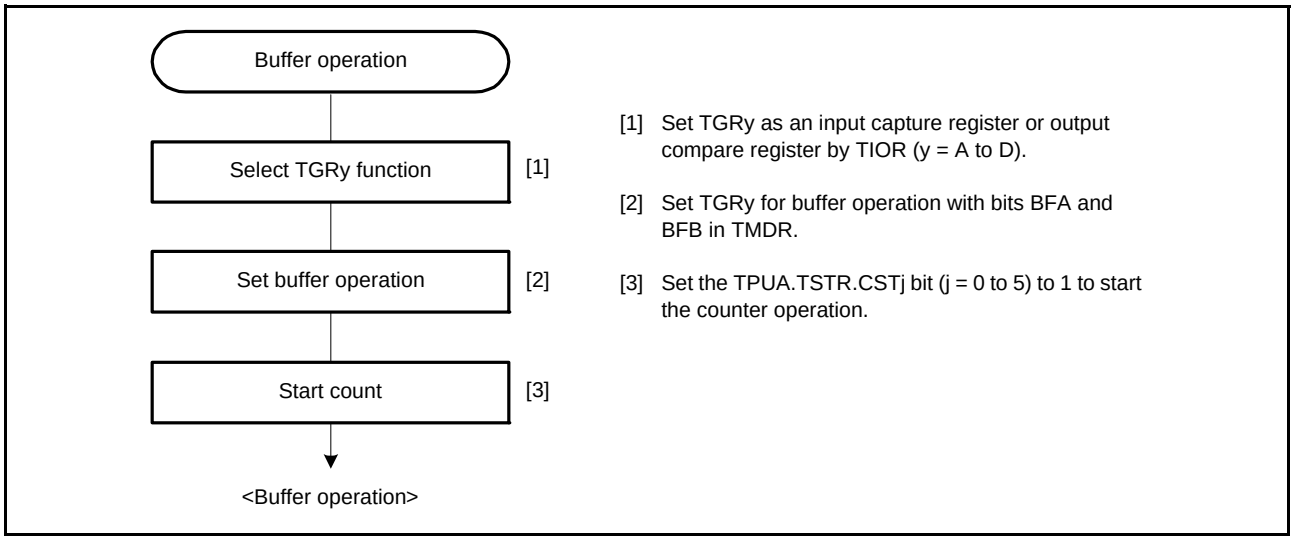


Figure 27.14 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TPUm.TGRy is an output compare register

Figure 27.15 shows an operation example in which PWM mode 1 has been set for TPU0, and buffer operation has been set for TPU0.TGRA and TPU0.TGRC. The settings used in this example are TPU0.TCNT clearing by compare match B, high output at compare match A, and low output at compare match B.

As buffer operation has been set, when compare match A occurs, the output changes and the TPU0.TGRC value is simultaneously transferred to TPU0.TGRA. This operation is repeated each time compare match A occurs.

For details on PWM modes, see section 27.3.5, PWM Modes.

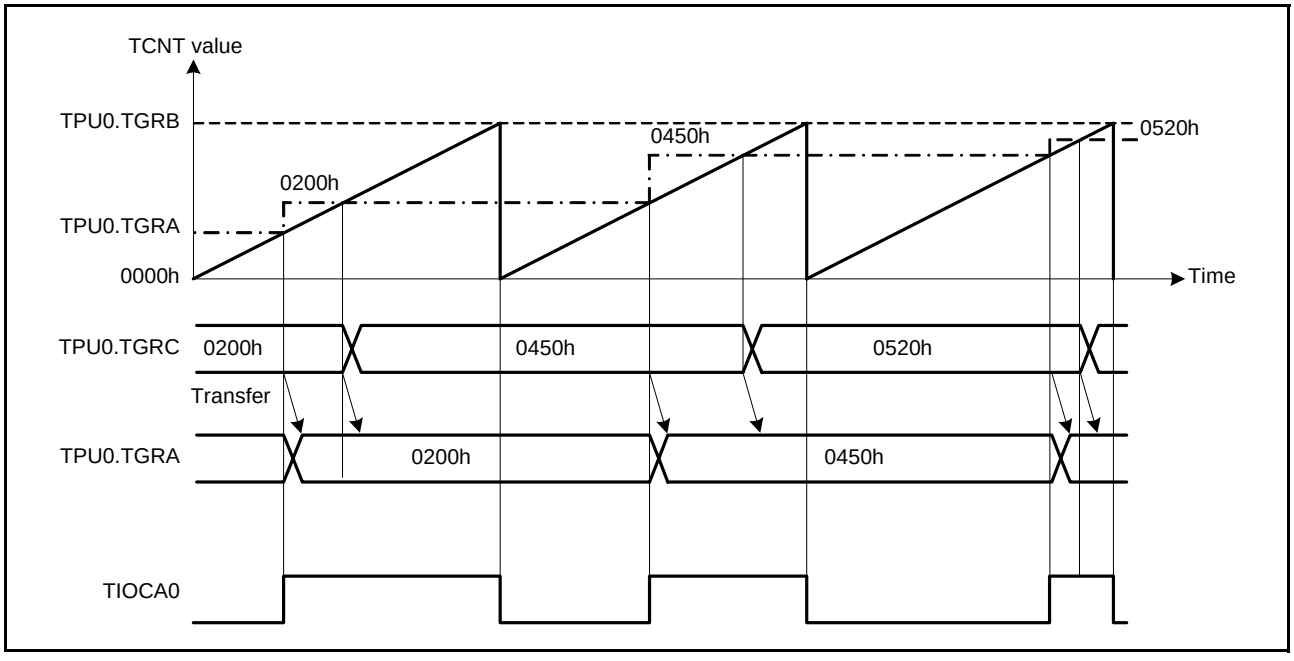


Figure 27.15 Example of Buffer Operation (1)

(b) When TPUm.TGRy is an input capture register

Figure 27.16 shows an operation example in which TPUm.TGRA has been set as an input capture register, and buffer operation has been set for the TGRA register and TPUm.TGRC.
Counter clearing by TGRA input capture has been set for TPUm.TCNT, and both rising and falling edges have been selected as the TIOCAn pin input capture input edge.
As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

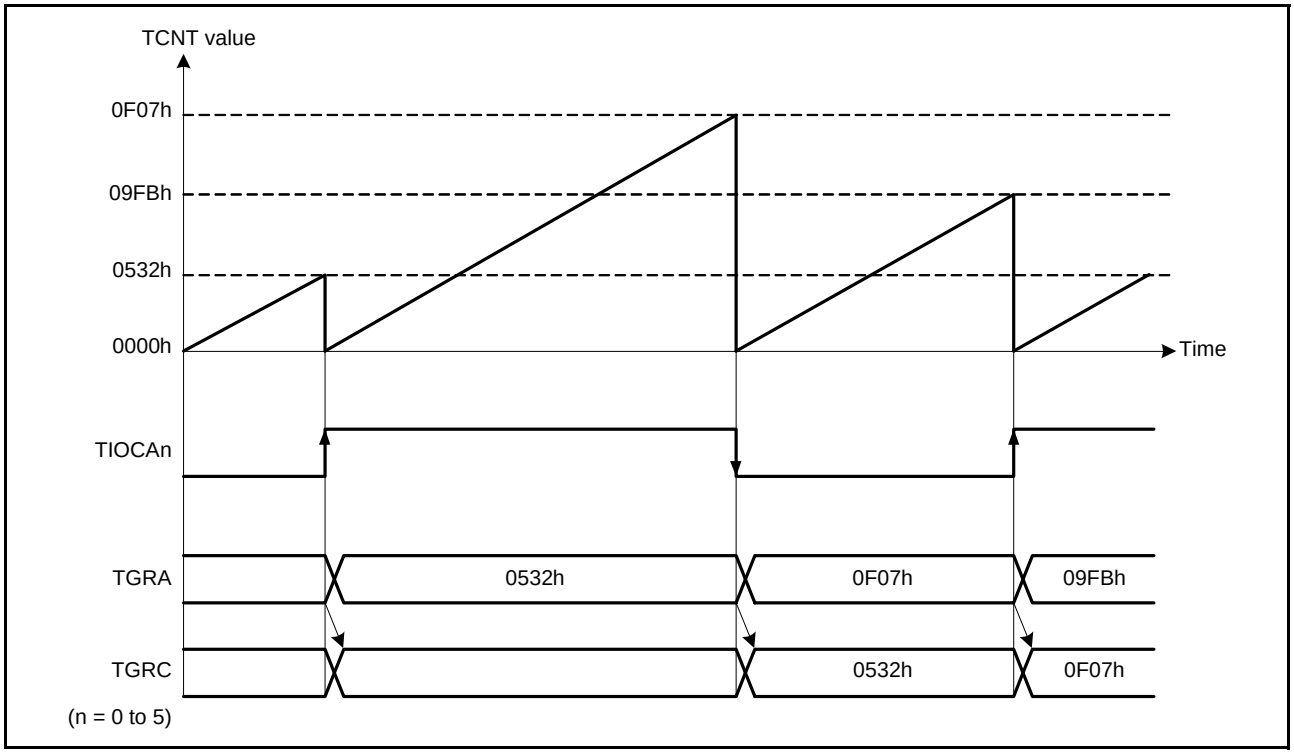


Figure 27.16 Example of Buffer Operation (2)

27.3.4 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.
This function works by counting the TPU1 (TPU4) count clock at overflow/underflow of TPU2.TCNT (TPU5.TCNT) as set by the TPSC[2:0] bits in TPU1.TCR (TPSC[2:0] bits in TPU4.TCR).
Underflow occurs only when the lower 16-bit TPUm.TCNT is in phase counting mode.
Table 27.22 lists the register combinations used in cascaded operation.

Note: When phase counting mode is set for TPU1 or TPU4, the count clock setting is invalid and the counter operates independently in phase counting mode.

Table 27.22 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
TPU1 and TPU2	TPU1.TCNT	TPU2.TCNT
TPU4 and TPU5	TPU4.TCNT	TPU5.TCNT

(1) Example of Cascaded Operation Setting Procedure

Figure 27.17 shows an example of the setting procedure for cascaded operation.

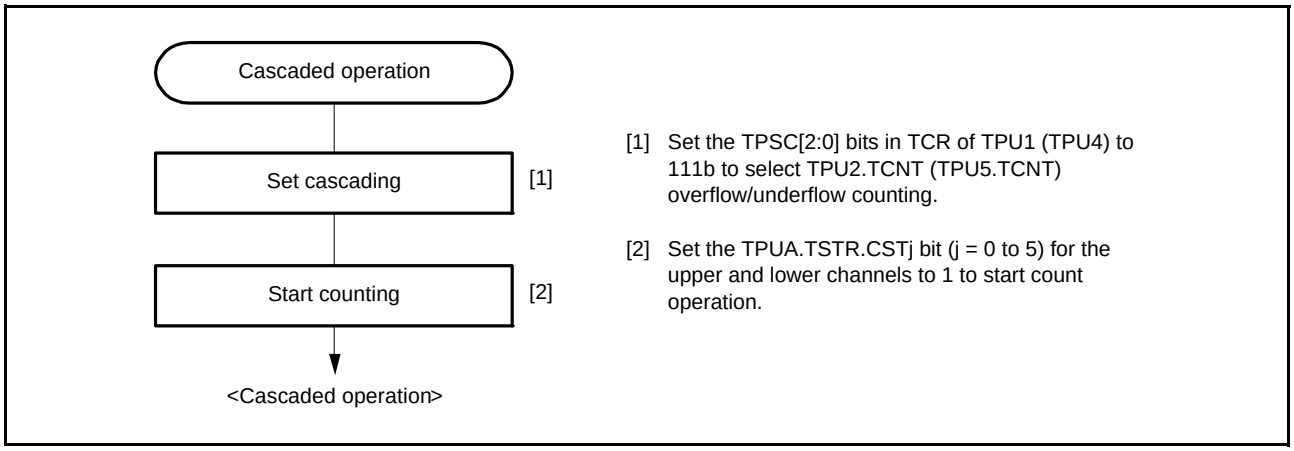


Figure 27.17 Cascaded Operation Setting Procedure

(2) Examples of Cascaded Operation

Figure 27.18 shows the operation when counting upon TPU2.TCNT overflow/underflow has been set for TPU1.TCNT, TPU1.TGRA and TPU2.TGRA have been set as input capture registers, and the rising edge of the TIOCA1 and TIOCA2 pins has been selected.

When a rising edge is input to the TIOCA1 and TIOCA2 pins simultaneously, the upper 16 bits of the 32-bit data are transferred to TPU1.TGRA, and the lower 16 bits to TPU2.TGRA.

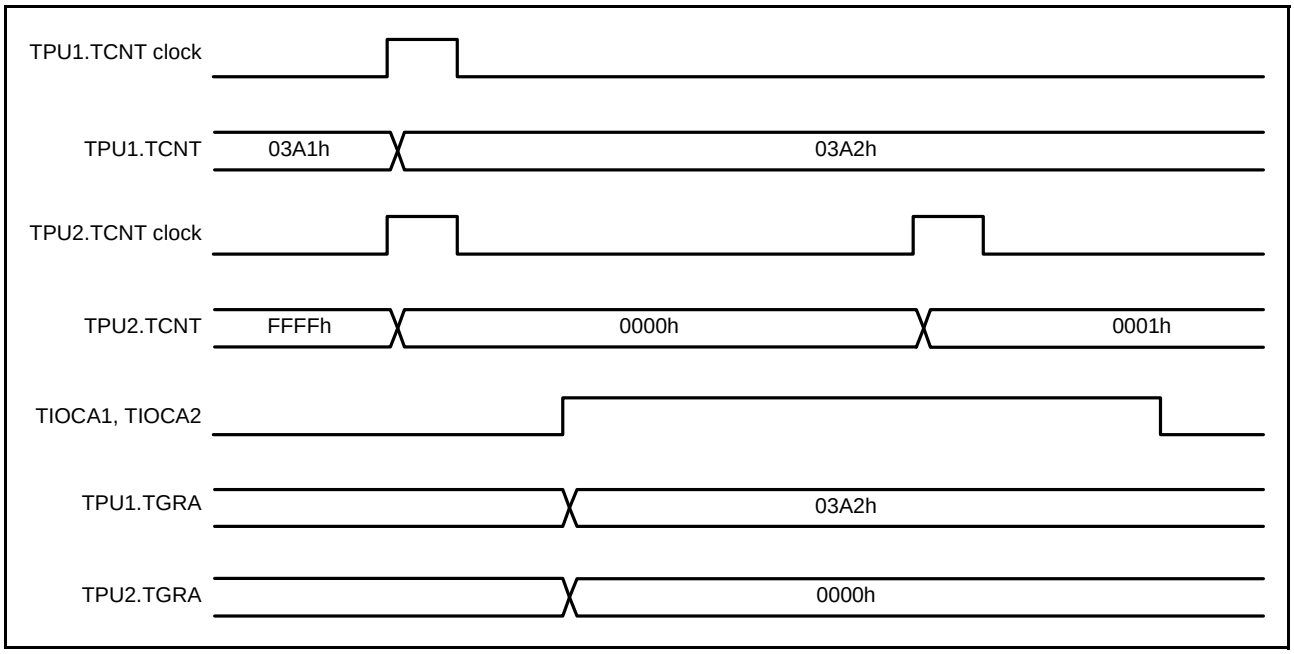


Figure 27.18 Example of Cascaded Operation (1)

Figure 27.19 shows the operation when counting upon TPU2.TCNT overflow/underflow has been set for TPU1.TCNT, and phase counting mode 1 has been specified for TPU2.

TPU1.TCNT is incremented by TPU2.TCNT overflow and decremented by TPU2.TCNT underflow.

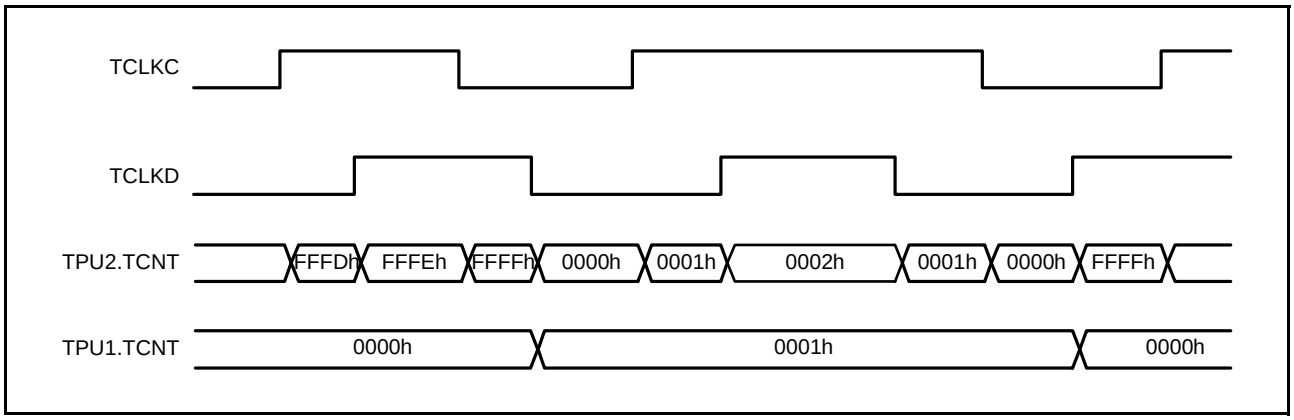


Figure 27.19 Example of Cascaded Operation (2)

27.3.5 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. low, high, or toggle output can be selected as the output level in response to compare match of each TPUm.TGRy.

Settings of TGRy registers can output a PWM waveform in the range of 0% to 100% duty cycle.

Specifying TGRy compare match as the counter clearing source enables the cycle to be set in that register. All channels can be set for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

1. PWM mode 1

   PWM waveform is generated from the TIOCAn and TIOCCn pins by pairing TPUm.TGRA with TPUm.TGRB and TPUm.TGRC with TPUm.TGRD. The outputs specified by the IOA[3:0] bits in TPUm.TIOR(H) and IOC[3:0] bits in TPUm.TIORL are output from the TIOCAn and TIOCCn pins at compare matches A and C, respectively. The outputs specified by the IOB[3:0] bits in TPUm.TIOR(H) and IOD[3:0] bits in TPUm.TIORL are output from the TIOCAn and TIOCCn pins at compare matches B and D, respectively. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRy registers are identical, the output value does not change even when a compare match occurs.

   In PWM mode 1, a maximum 8-phase PWM output is possible.

2. PWM mode 2

   PWM waveform is generated by using one TPUm.TGRy as the cycle register and the others as duty cycle registers. The output specified in TPUm.TIORH, TPUm.TIORL, or TPUm.TIOR is performed by compare matches. Upon counter clearing by a synchronous register compare match, the output value of each pin is the initial value set in TIORH, TIORL, or TIOR. If the set values of the cycle register and duty cycle register are identical, the output value does not change even when a compare match occurs.

   In PWM mode 2, a maximum 15-phase PWM waveform is possible by combined use with synchronous operation.

The correspondence between PWM output pins and registers is listed in Table 27.23.

Table 27.23 PWM Output Registers and Output Pins

Channel	Register	Output Pin	
		PWM Mode 1	PWM Mode 2
TPU0	TPU0.TGRA	TIOCA0	TIOCA0
	TPU0.TGRB		TIOCB0
	TPU0.TGRC	TIOCC0	TIOCC0
	TPU0.TGRD		TIOCD0
TPU1	TPU1.TGRA	TIOCA1	TIOCA1
	TPU1.TGRB		TIOCB1
TPU2	TPU2.TGRA	TIOCA2	TIOCA2
	TPU2.TGRB		TIOCB2
TPU3	TPU3.TGRA	TIOCA3	TIOCA3
	TPU3.TGRB		TIOCB3
	TPU3.TGRC	TIOCC3	TIOCC3
	TPU3.TGRD		TIOCD3
TPU4	TPU4.TGRA	TIOCA4	TIOCA4
	TPU4.TGRB		TIOCB4
TPU5	TPU5.TGRA	TIOCA5	TIOCA5
	TPU5.TGRB		TIOCB5

Note: In PWM mode 2, PWM waveform output is not possible for the TPUm.TGRy register in which the cycle is set.

(1) Example of PWM Mode Setting Procedure

Figure 27.20 shows an example of the PWM mode setting procedure.

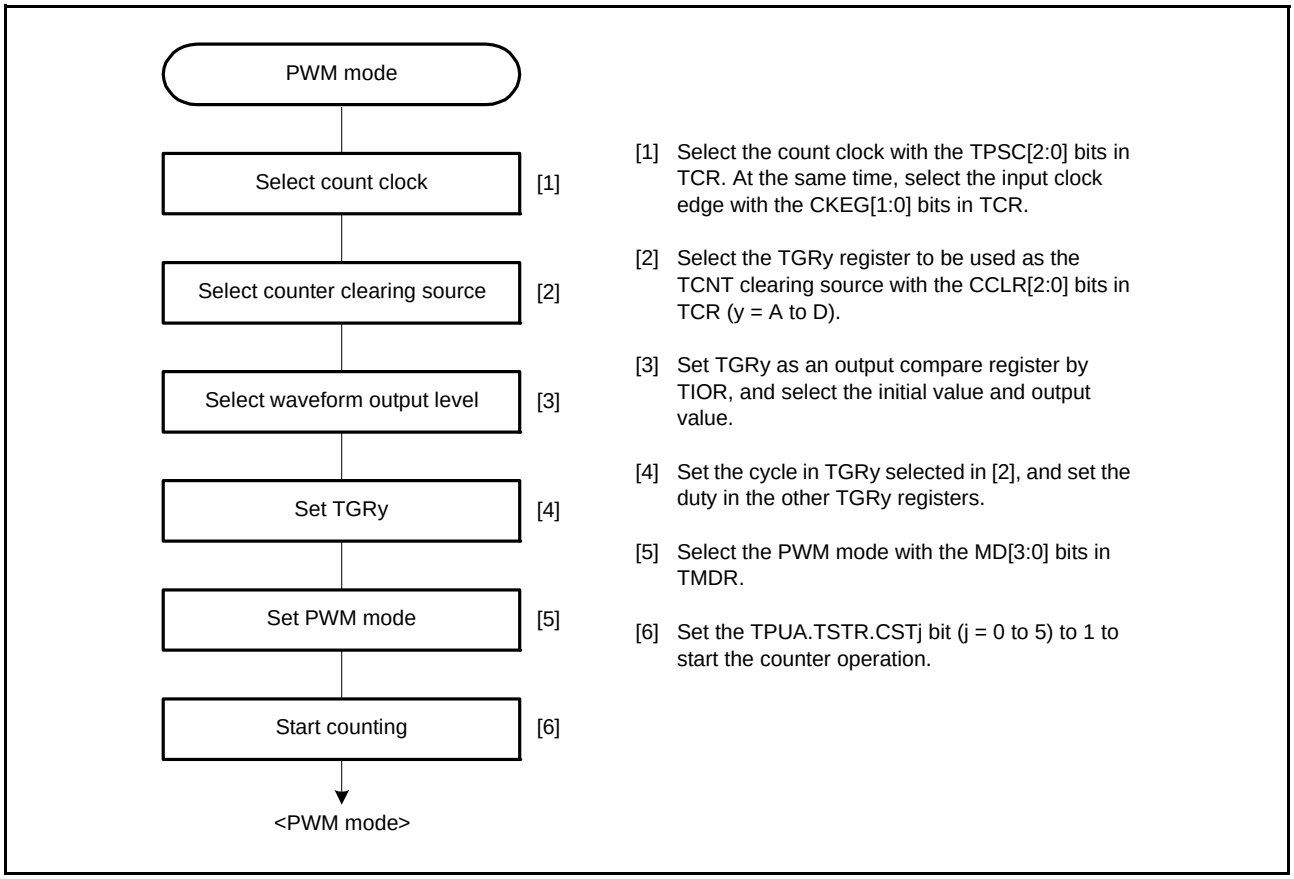


Figure 27.20 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 27.21 shows an example of PWM mode 1 operation.
In this example, TPUm.TGRA compare match is set as the TPUm.TCNT clearing source, low is set for the TGRA initial output value and output value, and high is set as the TPUm.TGRB output value.
In this case, the value set in TGRA is used as the cycle, and the value set in TGRB is used as the duty cycle.

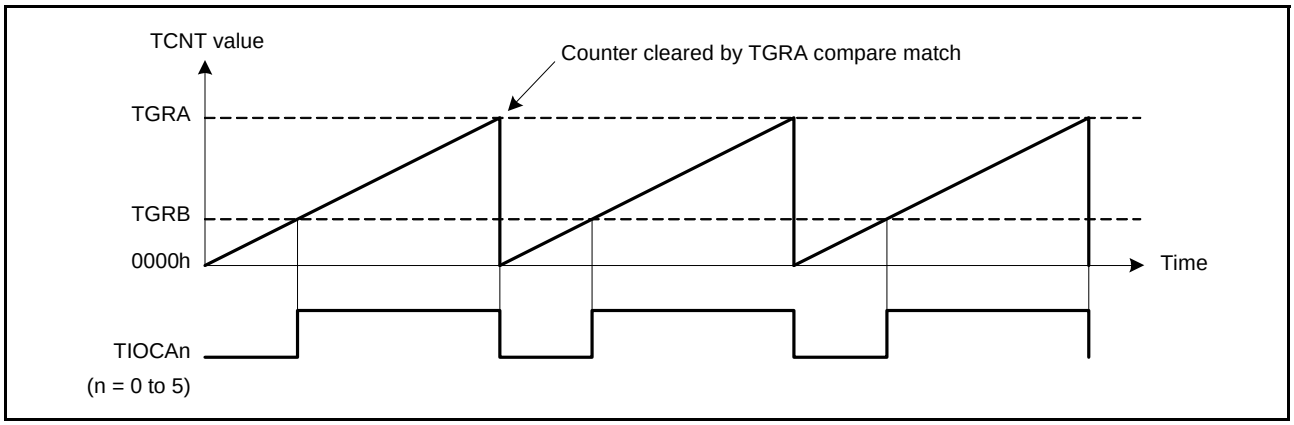


Figure 27.21 Example of PWM Mode Operation (1)

Figure 27.22 shows an example of PWM mode 2 operation.
In this example, synchronous operation is specified for TPU0 and TPU1, TPU1.TGRB compare match is set as the TPUm.TCNT clearing source, and low is set for the initial output value and high for the output value of the other TPUm.TGRy registers (TPU0.TGRA to TPU0.TGRD and TPU1.TGRA), to output a 5-phase PWM waveform.
In this case, the value set in TPU1.TGRB is used as the cycle, and the values set in the other TGRy registers are used as the duty cycle.

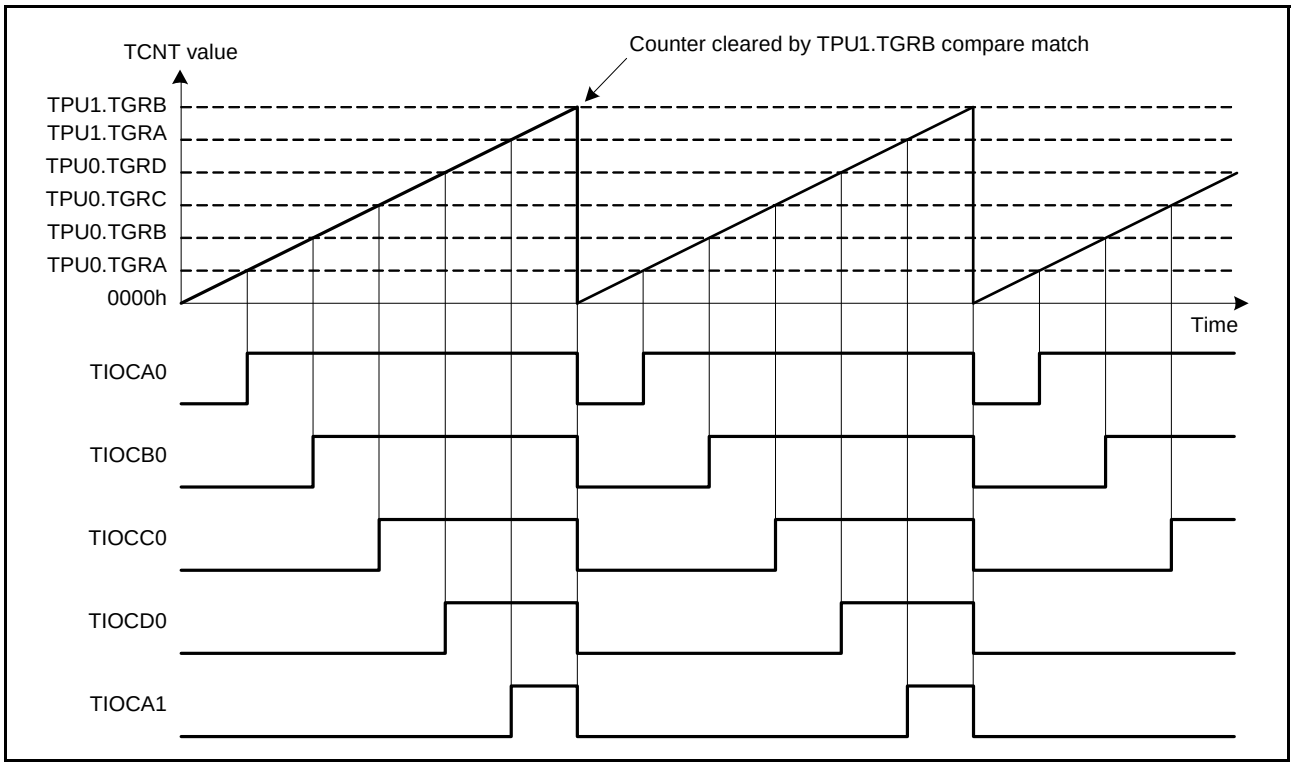


Figure 27.22 Example of PWM Mode Operation (2)

Figure 27.23 shows examples of PWM waveform output with 0% duty cycle and 100% duty cycle in PWM mode.

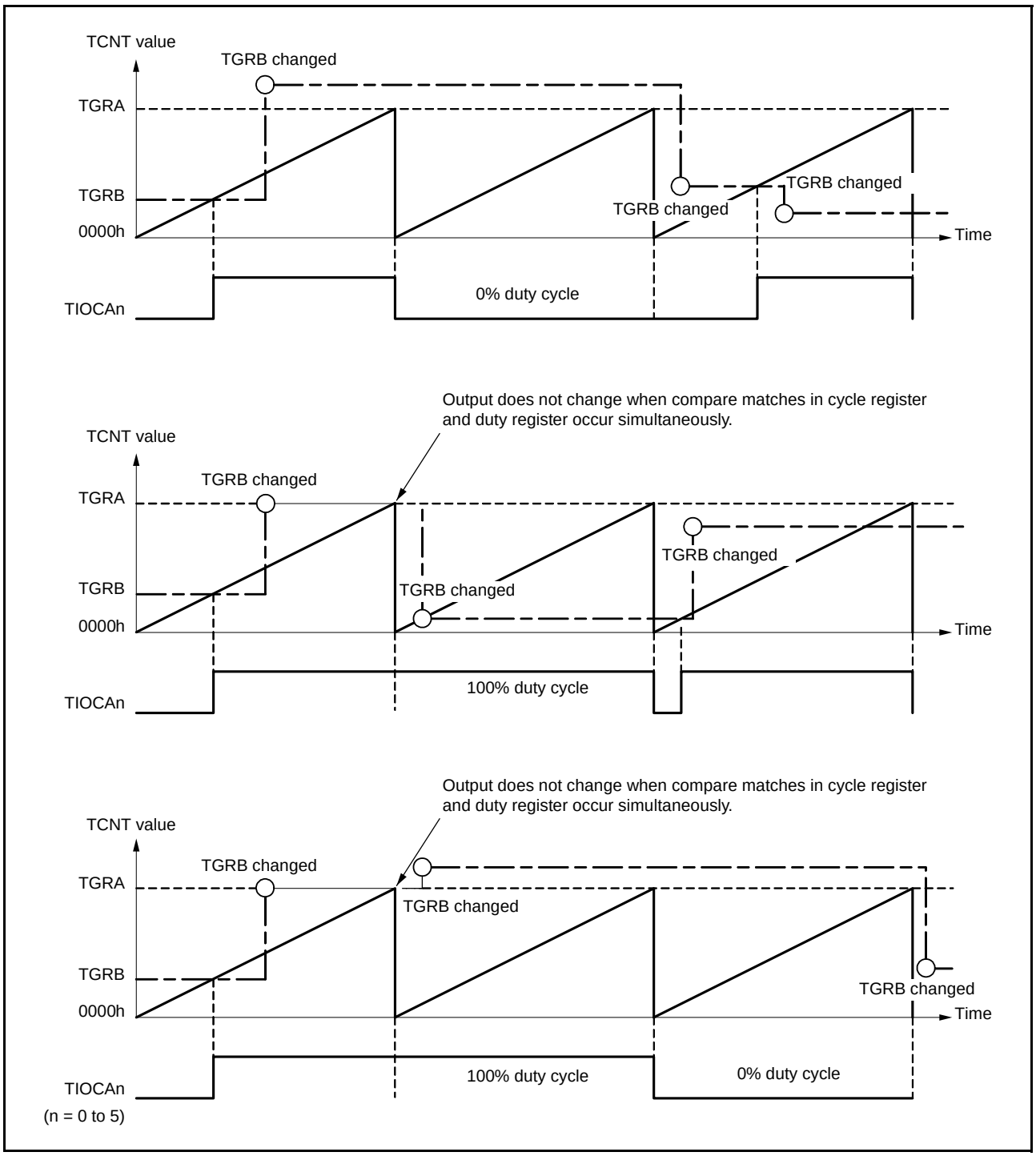


Figure 27.23 Example of PWM Mode Operation (3)

27.3.6 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected by the settings for channels 1, 2, 4, and 5, and TPUm.TCNT is incremented/decremented accordingly.
When phase counting mode is set, an external clock is selected as the count clock and TCNT operates as an up-/down-counter regardless of the setting of the TPSC[2:0] bits and CKEG[1:0] bits in TPUm.TCR. However, the lower 2 bits of the CCLR[2:0] bits in TPUm.TCR and the functions of TPUm.TIORH, TPUm.TIORL, TPUm.TIOR, TPUm.TIER, and TPUm.TGRy are valid, and therefore input capture/compare match and interrupt functions are available.
When an overflow occurs while TCNT is counting up, a TCIV interrupt request is generated; when an underflow occurs while TCNT is counting down, a TCIU interrupt request is generated. The TCFD bit in TPUm.TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down.
In phase counting mode, the external clock pins TCLKA, TCLKB, TCLKC, and TCLKD can be used as 2-phase encoder pulse input.
Table 27.24 lists the correspondence between external clock pins and channels.

Table 27.24 Clock Input Pins in Phase Counting Mode

	External Clock Pins	
Channel	**A-Phase**	**B-Phase**
When TPU1 or TPU5 is set to phase counting mode	TCLKA	TCLKB
When TPU2 or TPU4 is set to phase counting mode	TCLKC	TCLKD

(1) Example of Phase Counting Mode Setting Procedure

Figure 27.24 shows an example of the phase counting mode setting procedure.

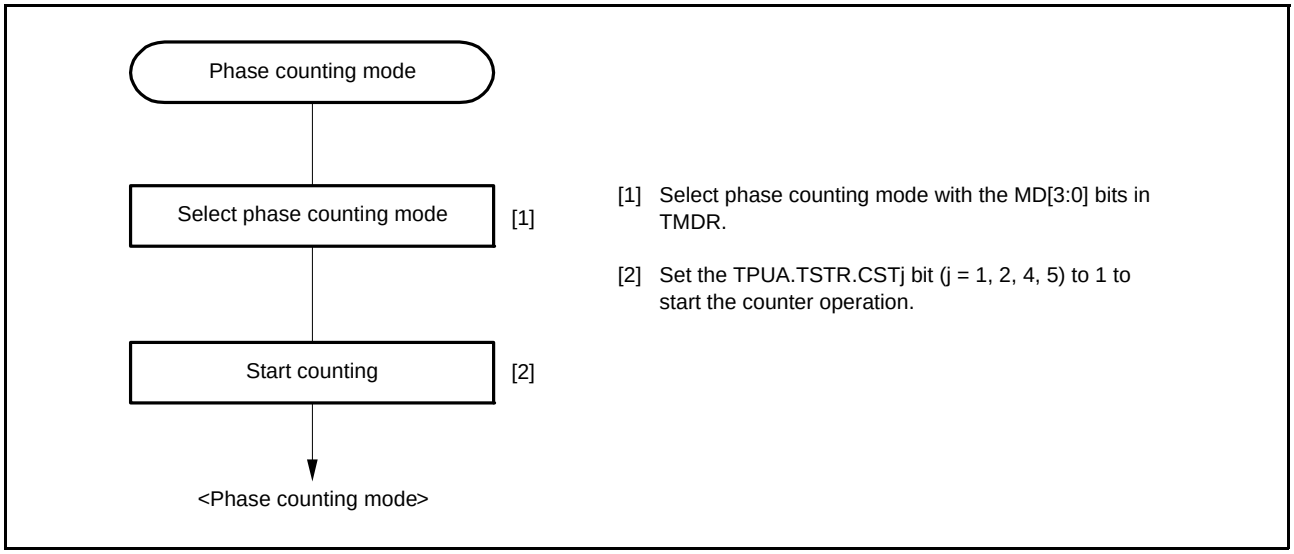


Figure 27.24 Example of Phase Counting Mode Setting Procedure

(2) Examples of Phase Counting Mode Operation

In phase counting mode, TPUm.TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

(a) Phase counting mode 1

Figure 27.25 shows an example of phase counting mode 1 operation, and Table 27.25 lists the TPUm.TCNT up-/down-count conditions.

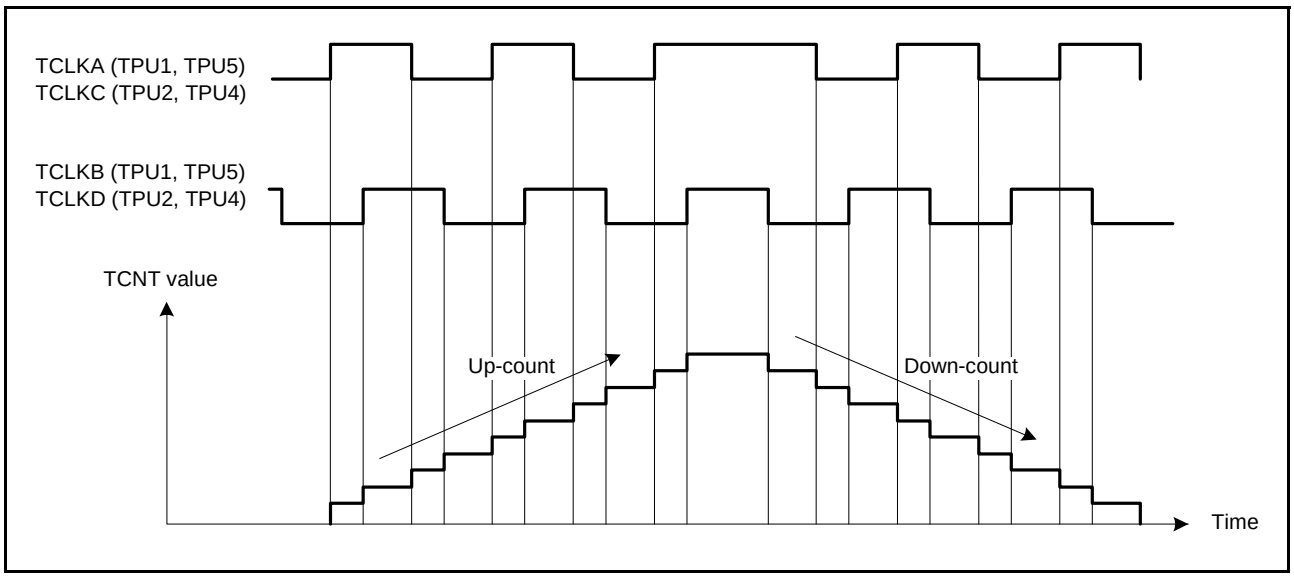


Figure 27.25 Example of Phase Counting Mode 1 Operation

Table 27.25 Up-/Down-Count Conditions in Phase Counting Mode 1

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4)	Operation
High	↑	Up-count
Low	↓	
↑	Low	
↓	High	
High	↓	Down-count
Low	↑	
↑	High	
↓	Low	

↑ : Rising edge

↓ : Falling edge

(b) Phase counting mode 2

Figure 27.26 shows an example of phase counting mode 2 operation, and Table 27.26 lists the TPUm.TCNT up-/down-count conditions.

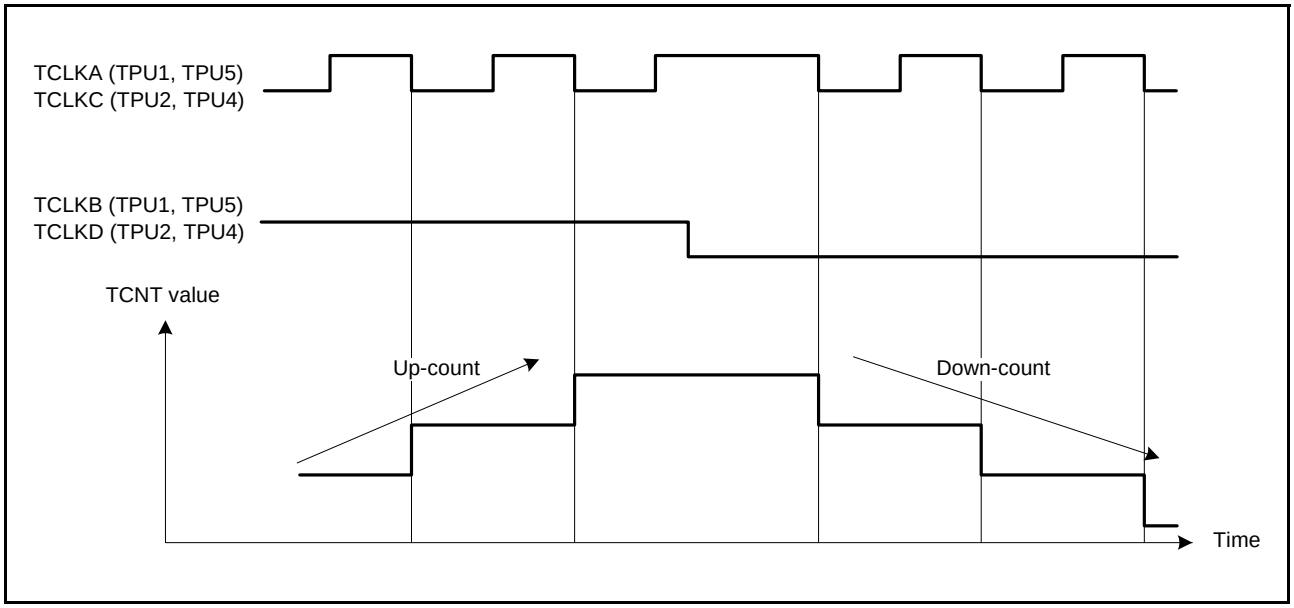


Figure 27.26 Example of Phase Counting Mode 2 Operation

Table 27.26 Up-/Down-Count Conditions in Phase Counting Mode 2

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4)	Operation
High	↑	Don't care
Low	↓	Don't care
↑	Low	Don't care
↓	High	Up-count
High	↓	Don't care
Low	↑	Don't care
↑	High	Don't care
↓	Low	Down-count

↑ : Rising edge

↓ : Falling edge

(c) Phase counting mode 3

Figure 27.27 shows an example of phase counting mode 3 operation, and Table 27.27 lists the TPUm.TCNT up-/down-count conditions.

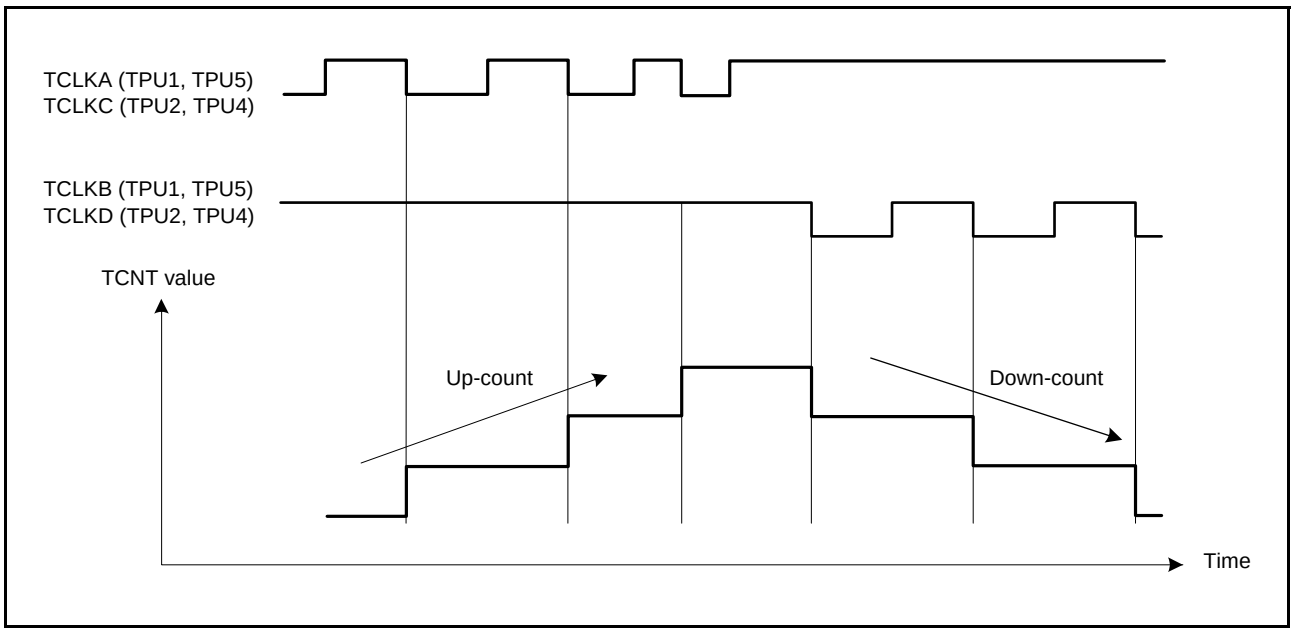


Figure 27.27 Example of Phase Counting Mode 3 Operation

Table 27.27 Up-/Down-Count Conditions in Phase Counting Mode 3

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4)	Operation
High	↑	Don't care
Low	↓	Don't care
↑	Low	Don't care
↓	High	Up-count
High	↓	Down-count
Low	↑	Don't care
↑	High	Don't care
↓	Low	Don't care

↑ : Rising edge

↓ : Falling edge

(d) Phase counting mode 4

Figure 27.28 shows an example of phase counting mode 4 operation, and Table 27.28 lists the TPUm.TCNT up-/down-count conditions.

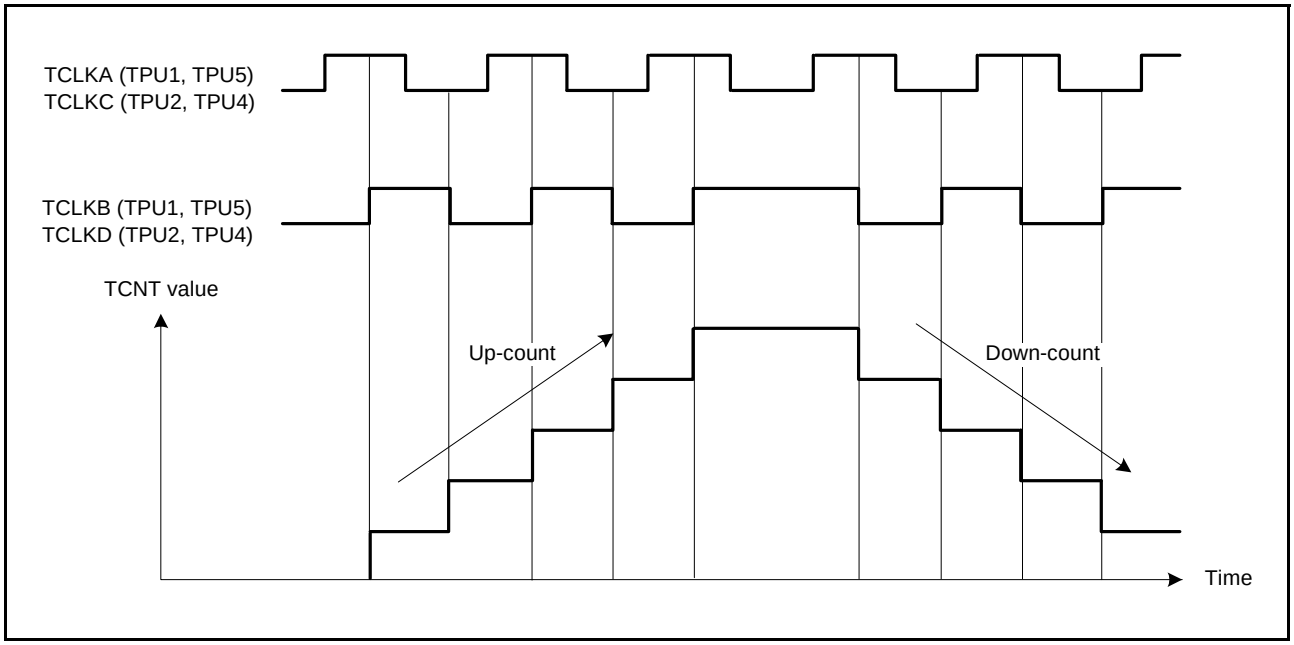


Figure 27.28 Example of Phase Counting Mode 4 Operation

Table 27.28 Up-/Down-Count Conditions in Phase Counting Mode 4

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4)	Operation
High	↑	Up-count
Low	↓	
↑	Low	Don't care
↓	High	
High	↓	Down-count
Low	↑	
↑	High	Don't care
↓	Low	

↑ : Rising edge

↓ : Falling edge

27.3.6.1 Phase Counting Mode Application Example

Figure 27.29 shows an example in which phase counting mode is set for TPU1, and TPU1 is coupled with TPU0 to input servo motor 2-phase encoder pulses in order to detect the position or speed.

TPU1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to the TCLKA and TCLKB pins.

TPU0 operates with TPU0.TCNT clearing by TPU0.TGRC compare match; TPU0.TGRA and TPU0.TGRC are used for the compare match function and are set with the speed control cycle and position control cycle. TPU0.TGRB is used for input capture, with TPU0.TGRB and TPU0.TGRD operating in buffer mode. The TPU1 count clock is specified as the TPU0.TGRB input capture source, and the pulse width of 2-phase encoder 4-multiplication pulses is detected.

TPU1.TGRA and TPU1.TGRB for TPU1 are specified for input capture, TPU0.TGRA and TPU0.TGRC compare matches are selected as the input capture source, and the up-/down-counter values for the control cycles are stored.

This procedure enables accurate position/speed detection to be achieved.

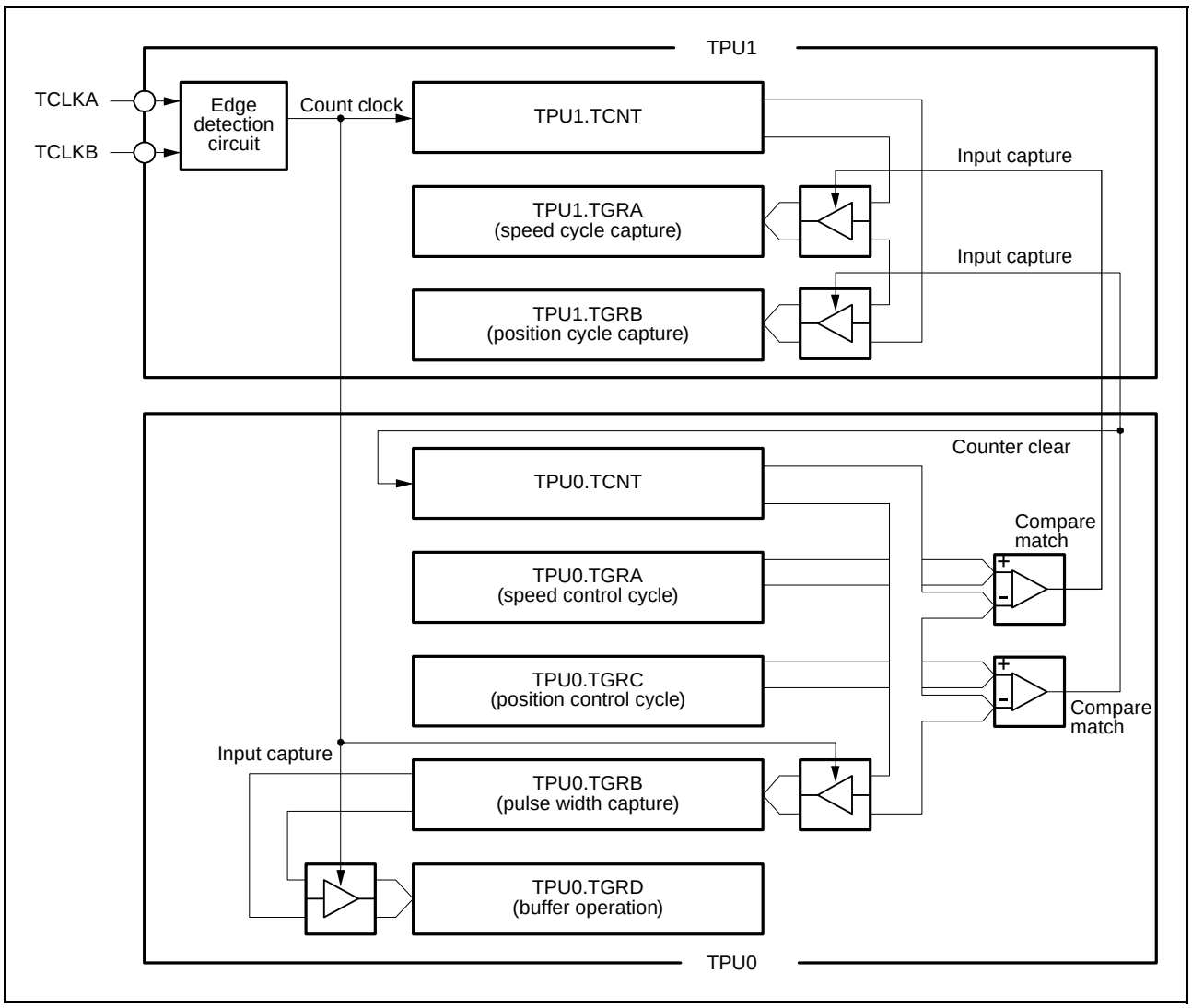


Figure 27.29 Phase Counting Mode Application Example

27.3.7 Noise Filters

Each pin for use in input capture by TPU is equipped with a noise filter. The noise filter samples the level on the pin three times at the selected sampling interval, conveys the level to the internal circuits if the samples match, and continues to convey that level until the other level is sampled from the pins three times in a row. The noise filter function can be enabled or disabled for each pin. Furthermore, sampling clock settings can be made for each channel.
Figure 27.30 is a timing chart for the noise filter.

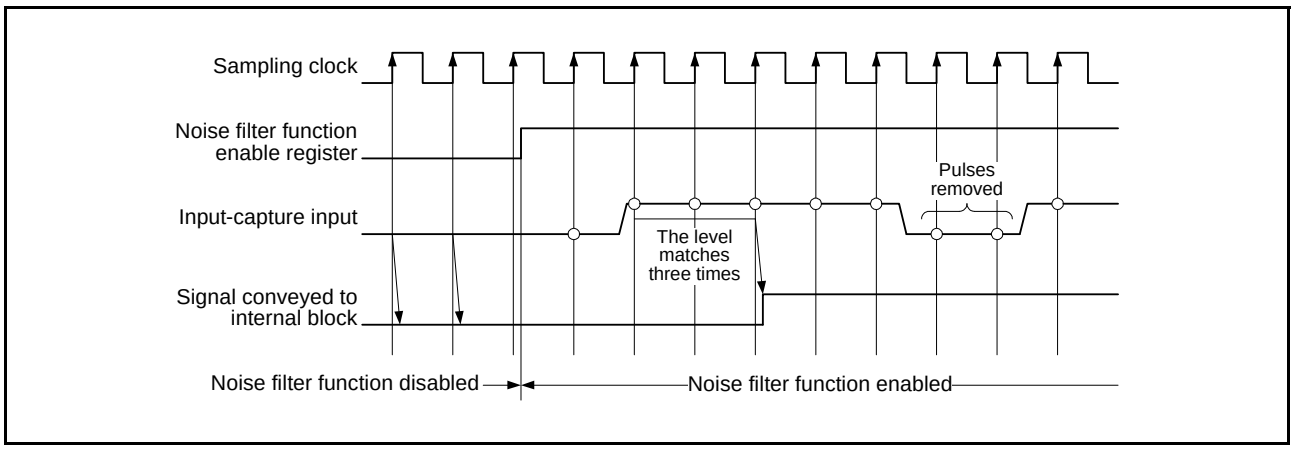


Figure 27.30 Timing Chart for the Noise Filter

If noise filtering is set, input capture operation is performed on the edges of noise-filtered signal after a minimum delay of (sampling interval × 2 + PCLK) due to noise filtering for the input capture input.

27.4 Interrupt Sources

There are three kinds of TPU interrupt sources: TPUm.TGRy input capture/compare match, TPUm.TCNT overflow, and TPUm.TCNT underflow.

Relative channel priority levels can be changed by the interrupt controller, but the priority within a channel is fixed. For details, see section 15, Interrupt Controller (ICUA).

Table 27.29 lists the TPU interrupt sources.

Table 27.29 TPU Interrupt Sources

Channel	Name	Interrupt Source	DTC Activation	DMAC Activation
TPU0	TGI0A	TPU0.TGRA input capture/compare match	Possible	Possible
	TGI0B	TPU0.TGRB input capture/compare match	Possible	Not possible
	TGI0C	TPU0.TGRC input capture/compare match	Possible	Not possible
	TGI0D	TPU0.TGRD input capture/compare match	Possible	Not possible
	TCI0V	TPU0.TCNT overflow	Not possible	Not possible
TPU1	TGI1A	TPU1.TGRA input capture/compare match	Possible	Possible
	TGI1B	TPU1.TGRB input capture/compare match	Possible	Not possible
	TCI1V	TPU1.TCNT overflow	Not possible	Not possible
	TCI1U	TPU1.TCNT underflow	Not possible	Not possible
TPU2	TGI2A	TPU2.TGRA input capture/compare match	Possible	Possible
	TGI2B	TPU2.TGRB input capture/compare match	Possible	Not possible
	TCI2V	TPU2.TCNT overflow	Not possible	Not possible
	TCI2U	TPU2.TCNT underflow	Not possible	Not possible
TPU3	TGI3A	TPU3.TGRA input capture/compare match	Possible	Possible
	TGI3B	TPU3.TGRB input capture/compare match	Possible	Not possible
	TGI3C	TPU3.TGRC input capture/compare match	Possible	Not possible
	TGI3D	TPU3.TGRD input capture/compare match	Possible	Not possible
	TCI3V	TPU3.TCNT overflow	Not possible	Not possible
TPU4	TGI4A	TPU4.TGRA input capture/compare match	Possible	Possible
	TGI4B	TPU4.TGRB input capture/compare match	Possible	Not possible
	TCI4V	TPU4.TCNT overflow	Not possible	Not possible
	TCI4U	TPU4.TCNT underflow	Not possible	Not possible
TPU5	TGI5A	TPU5.TGRA input capture/compare match	Possible	Possible
	TGI5B	TPU5.TGRB input capture/compare match	Possible	Not possible
	TCI5V	TPU5.TCNT overflow	Not possible	Not possible
	TCI5U	TPU5.TCNT underflow	Not possible	Not possible

Note: This table lists the initial state immediately after a reset. The relative channel priority levels can be changed by the interrupt controller.

(1) Input Capture/Compare Match Interrupt

An interrupt is requested when the TGIEy bit (y = A, B, C, D) in TPUm.TIER is set to 1 by the occurrence of a TPUm.TGRy input capture/compare match on a channel. The TPU has 16 input capture/compare match interrupts, four each for TPU0 and TPU3, and two each for TPU1, TPU2, TPU4, and TPU5.

(2) Overflow Interrupt

An interrupt is requested when the TCIEV bit in TPUm.TIER is set to 1 by the occurrence of a TPUm.TCNT overflow on a channel. The TPU has six overflow interrupts, one for each channel.

(3) Underflow Interrupt

An interrupt is requested when the TCIEU bit in TPUm.TIER is set to 1 by the occurrence of a TPUm.TCNT underflow on a channel. The TPU has four underflow interrupts, one each for TPU1, TPU2, TPU4, and TPU5.

27.5 DTC Activation

The DTC can be activated by the TPUm.TGRy input capture/compare match interrupt of each channel. For details, see section 20, Data Transfer Controller (DTCa).
A total of 16 input capture/compare match interrupts can be used as DTC activation sources, four each for TPU0 and TPU3, and two each for TPU1, TPU2, TPU4, and TPU5.

27.6 DMAC Activation

The DMAC can be activated by the TPUm.TGRA input capture/compare match interrupt of each channel. For details, see section 18, DMA Controller (DMACAa).
A total of six TPUm.TGRA input capture/compare match interrupts can be used as DMAC activation sources, one for each channel.

27.7 A/D Converter Activation

The TPU can activate the A/D converter by the TPUm.TGRA input capture/compare match for each channel.
When the TTGE bit in TPUm.TIER is set to 1, the TPU requests the A/D converter to start A/D conversion by the occurrence of a TPUm.TGRA input capture/compare match on a particular channel.

27.8 PPG Trigger

Input capture to or compare match with TGRA and TGRB in TPU0 to TPU3 can be made to act as a PPG1 waveform trigger. For details, see section 28, Programmable Pulse Generator (PPG).

27.9 Operation Timing

27.9.1 Input/Output Timing

(1) TPUm.TCNT Count Timing

Figure 27.31 shows TPUm.TCNT count timing in internal clock operation, and Figure 27.32 shows TCNT count timing in external clock operation.

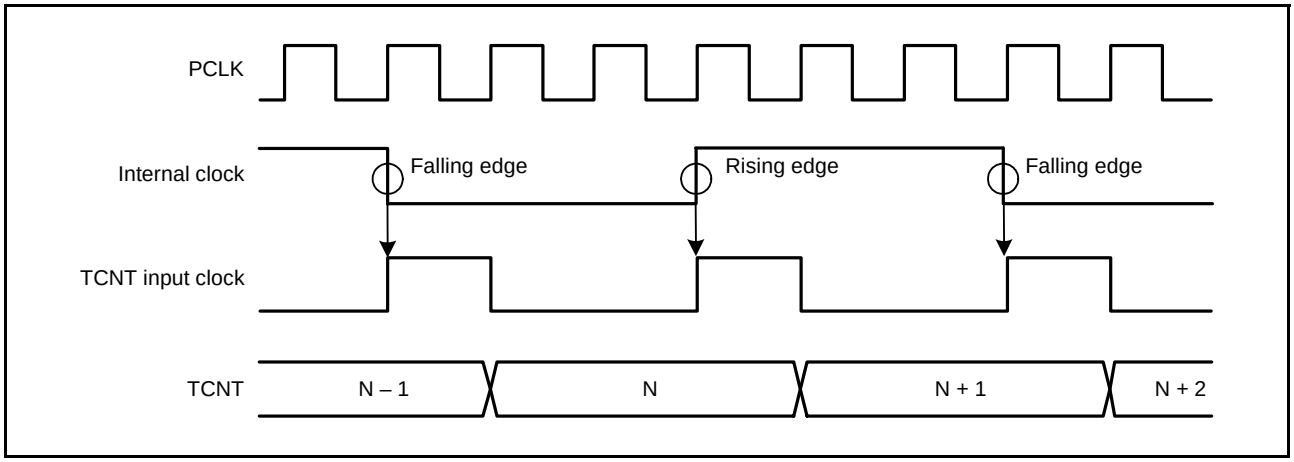


Figure 27.31 Count Timing in Internal Clock Operation

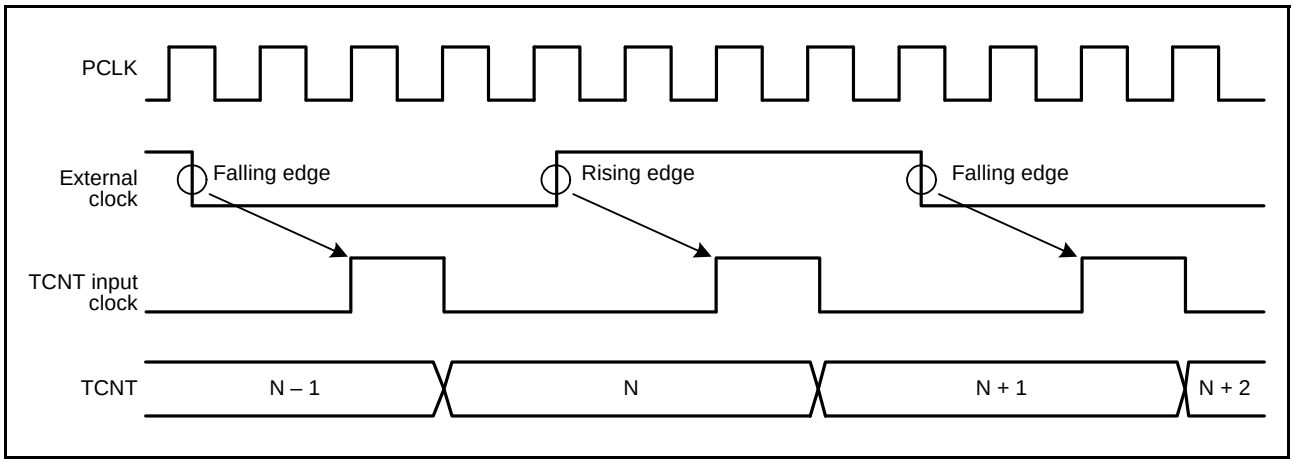


Figure 27.32 Count Timing in External Clock Operation

(2) Output Compare Output Timing

A compare match signal is generated in the final state in which TPUm.TCNT and TPUm.TGRy match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TPUm.TIORH, TPUm.TIORL, or TPUm.TIOR is output to the output compare output pin TIOCyn (y = A to D; n = 0 to 5). After a match between TCNT and TGRy, the compare match signal is not generated until the TCNT input clock is generated.

Figure 27.33 shows output compare output timing.

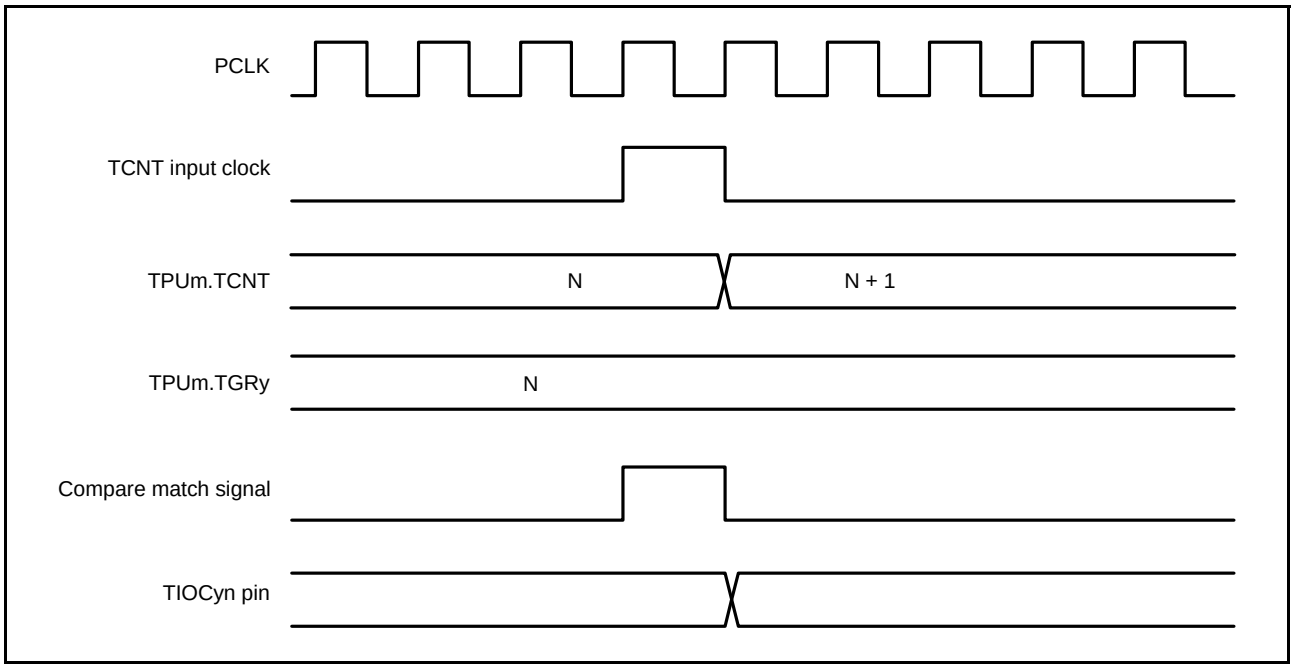


Figure 27.33 Output Compare Output Timing

(3) Input Capture Signal Timing

Figure 27.34 shows input capture signal timing.

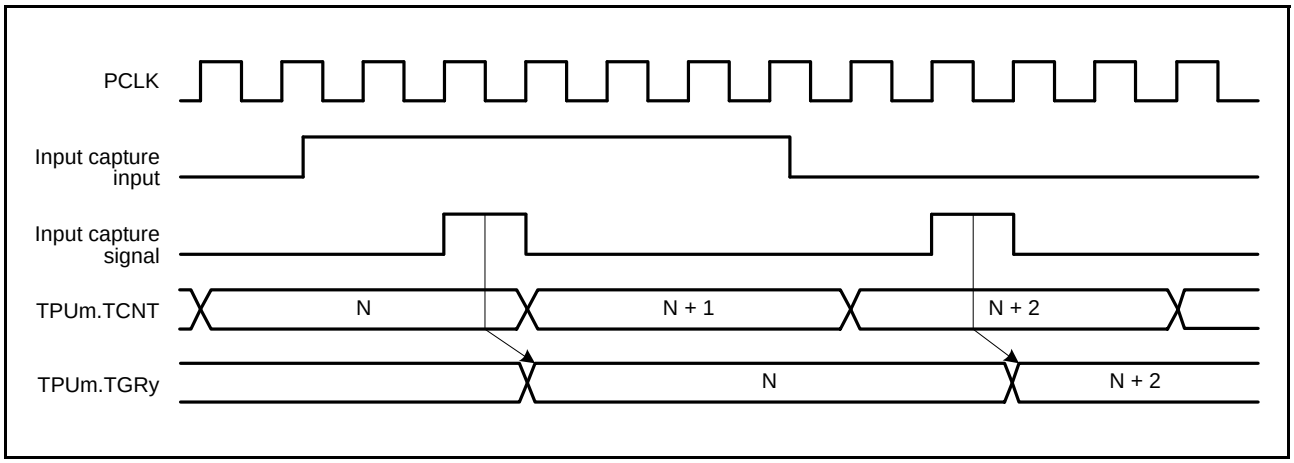


Figure 27.34 Input Capture Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 27.35 shows the timing when counter clearing by compare match occurrence is specified, and Figure 27.36 shows the timing when counter clearing by input capture occurrence is specified.

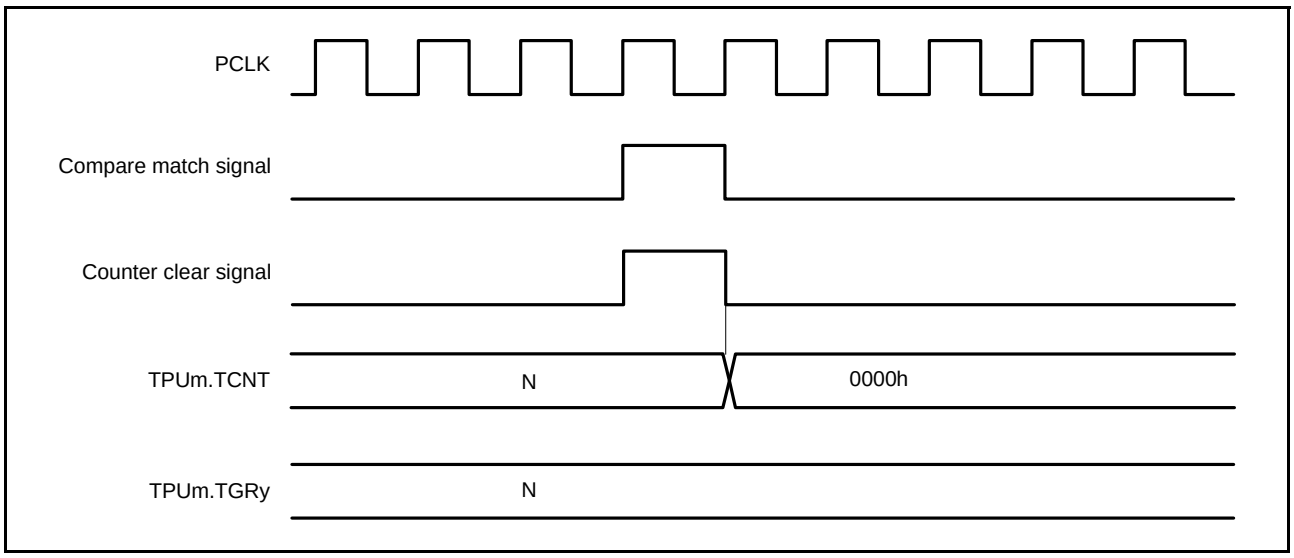


Figure 27.35 Counter Clear Timing (Compare Match)

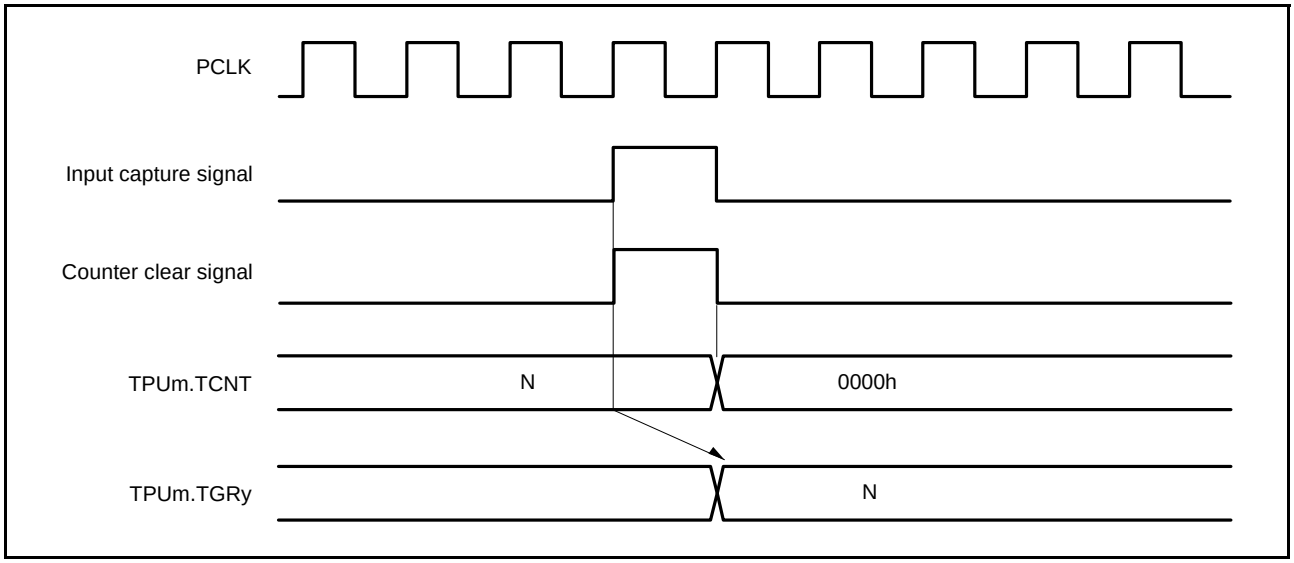


Figure 27.36 Counter Clear Timing (Input Capture)

(5) Buffer Operation Timing

Figure 27.37 and Figure 27.38 show the timings in buffer operation.

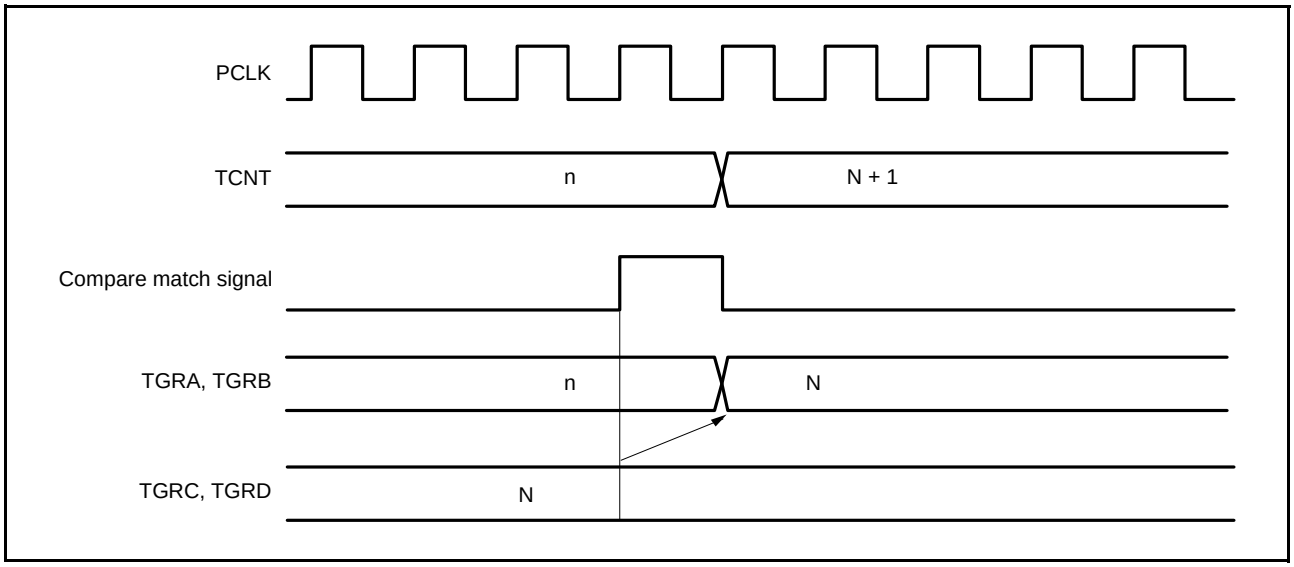


Figure 27.37 Buffer Operation Timing (Compare Match)

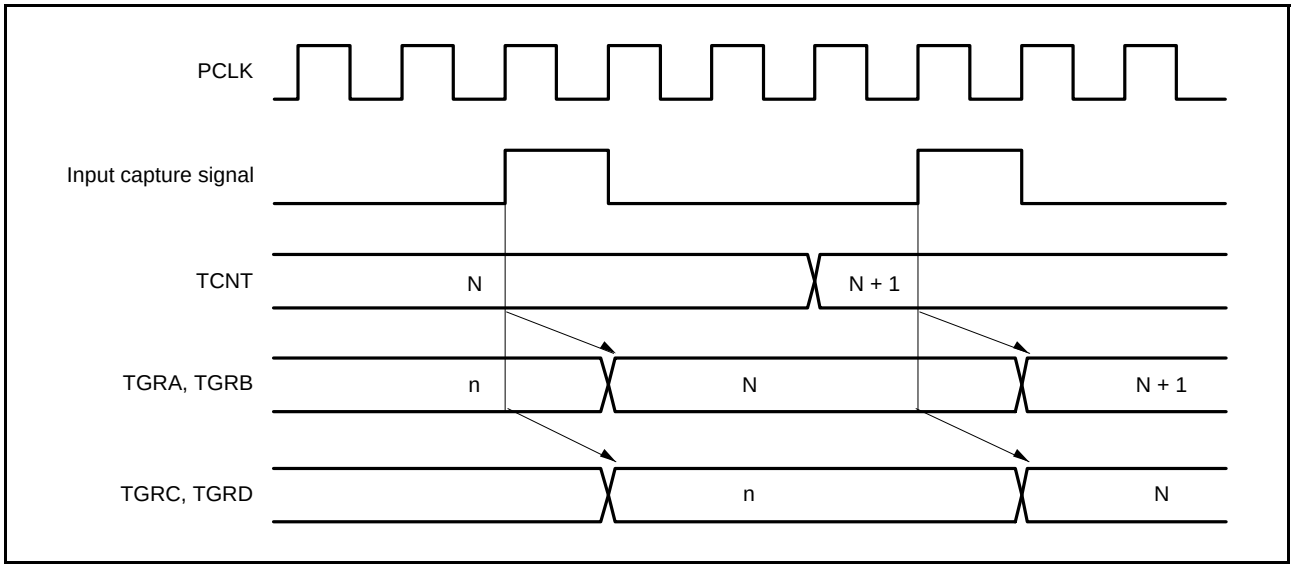


Figure 27.38 Buffer Operation Timing (Input Capture)

27.9.2 Interrupt Signal Timing

(1) Timing of Interrupt Signal Setting on Compare Match

Figure 27.39 shows the timing for setting the interrupt signal by compare match occurrence.

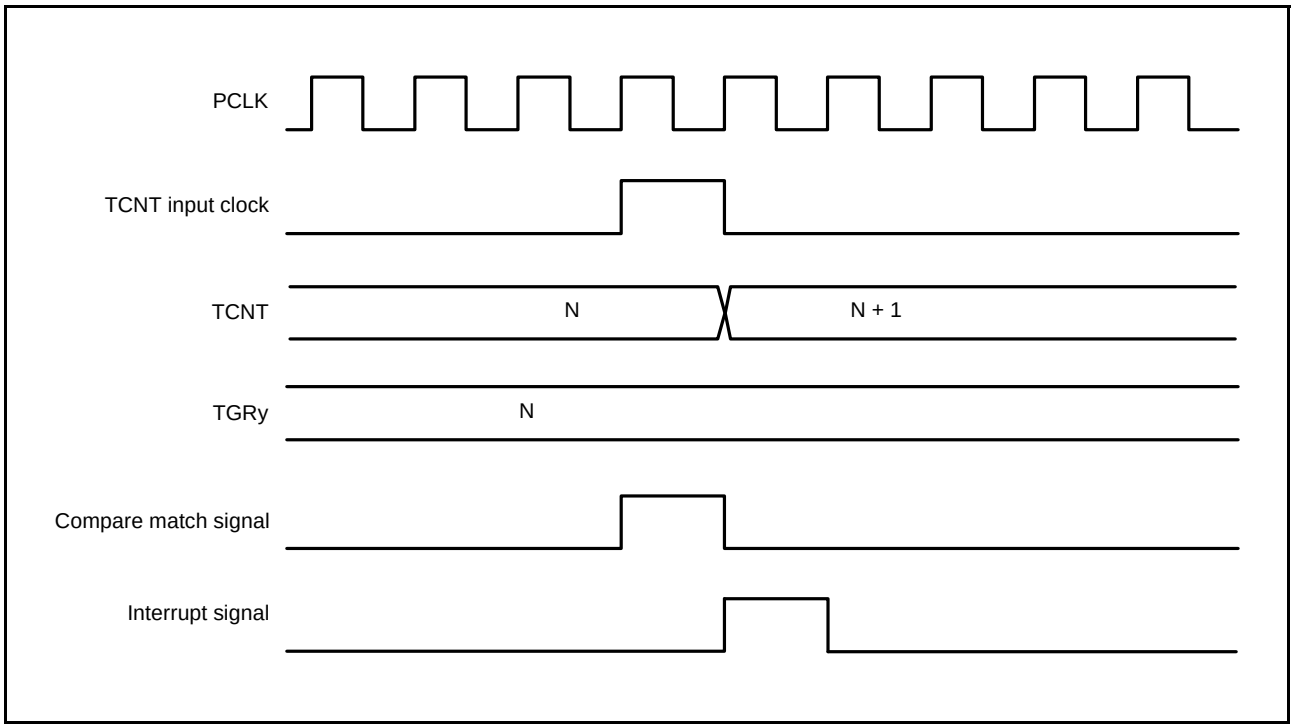


Figure 27.39 TGImy Interrupt Timing (Compare Match)

(2) Timing of Interrupt Signal Setting on Input Capture

Figure 27.40 shows the timing for setting the interrupt signal by input capture occurrence.

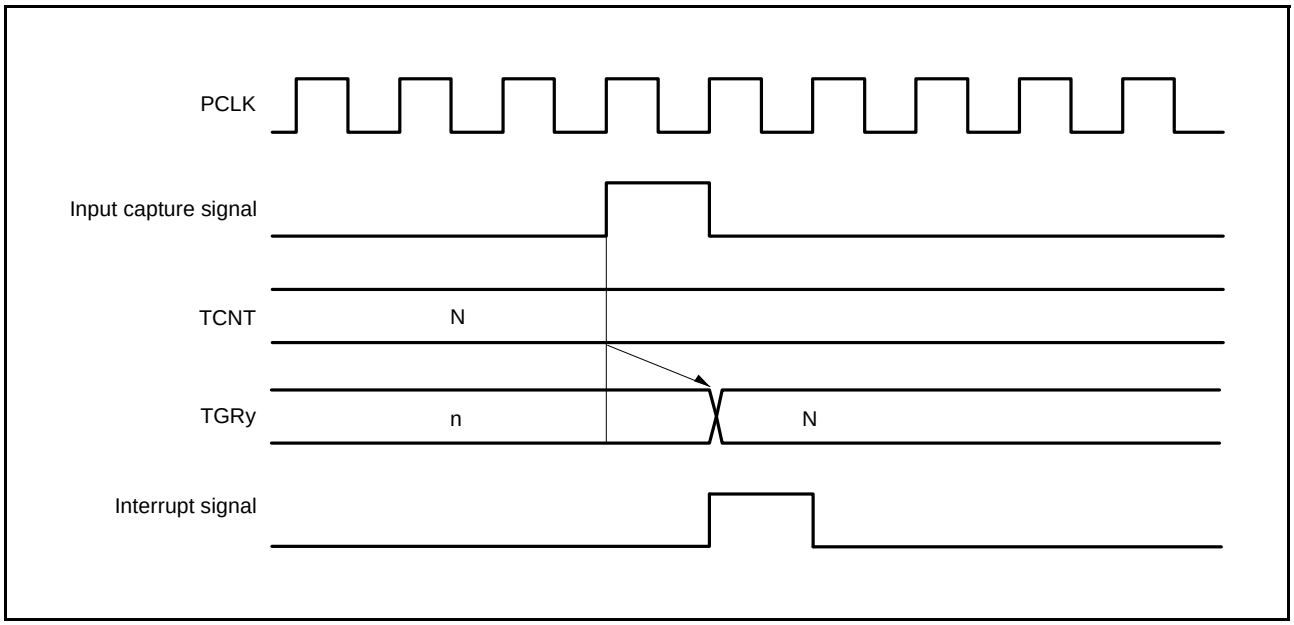


Figure 27.40 TGImy Interrupt Timing (Input Capture)

(3) Timing of TCImV/TCImU Interrupt Signal Setting

Figure 27.41 shows the timing for generating the TCImV interrupt signal by overflow occurrence.
Figure 27.42 shows the timing for generating the TCImU interrupt signal by underflow occurrence.

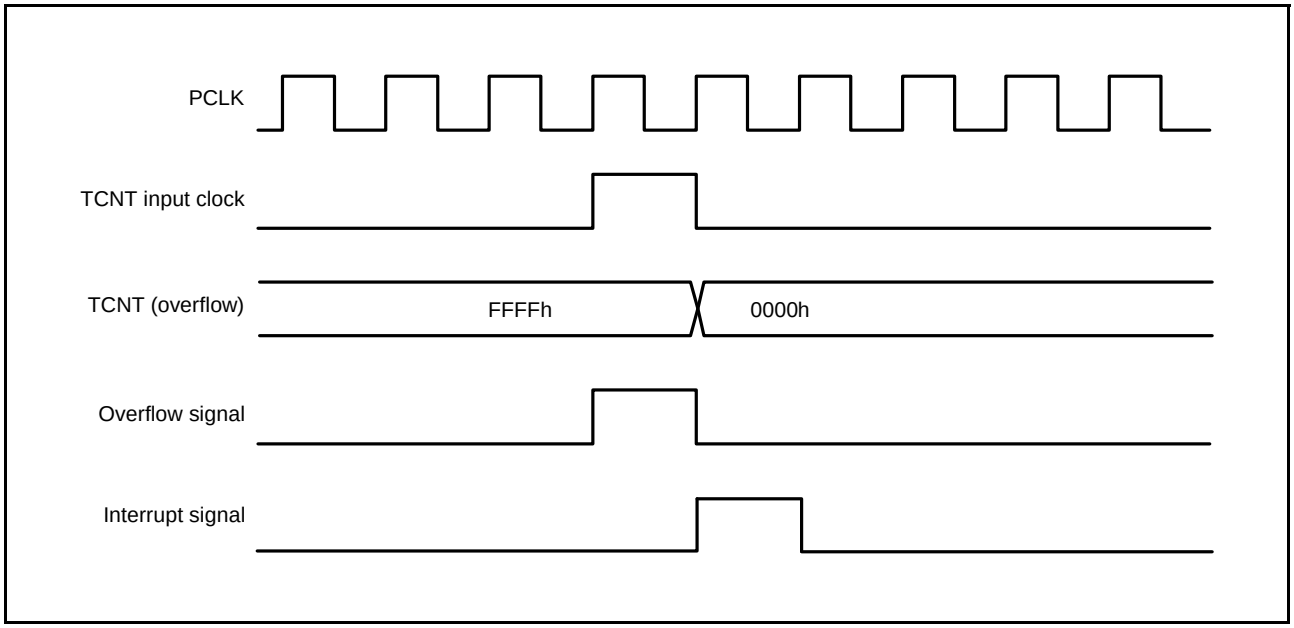


Figure 27.41 TCImV Interrupt Setting Timing

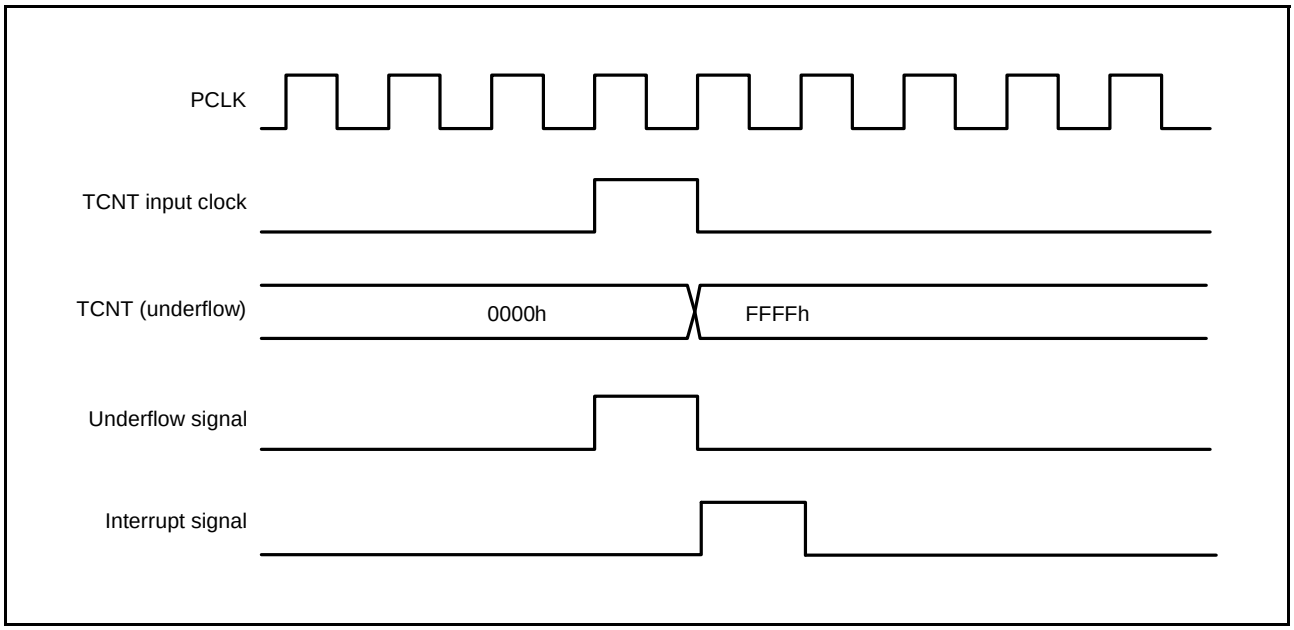


Figure 27.42 TCImU Interrupt Setting Timing

27.10 Usage Notes

27.10.1 Module Stop Function Setting

Operation of the TPU can be disabled or enabled using the module stop control register. The TPU does not operate with the initial setting. Register access is enabled by releasing the module stop state. For details, see section 11, Low Power Consumption.

27.10.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 PCLK cycles in the case of single-edge detection, and at least 2.5 PCLK cycles in the case of both-edge detection. The TPU will not operate properly with a narrower pulse width.
In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 PCLK cycles, and the pulse width must be at least 2.5 PCLK cycles. Figure 27.43 shows the input clock conditions in phase counting mode.

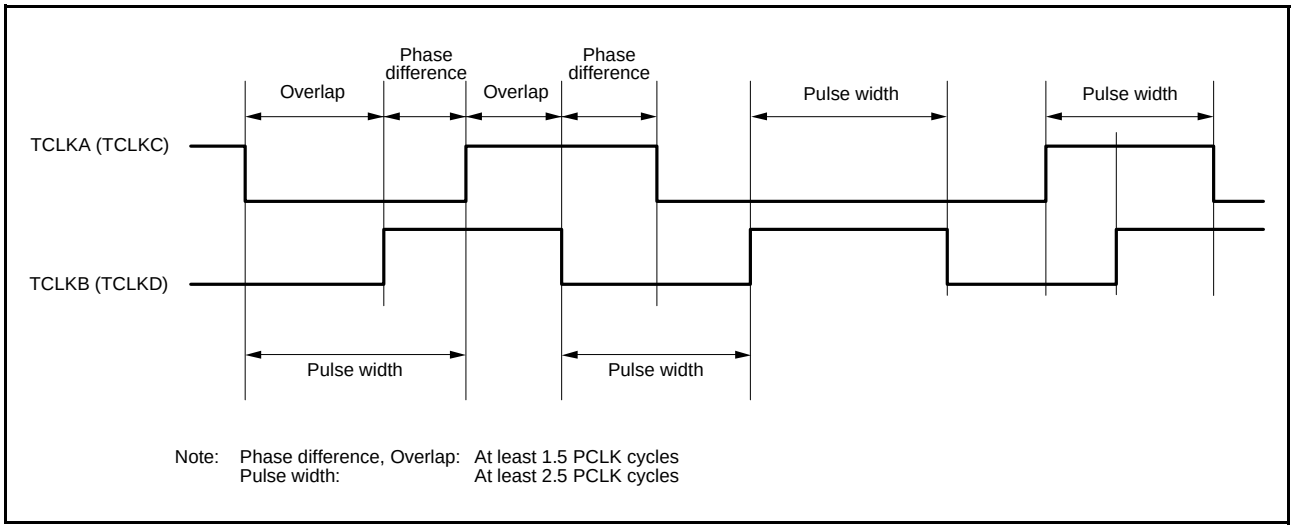


Figure 27.43 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

27.10.3 Notes on Cycle Setting

When counter clearing by compare match is set, TPUm.TCNT is cleared in the final state in which it matches the TPUm.TGRy value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{f_{TCNT\_CLK}}{(N + 1)}$$

f: Counter frequency
f_{TCNT_CLK}: Count clock frequency
N: TGRy set value

27.10.4 Conflict between TPUm.TCNT Write and Clear Operations

If the counter clearing signal is generated in a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed. Figure 27.44 shows the timing in this case.

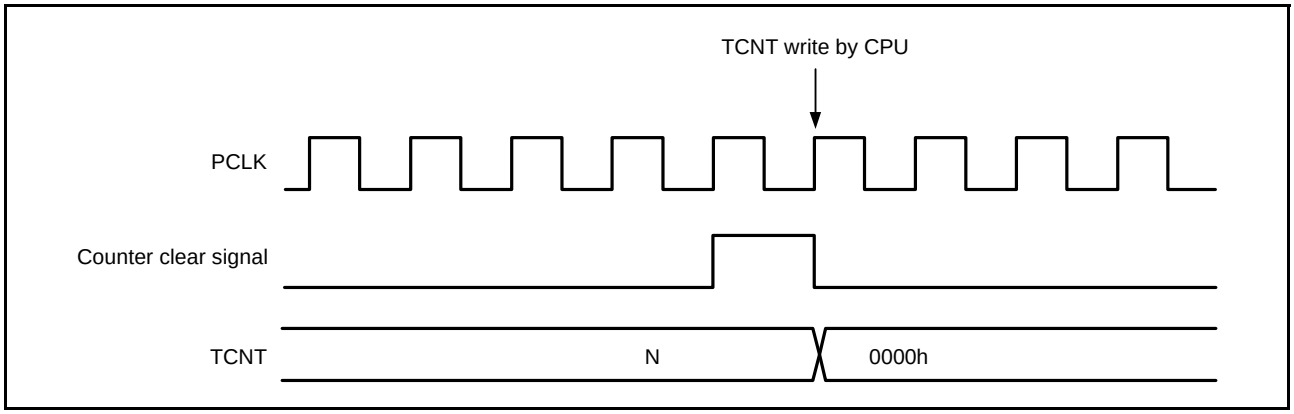


Figure 27.44 Conflict between TPUm.TCNT Write and Clear Operations

27.10.5 Conflict between TPUm.TCNT Write and Increment Operations

If incrementing occurs in a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented. Figure 27.45 shows the timing in this case.

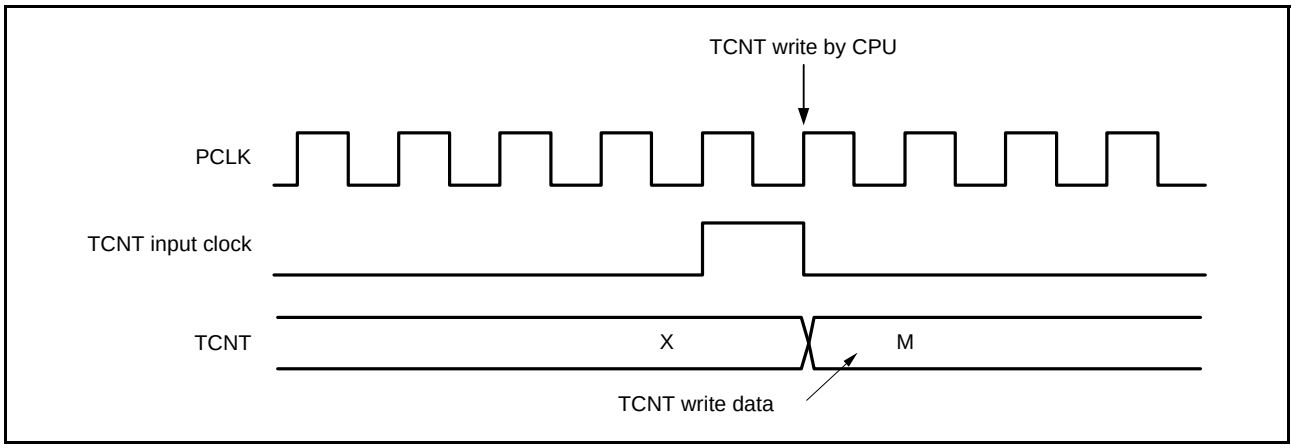


Figure 27.45 Conflict between TPUm.TCNT Write and Increment Operations

27.10.6 Conflict between TPUm.TGRy Write and Compare Match

If a compare match occurs in a TGRy write cycle, the TGRy write takes precedence and the compare match signal is disabled. A compare match also does not occur when the same value as before is written.
Figure 27.46 shows the timing in this case.

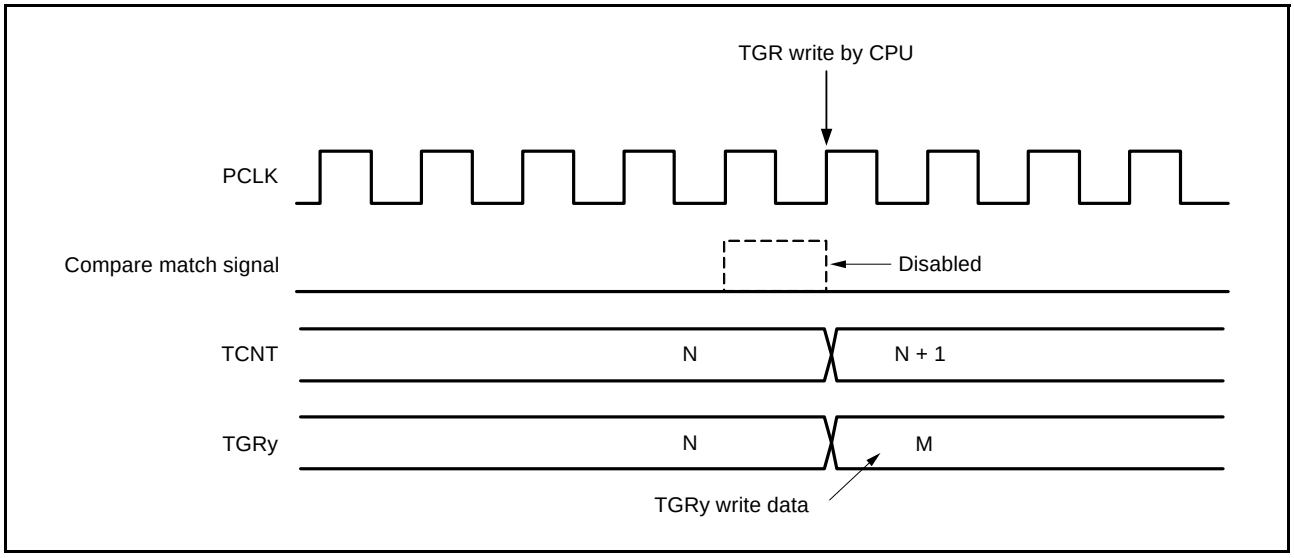


Figure 27.46 Conflict between TPUm.TGRy Write and Compare Match

27.10.7 Conflict between Buffer Register Write and Compare Match

If a compare match occurs in a TPUm.TGRy write cycle, the data transferred to TGRy by the buffer operation will be the data before writing.
Figure 27.47 shows the timing in this case.

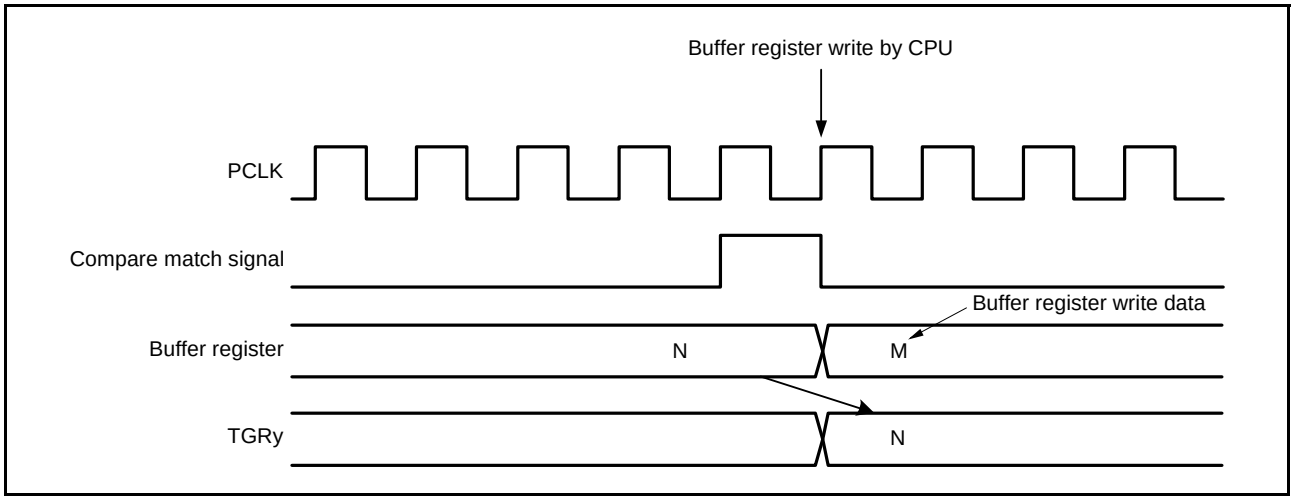


Figure 27.47 Conflict between Buffer Register Write and Compare Match

27.10.8 Conflict between TPUm.TGRy Read and Input Capture

If the input capture signal is generated in a TGRy read cycle, the data that is read will be the data before input capture transfer.

Figure 27.48 shows the timing in this case.

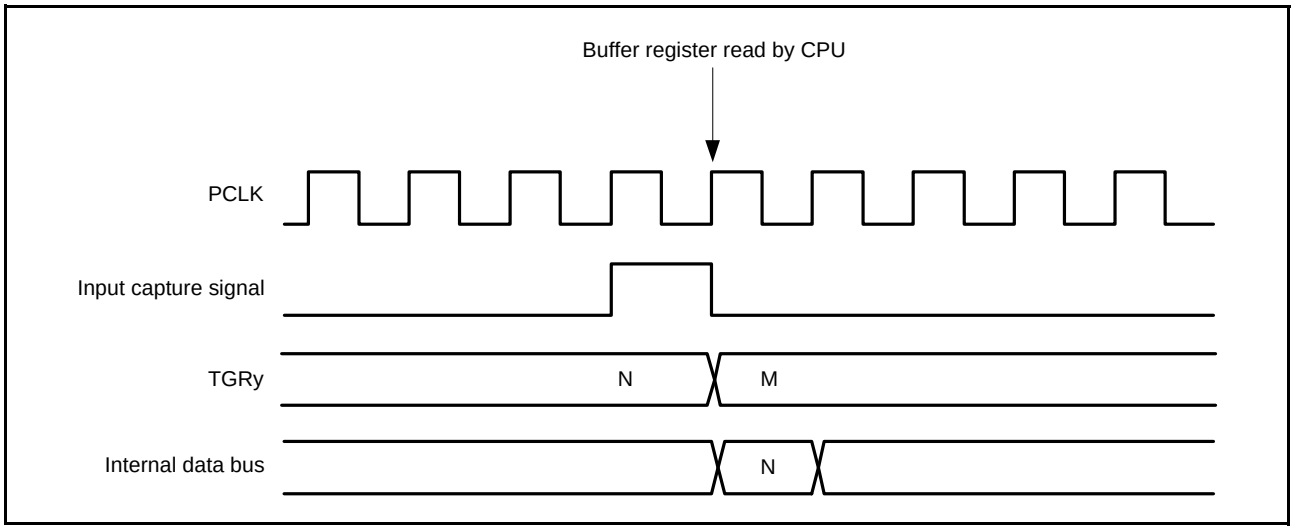


Figure 27.48 Conflict between TPUm.TGRy Read and Input Capture

27.10.9 Conflict between TPUm.TGRy Write and Input Capture

If the input capture signal is generated in a TGRy write cycle, the input capture operation takes precedence and the write to TGRy is not performed. Figure 27.49 shows the timing in this case.

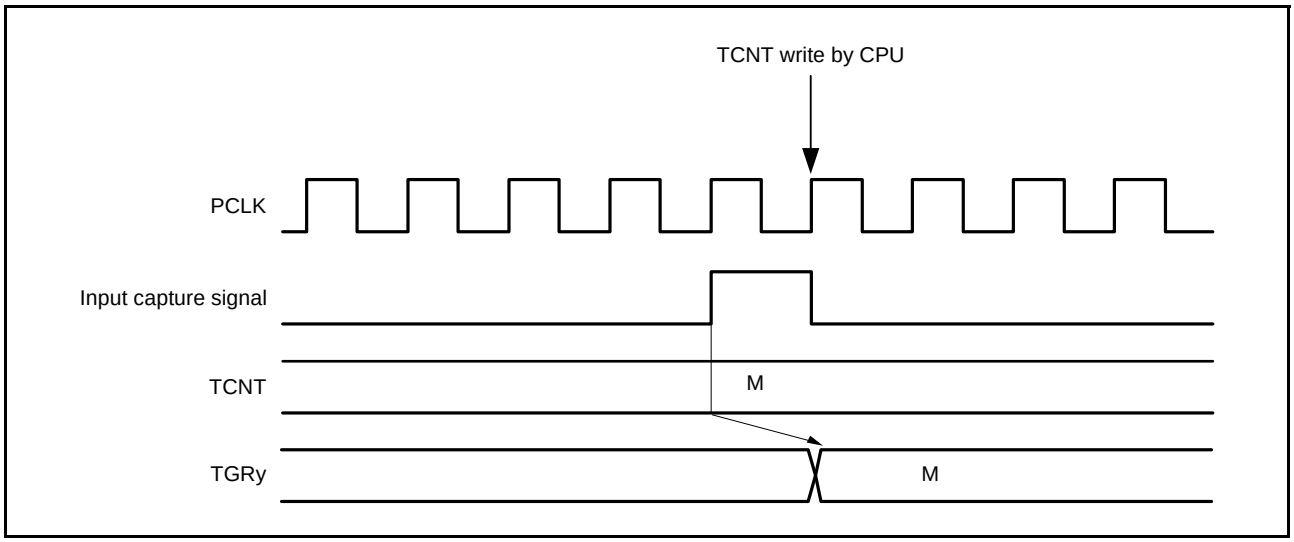


Figure 27.49 Conflict between TPUm.TGRy Write and Input Capture

27.10.10 Conflict between Buffer Register Write and Input Capture

If the input capture signal is generated in a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed. Figure 27.50 shows the timing in this case.

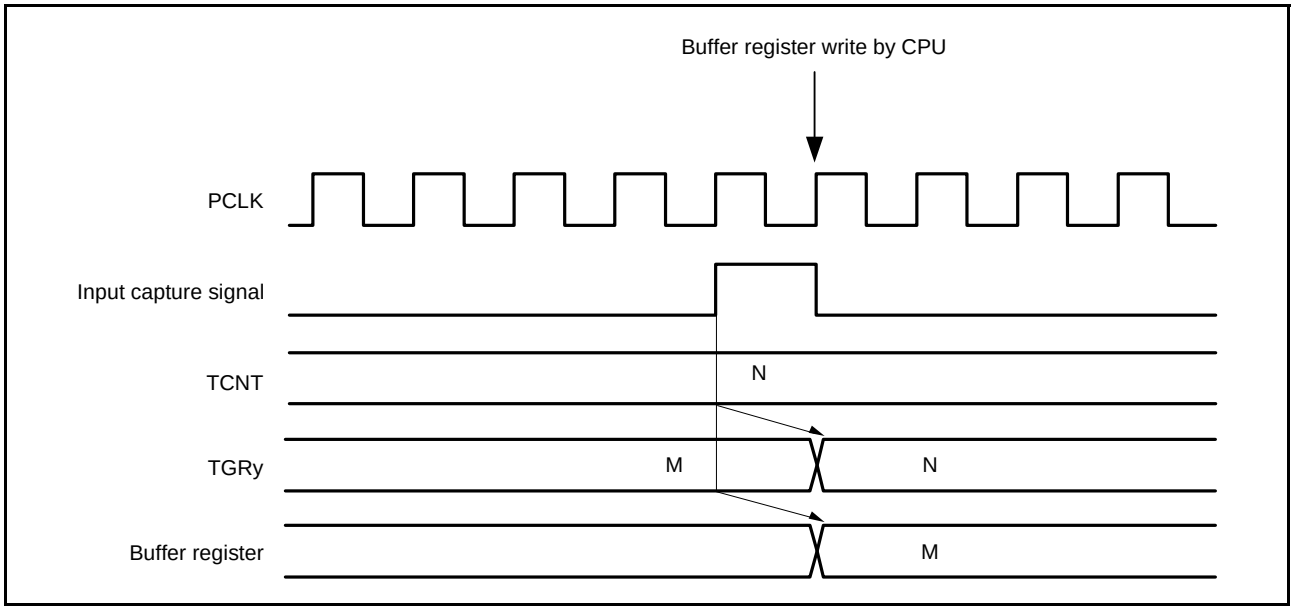


Figure 27.50 Conflict between Buffer Register Write and Input Capture

27.10.11 Conflict between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing*1 occur simultaneously, TPUm.TCNT is cleared with the generation of the compare match interrupt and an overflow interrupt is generated.

Figure 27.51 shows the operation timing when a TPUm.TGRy compare match is specified as the clearing source and FFFFh is set in TGRy.

Note 1. There are four counter clearing sources:
- Compare match
- Input capture
- Synchronous clearing
- Counter restart operation by an event signal

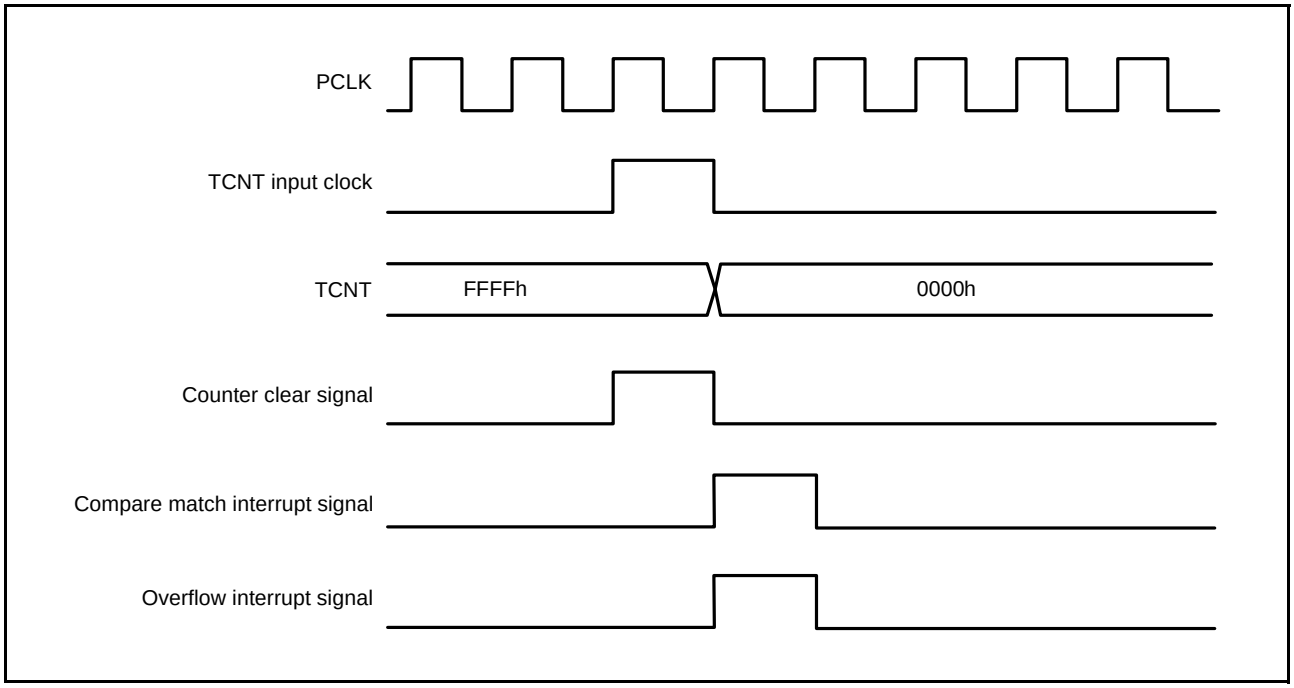


Figure 27.51 Conflict between Overflow and Counter Clearing

27.10.12 Conflict between TPUm.TCNT Write and Overflow/Underflow

If an overflow/underflow occurs due to increment/decrement in a TCNT write cycle, the TCNT write takes precedence. Figure 27.52 shows the operation timing when there is conflict between TCNT write and overflow.

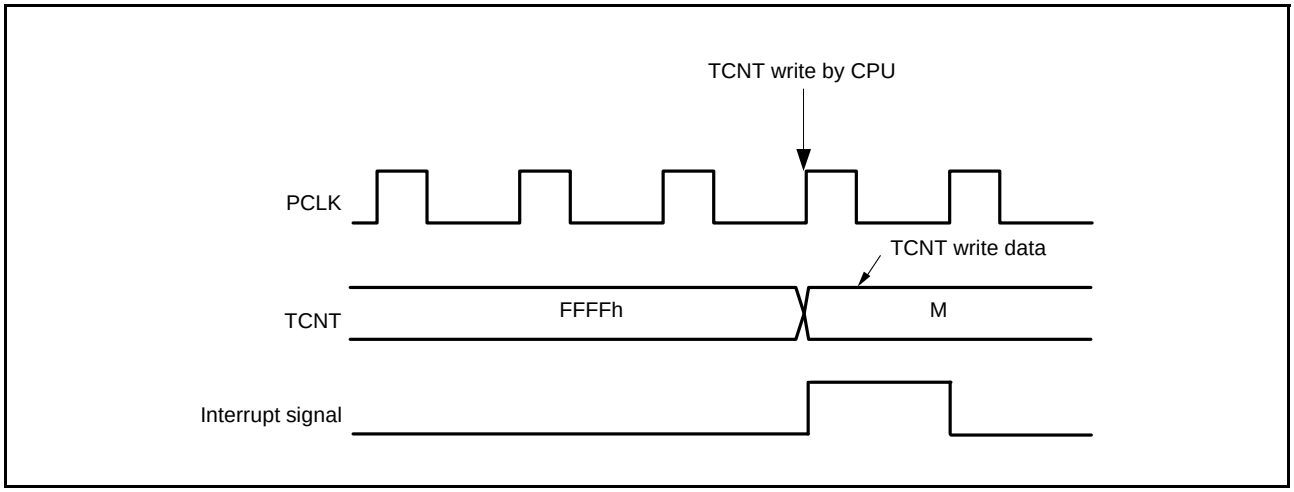


Figure 27.52 Conflict between TPUm.TCNT Write and Overflow

27.10.13 Multiplexing of I/O Pins

In this MCU, the TCLKA input pin is multiplexed with the TIOCB5 I/O pin, the TCLKB input pin with the TIOCB2 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, the TCLKD input pin with the TIOCB0 I/O pin, the TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCC3 I/O pin, and the TCLKD input pin with the TIOCD3 I/O pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.

27.10.14 Continuous Output of Compare-Match Pulse Interrupt Signal

When TGR is set to 0000h, PCLK/1 is set as the count clock, and compare match is set as the counter clear source, the TCNT remains 0000h and is not updated, and a compare-match pulse interrupt signal is output continuously to form a flat signal level.

When a pulse interrupt signal is used, the interrupt controller cannot detect the second and subsequent interrupts.

Figure 27.53 shows an operation timing when the compare-match pulse interrupt signal is continuously output.

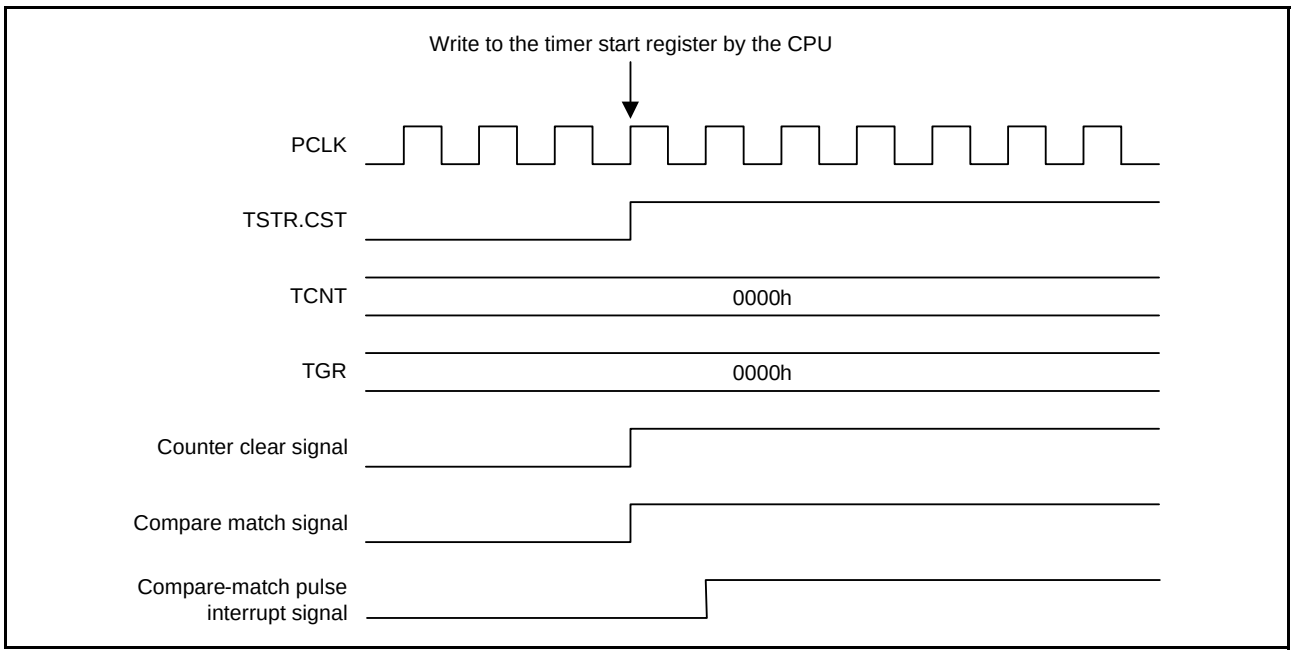


Figure 27.53 Continuous Output of Compare-Match Pulse Interrupt Signal

27.10.15 Continuous Output of Input-Capture Pulse Interrupt Signal

When input-capture signal is set on both edges and when the pulse width of the input-capture input equals to one PCLK cycle detected by internal sampling, input capture is generated continuously on the rising and falling edges. Therefore, an input-capture pulse interrupt signal is output continuously to form a flat signal level.
When a pulse interrupt signal is used, the interrupt controller cannot detect the second and subsequent interrupts.
Figure 27.54 shows an operation timing when the input-capture pulse interrupt signal is output continuously.

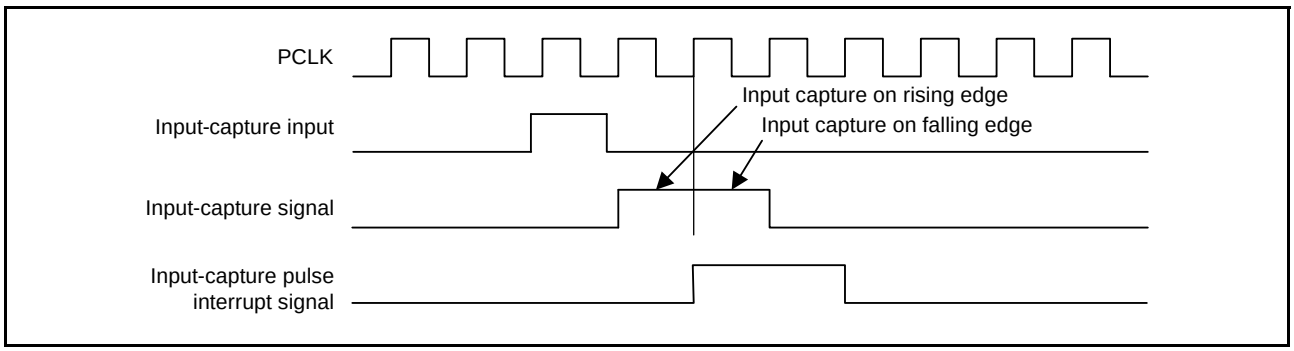


Figure 27.54 Continuous Output of Input-Capture Pulse Interrupt Signal

27.10.16 Continuous Output of Underflow Pulse Interrupt Signal

If two external clock signals' same direction edges to be phase counted are generated within two PCLK cycles in phase counting mode 1, with TGR being 0000h, and compare match set as the counter clear source, the TCNT remains 0000h and is not updated, and a compare-match pulse interrupt signal and an underflow interrupt signal are output continuously to form a flat signal level.

When a pulse interrupt signal is used, the interrupt controller cannot detect the second and subsequent interrupts.

Figure 27.55 shows an operation timing when the underflow pulse interrupt signal is output continuously.

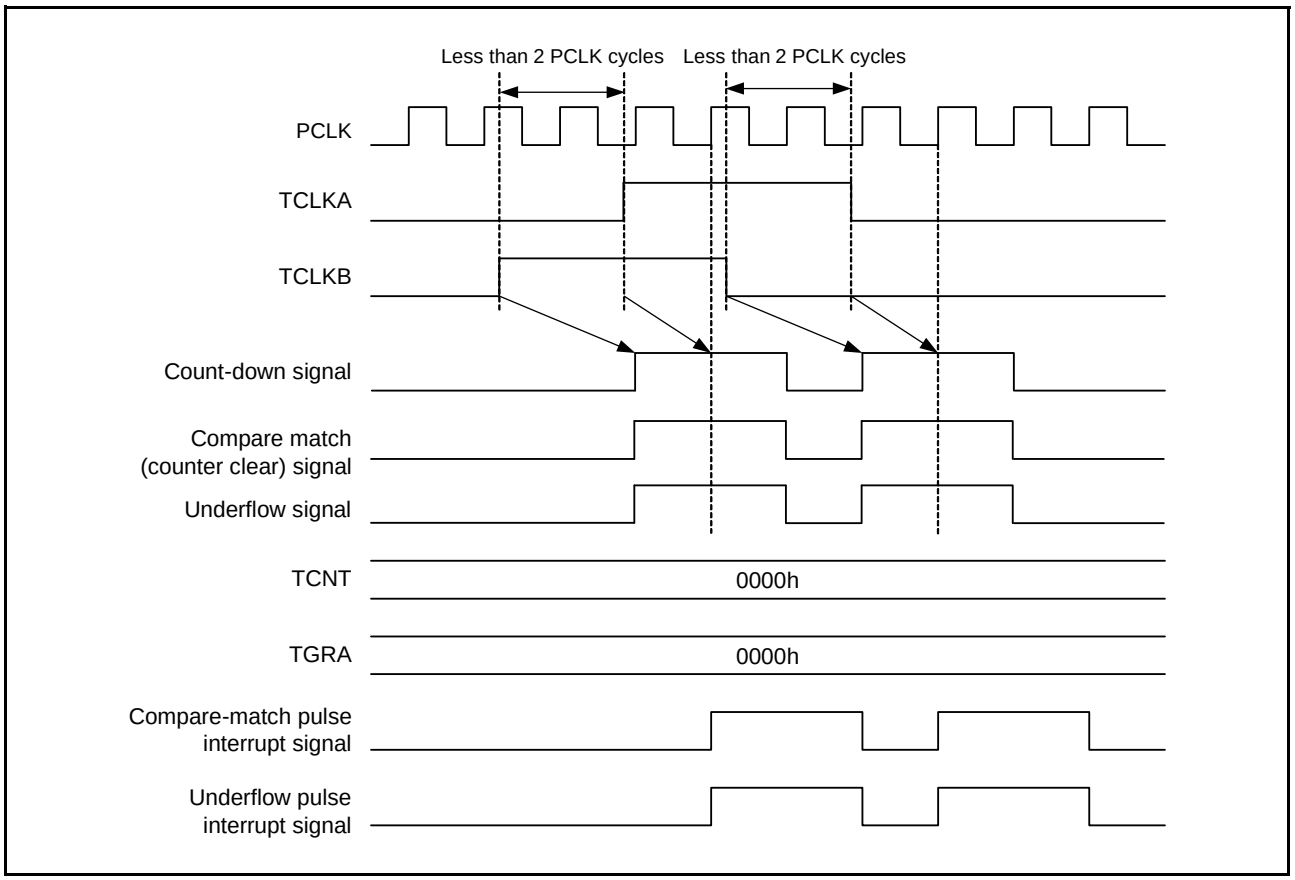


Figure 27.55 Continuous Output of Underflow Pulse Interrupt Signal

27.11 Event Link Operation

27.11.1 Event Signal Output to ELC

The TPU uses the ELC (event link controller) to perform link operation to the previously specified module using the interrupt request signal as the event signal.
The event signal can be output regardless of the setting of the corresponding interrupt request enable bits (TGIEA, TGIEB, TGIEC, or TGIED, and TCIEV, or TCIEU).

27.11.2 Event Signal Input from ELC

The TPU can perform any of the following three operations using the event link setting register of the ELC (event link controller).

(1) Start Counting

When an event signal is input while the TPU count start operation is selected, the CSTn bit in TSTRA register (the timer start register) is set to 1 and counting starts.
However, if this event is generated for the channels when the CSTn bit is set to 1, the event is ignored.
Table 27.30 lists the TSTRA.CSTn bit used for each channel.
Figure 27.56 shows the timing of the count start operation.

For details on the setting procedure to start counting, see section 27.3.1, (1) Counter Operation.

Table 27.30 Correspondence between Channels and TSTRA.CSTn bit

Channel No.	TSTRA.CSTn bit
TPU0	CST0
TPU1	CST1
TPU2	CST2
TPU3	CST3

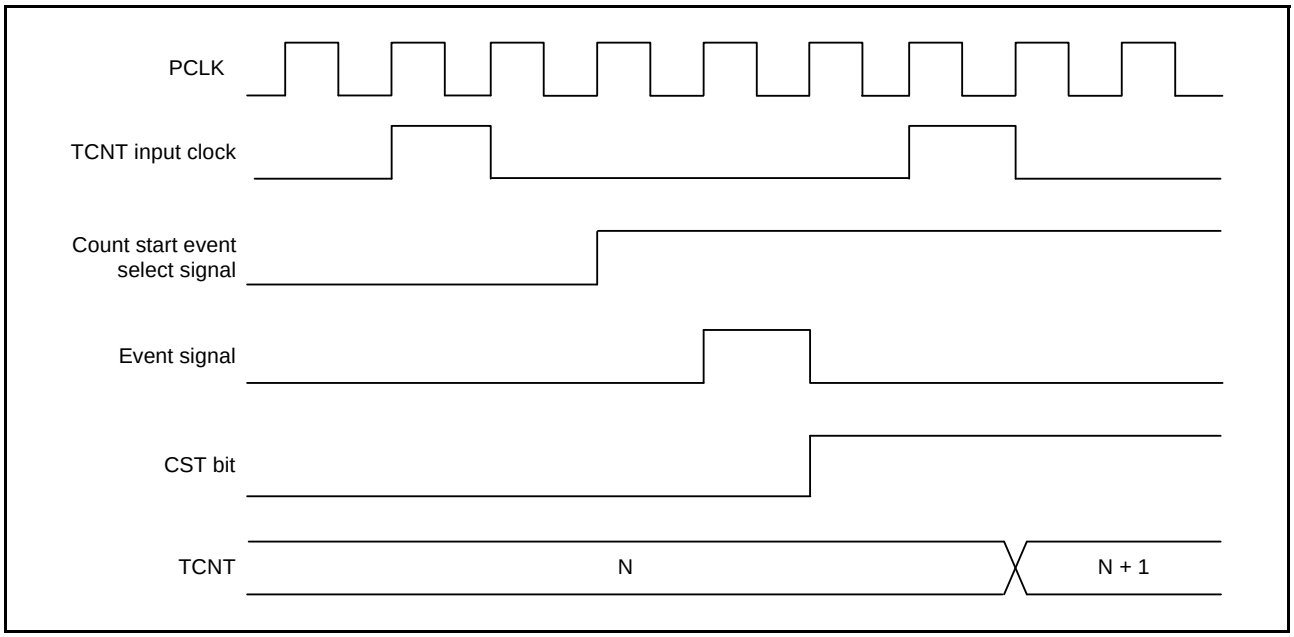


Figure 27.56 Start Counting on Input of the Event Signal

(2) Restart Counting

When an event signal is input while the TPU count restart operation is selected, the value of TCNTn counter (the timer start register) is returned to its initial value (0000h). If the CSTn bit in TSTRA register (the timer start register) is set to 1, however, counting operation will then continue.
Table 27.30 lists the TSTRA.CSTn bit used for each channel.
Figure 27.57 shows the timing of the count restart operation.

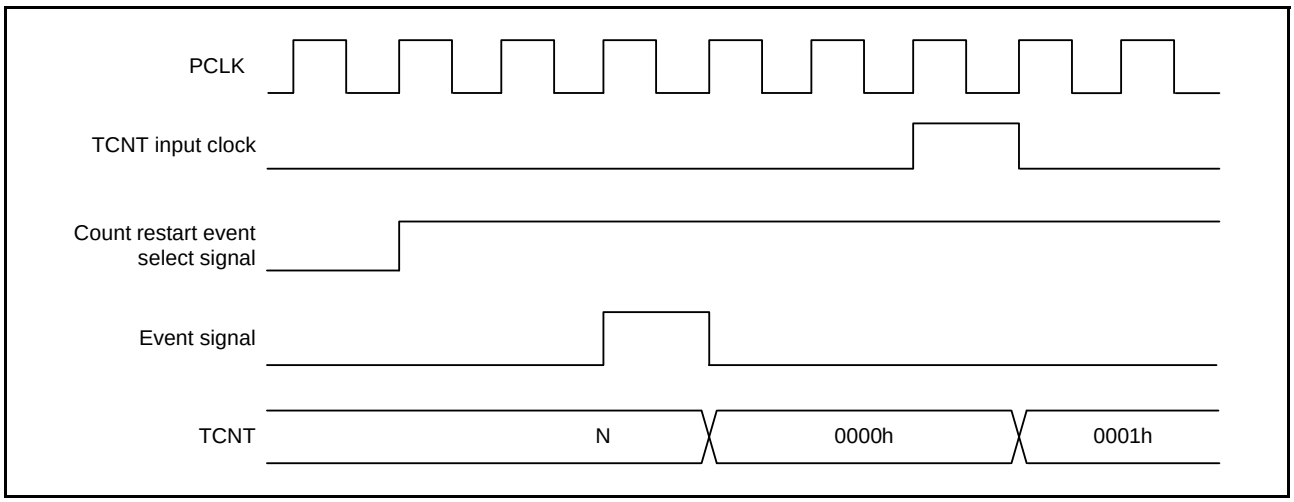


Figure 27.57 Restart Counting on Input of the Event Signal

(3) Input Capture Operation

When an event signal is input while the TPU input capture operation is selected, the value of TCNT counter (the timer count register) for the corresponding channel is captured in TGR register (the timer general register). When using input capture due to the event link, set the bit in TIOR (the timer I/O control register) to specify input capture, and then set the CSTn bit in TSTRA register (the timer start register) to 1 to start counting.
Table 27.31 lists TGR register and TIOR register bits used for each channel. For the TSTRA.CSTn bit used for each channel, see Table 27.30.
Figure 27.58 shows the timing of input capture operation.
When input capture operation by event linking is selected, the setting of TIOR register and the corresponding input capture (the linkage of the TIOCnA pin (input capture pin) input with the specific operation of other channels) are not effective (this also applies when the event select signal is set to 1 at the same time).

For details on the setting procedure for input capture, see section 27.3.1, (3) Input Capture Function.

Table 27.31 TGR and TIOR Used for Input Capture by ELC

Channel No.	Capture Destination Registers	Bits in TIOR
TPU0	TGRA register (channel 0)	IOA[3:0] bits (TIORH0)
TPU1	TGRA register (channel 1)	IOA[3:0] bits (TIOR1)
TPU2	TGRA register (channel 2)	IOA[3:0] bits (TIOR2)
TPU3	TGRA register (channel 3)	IOA[3:0] bits (TIORH3)

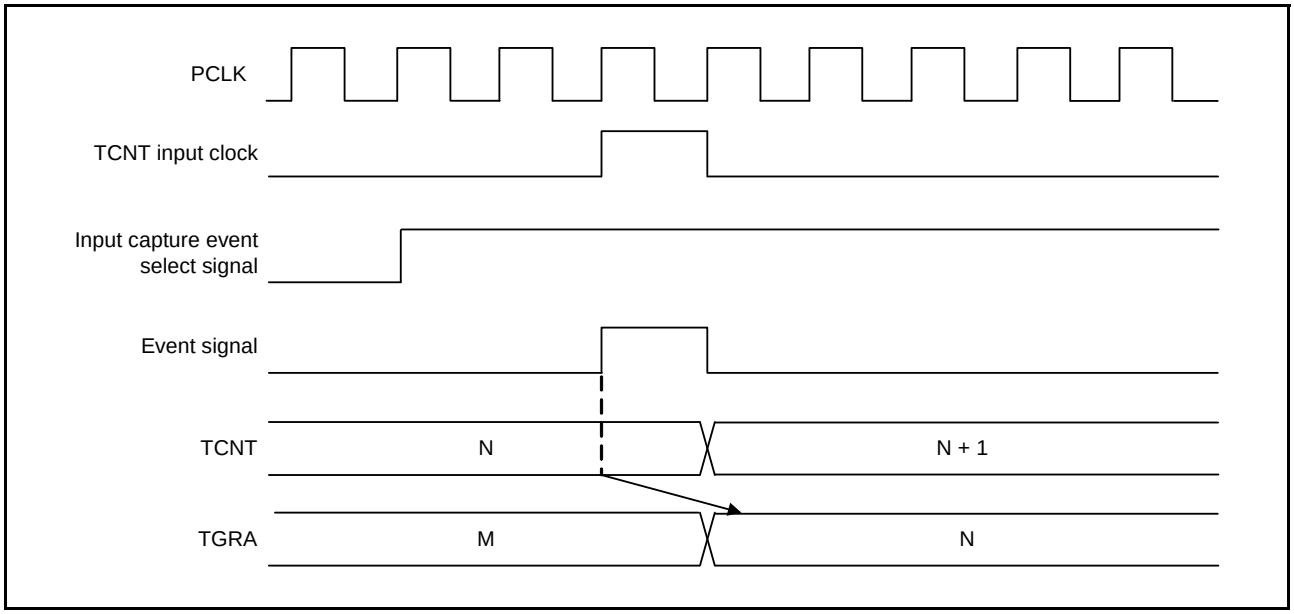


Figure 27.58 Input Capture on Input of the Event Signal

27.11.3 Usage Notes on Operation on Input of the Event Signal

The followings are the notes on using the TPU for event link operations.

(1) Start Counting

When writing to the TSTRA.CSTn bit (the timer start register) and a counting start are in contention, writing to the CSTn bit does not proceed since setting of the CSTn bit to 1 in response to the event takes priority.
Figure 27.59 shows the timing in this case.
Furthermore, even when a counting start due to the event link is selected, CPU writing to the TSTRA.CSTn bit proceeds if the event signal is low.

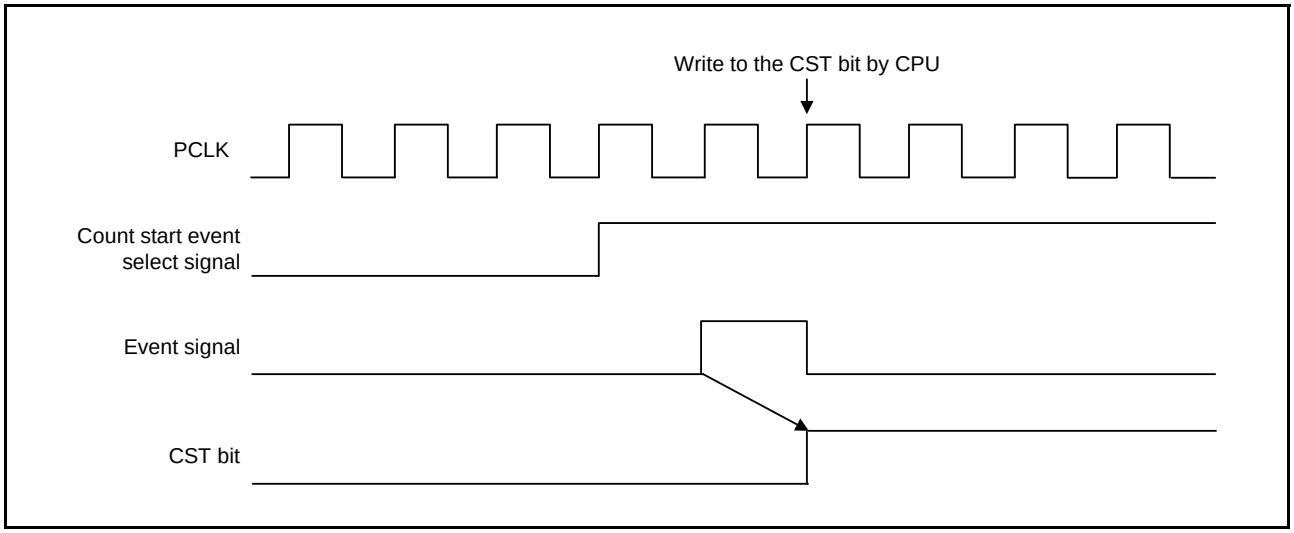


Figure 27.59 Conflict between Writing to the CSTn Bit and Counting Start

(2) Restart Counting

When a TCNTn counter (the timer count register) write cycle and a counting restart are in contention, writing to TCNTn counter does not proceed since the counter value initialization in response to the counting restart takes priority.
Figure 27.60 shows the timing in this case.
Furthermore, even when a counting restart due to the event link is selected, CPU writing to TCNTn counter proceeds if the event signal is low.

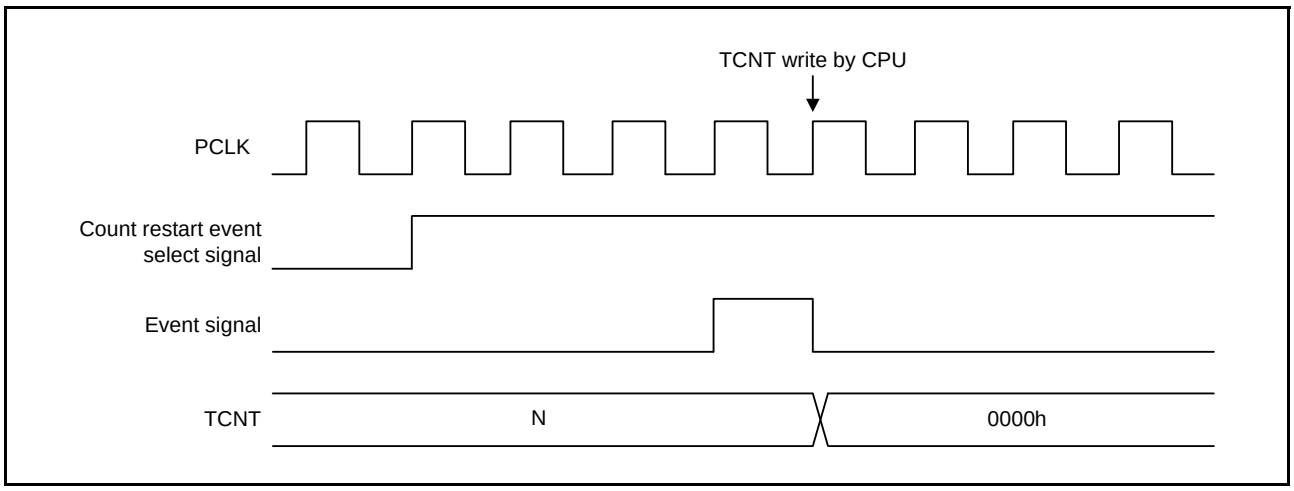


Figure 27.60 Conflict between TCNTn Write Cycle and Counting Restart

(3) Input Capture Operation

If a TGRA register (the timer general register) read/write cycle and input capture operation are in contention, operation proceeds as follows:

(a) Conflict between TGR Read Cycle and Input Capture

The internal data bus reads the data before input capture transfer.
Figure 27.61 shows the timing in this case.

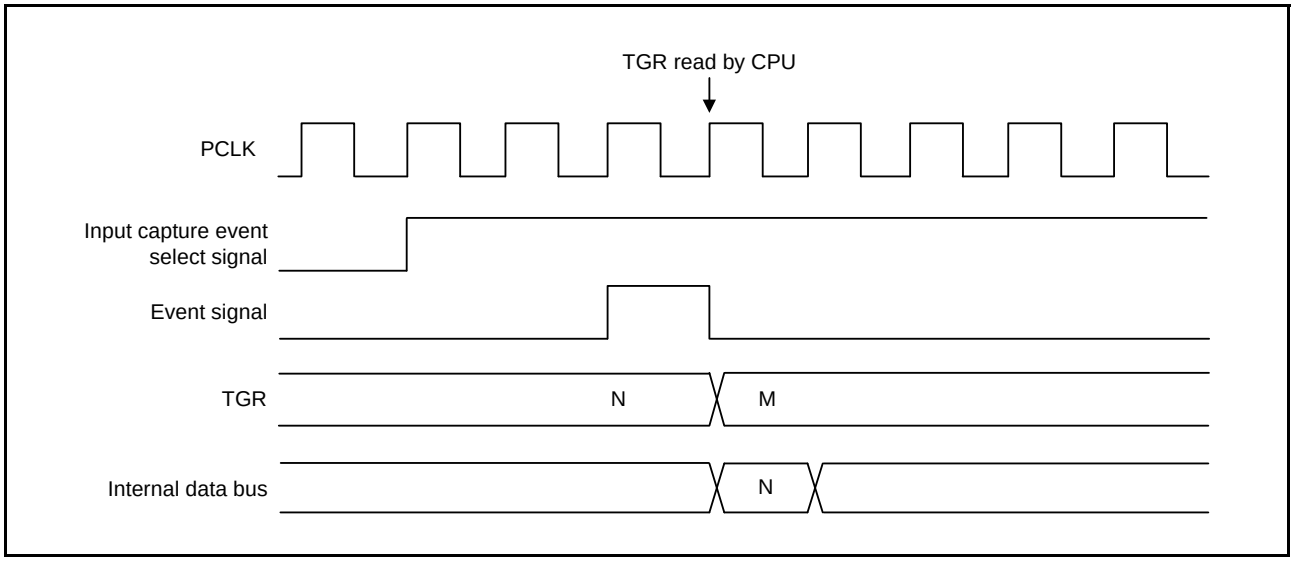


Figure 27.61 Conflict between TGR Read Cycle and Input Capture Operation

(b) Conflict between TGR Write Cycle and Input Capture

Writing to TGRA register does not proceed since input capture takes priority.

Figure 27.62 shows the timing in this case.

Furthermore, even when input capture operation due to the event link is selected, CPU writing to TGR proceeds if the event signal is low.

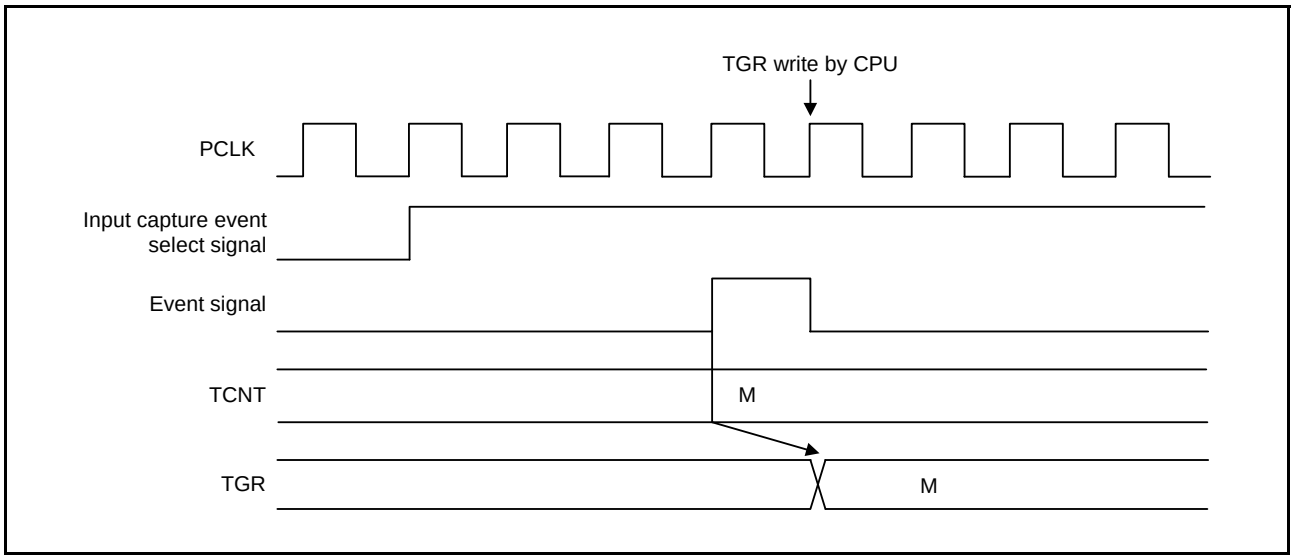


Figure 27.62 Conflict between TGR Write Cycle and Input Capture Operation

27.11.4 Notes on Output of the Event Signal

The followings are the notes on output of the event signal.

(1) Output of the Compare Match Event Signal

When the TGR register is set to 0000h, PCLK/1 is set as the counter clock, and compare match is set as the trigger for clearing of the counter clock (TCRn.TPSC[2:0] = 000b), the value of the TCNT remains 0000h, and the event output signal will be output continuously (i.e. its level will be flat) rather than output over a single cycle.
Figure 27.63 shows the timing for continuous output of the event output signal in response to a compare match.

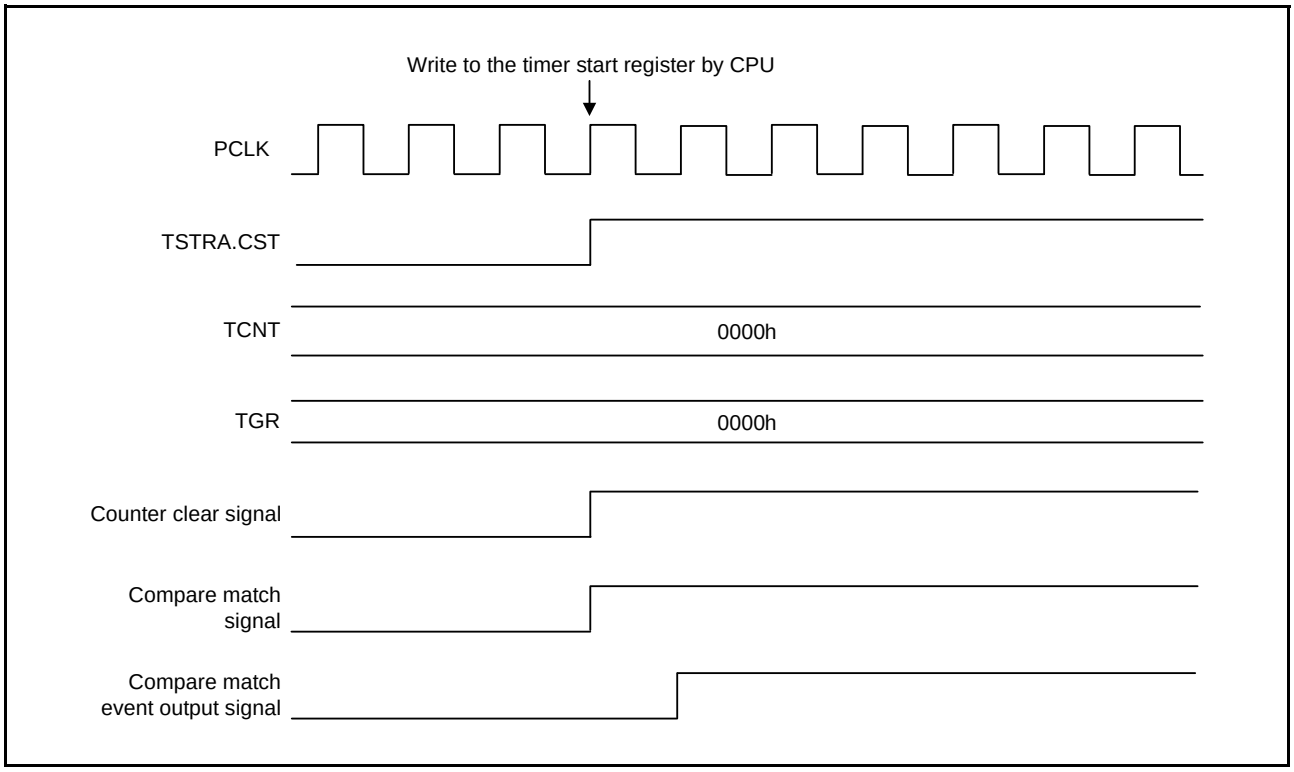


Figure 27.63 Continuous Output of the Compare Match Event Output Signal

(2) Output of the Underflow Event Signal

If two external clock signals' same direction edges to be phase counted are generated within two PCLK cycles in phase counting mode 1, with TGR being 0000h, and compare match set as the counter clear source, the TCNT counter remains 0000h, and a compare-match event signal and an underflow event signal are output continuously to form a flat signal level.

Figure 27.64 shows the timing for continuous output of the event output signal in response to underflow.

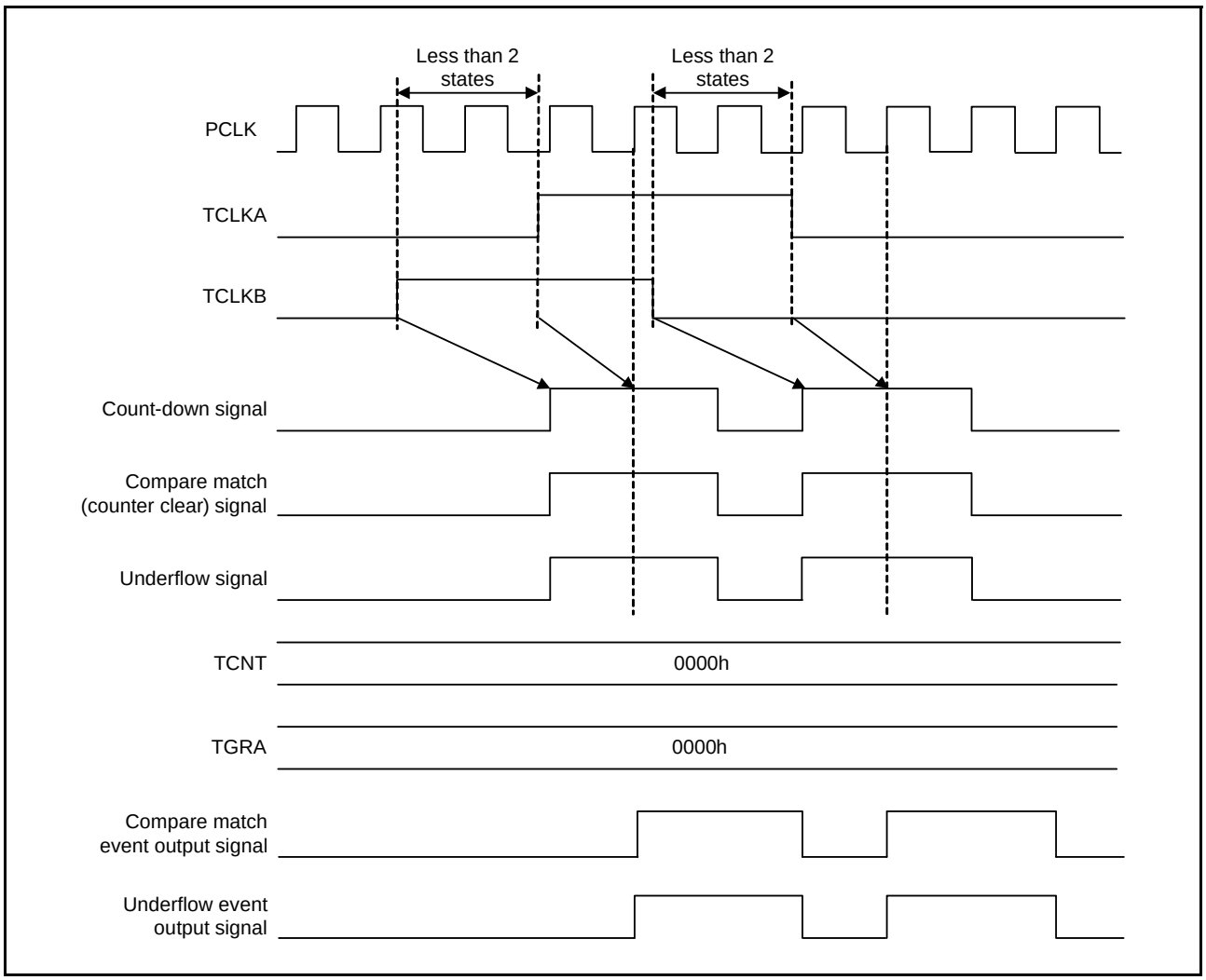


Figure 27.64 Continuous Output of the Underflow Event Output Signal

28. Programmable Pulse Generator (PPG)

The programmable pulse generator (PPG) generates pulse outputs by using the 16-bit timer pulse unit (TPU) and the multi-function timer pulse unit 3 (MTU3) as a time base.
This MCU has two PPG units, each of which controls up to 16 pulse output pins. The pulse outputs from the PPGs are divided into 4-bit groups that can operate all simultaneously and independently.

28.1 Overview

Table 28.1 lists the specifications of the PPG and Table 28.2 lists PPG functions.
Figure 28.1 and Figure 28.2 show block diagrams of the PPGs.

Table 28.1 Specifications of PPG

Item	Specifications
Number of output bits	Up to 32 bits*1
Pulse output	• Two units, each capable of output through four pin groups • Output trigger signals are selectable. • Non-overlapping operation is possible. • Inverted output is selectable.
Output data transfer	Can operate together with the DTC and DMAC (when TPU and MTU3 interrupts are in use)
Power consumption reducing function	Module-stop state can be set for each unit.

Note 1. When setting PPG output trigger in MTU3, make settings so that PCLKA run at the same frequency as PCLKB.

Table 28.2 List of PPG Functions

Item			PPG0	PPG1
PPG output trigger	MTU3 channels 0 to 3 (MTU0 to MTU3)*1	Compare match	○	○
		Input capture	○	○
	TPU (unit 0) channels 0 to 3 (TPU0 to TPU3)	Compare match	—	○
		Input capture	—	○
Non-overlapping operation			○	○
Output data transfer	DTC		○	○
	DMAC		○	○
Selecting inverted output			○	○
Setting the module-stop state*2			The MSTPA11 bit in MSTPCRA	The MSTPA10 bit in MSTPCRA

○: Possible
—: Not possible
Note 1. When setting PPG output trigger in MTU3, make settings so that PCLKA run at the same frequency as PCLKB.
Note 2. For details, see section 11, Low Power Consumption.

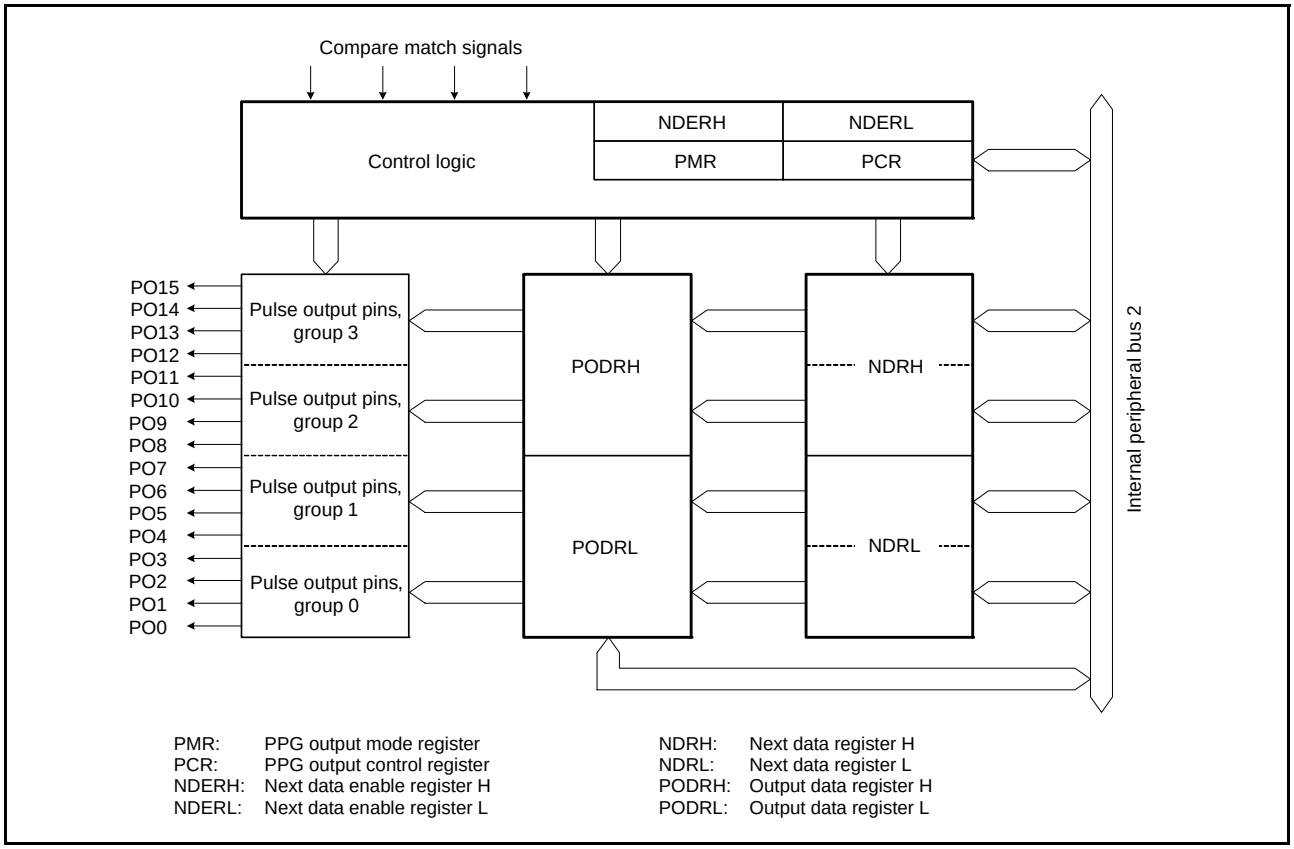


Figure 28.1 Block Diagram of PPG0

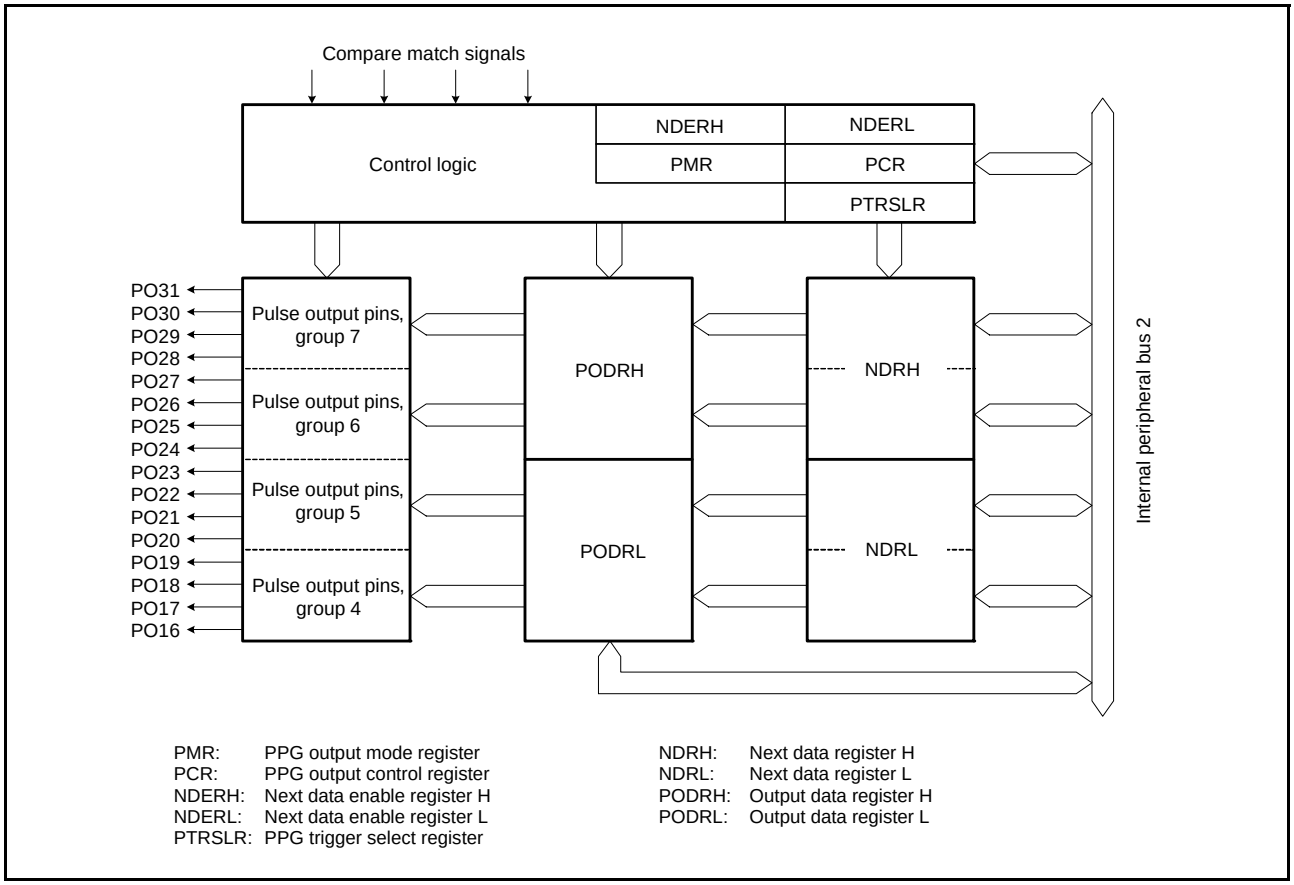


Figure 28.2 Block Diagram of PPG1

Table 28.3 lists the pin configuration of the PPG.

Table 28.3 Pin Configuration of PPG

Unit	Pin Name	I/O	Function
PPG0	PO0	Output	Group 0 pulse output
	PO1	Output	
	PO2	Output	
	PO3	Output	
	PO4	Output	Group 1 pulse output
	PO5	Output	
	PO6	Output	
	PO7	Output	
	PO8	Output	Group 2 pulse output
	PO9	Output	
	PO10	Output	
	PO11	Output	
	PO12	Output	Group 3 pulse output
	PO13	Output	
	PO14	Output	
	PO15	Output	
PPG1	PO16	Output	Group 4 pulse output
	PO17	Output	
	PO18	Output	
	PO19	Output	
	PO20	Output	Group 5 pulse output
	PO21	Output	
	PO22	Output	
	PO23	Output	
	PO24	Output	Group 6 pulse output
	PO25	Output	
	PO26	Output	
	PO27	Output	
	PO28	Output	Group 7 pulse output
	PO29	Output	
	PO30	Output	
	PO31	Output	

28.2 Register Descriptions

28.2.1 PPG Trigger Select Register (PTRSLR)

Address(es): 0008 81F0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PTRSL
Value after reset:	0	0	0	0	0	0	0	1

- PPG1.PTRSLR

Bit	Symbol	Bit Name	Description	R/W
b0	PTRSL	PPG Trigger Select	0: Selects the set of MTU0 to MTU3 of MTU3 as the trigger channels for PPG1. 1: Selects the set of TPU0 to TPU3 of TPU as the trigger channels for PPG1.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PTRSL Bit (PPG Trigger Select)

This bit selects either MTU0 to MTU3 of MTU3 or TPU0 to TPU3 of TPU as a set of trigger channels for PPG1. When this bit is set to 0, MTU0 to MTU3 of MTU3 are selected as a set of trigger channels for PPG1. When it is set to 1, TPU0 to TPU3 of TPU are selected as a set of trigger channels for PPG1.

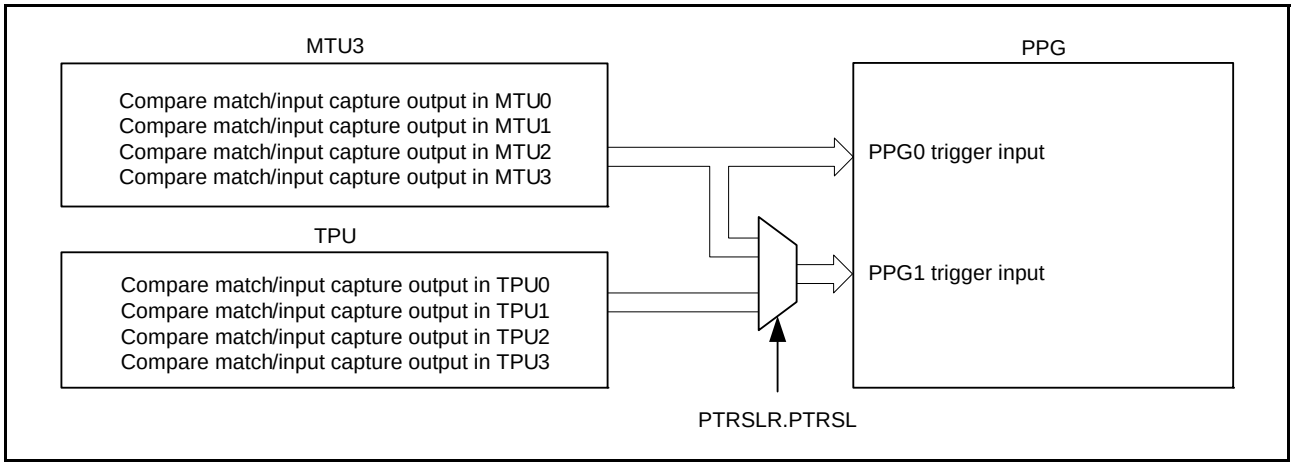


Figure 28.3 Block Diagram of PPG Trigger Selection

28.2.2 Next Data Enable Registers H (NDERH) Next Data Enable Registers L (NDERL)

Address(es): 0008 81E8h

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.NDERH	NDER 15	NDER 14	NDER 13	NDER 12	NDER 11	NDER 10	NDER9	NDER8
Value after reset:	0	0	0	0	0	0	0	0

Address(es): 0008 81E9h

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
Value after reset:	0	0	0	0	0	0	0	0

- PPG0.NDERH

Bit	Symbol	Bit Name	Description	R/W
b0	NDER8	Next Data Transfer Enable	0: Data transfer is disabled. 1: Data transfer is enabled.	R/W
b1	NDER9	Next Data Transfer Enable		R/W
b2	NDER 10	Next Data Transfer Enable		R/W
b3	NDER 11	Next Data Transfer Enable		R/W
b4	NDER 12	Next Data Transfer Enable		R/W
b5	NDER 13	Next Data Transfer Enable		R/W
b6	NDER 14	Next Data Transfer Enable		R/W
b7	NDER 15	Next Data Transfer Enable		R/W

PPG0.NDERH selects the pins (PO15 to PO8) for outputs of pulse from the PPG on a bit-by-bit basis.

NDERi Bits (Next Data Transfer Enable) (i = 15 to 8)

When these bits are set to 1, the PPG0.PCR specified trigger transfers data from the corresponding bit in PPG0.NDRH to the bit in PPG0.PODRH. When these bits are set to 0, the data are not transferred from PPG0.NDRH to PPG0.PODRH.

- PPG0.NDERL

Bit	Symbol	Bit Name	Description	R/W
b0	NDER0	Next Data Transfer Enable	0: Data transfer is disabled. 1: Data transfer is enabled.	R/W
b1	NDER1	Next Data Transfer Enable		R/W
b2	NDER2	Next Data Transfer Enable		R/W
b3	NDER3	Next Data Transfer Enable		R/W
b4	NDER4	Next Data Transfer Enable		R/W
b5	NDER5	Next Data Transfer Enable		R/W
b6	NDER6	Next Data Transfer Enable		R/W
b7	NDER7	Next Data Transfer Enable		R/W

PPG0.NDERL selects the pins (PO7 to PO0) for outputs of pulse from the PPG on a bit-by-bit basis.

NDERi Bits (Next Data Transfer Enable) (i = 7 to 0)

When these bits are set to 1, the PPG0.PCR specified trigger transfers data from the corresponding bit in PPG0.NDRL to the bit in PPG0.PODRL. When these bits are set to 0, the data are not transferred from PPG0.NDRL to PPG0.PODRL.

Address(es): 0008 81F8h

	b7	b6	b5	b4	b3	b2	b1	b0
PPG1.NDERH	NDER 31	NDER 30	NDER 29	NDER 28	NDER 27	NDER 26	NDER 25	NDER 24
Value after reset:	0	0	0	0	0	0	0	0

Address(es): 0008 81F9h

	b7	b6	b5	b4	b3	b2	b1	b0
PPG1.NDERL	NDER 23	NDER 22	NDER 21	NDER 20	NDER 19	NDER 18	NDER 17	NDER 16
Value after reset:	0	0	0	0	0	0	0	0

- PPG1.NDERH

Bit	Symbol	Bit Name	Description	R/W
b0	NDER 24	Next Data Transfer Enable	0: Data transfer is disabled. 1: Data transfer is enabled.	R/W
b1	NDER 25	Next Data Transfer Enable		R/W
b2	NDER 26	Next Data Transfer Enable		R/W
b3	NDER 27	Next Data Transfer Enable		R/W
b4	NDER 28	Next Data Transfer Enable		R/W
b5	NDER 29	Next Data Transfer Enable		R/W
b6	NDER 30	Next Data Transfer Enable		R/W
b7	NDER 31	Next Data Transfer Enable		R/W

PPG1.NDERH selects the pins (PO31 to PO24) for outputs of pulse from the PPG on a bit-by-bit basis.

NDERi Bits (Next Data Transfer Enable) (i = 31 to 24)

When these bits are set to 1, the PPG1.PCR specified trigger transfers data from the corresponding bit in PPG1.NDRH to the bit in PPG1.PODRH. When these bits are set to 0, the data are not transferred from PPG1.NDRH to PPG1.PODRH.

- PPG1.NDERL

Bit	Symbol	Bit Name	Description	R/W
b0	NDER 16	Next Data Transfer Enable	0: Data transfer is disabled. 1: Data transfer is enabled.	R/W
b1	NDER 17	Next Data Transfer Enable		R/W
b2	NDER 18	Next Data Transfer Enable		R/W
b3	NDER 19	Next Data Transfer Enable		R/W
b4	NDER 20	Next Data Transfer Enable		R/W
b5	NDER 21	Next Data Transfer Enable		R/W
b6	NDER 22	Next Data Transfer Enable		R/W
b7	NDER 23	Next Data Transfer Enable		R/W

PPG1.NDERL selects the pins (PO23 to PO16) for outputs of pulse from the PPG on a bit-by-bit basis.

NDERi Bits (Next Data Transfer Enable) (i = 23 to 16)

When these bits are set to 1, the PPG1.PCR specified trigger transfers data from the corresponding bit in PPG1.NDRL to the bit in PPG1.PODRL. When these bits are set to 0, the data are not transferred from PPG1.NDRL to PPG1.PODRL.

28.2.3 Output Data Registers H (PODRH) Output Data Registers L (PODRL)

Address(es): 0008 81EAh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8
Value after reset:	0	0	0	0	0	0	0	0

Address(es): 0008 81EBh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
Value after reset:	0	0	0	0	0	0	0	0

- PPG0.PODRH

Bit	Symbol	Bit Name	Description	R/W
b0	POD8	Output Data Register	0: The low level is output on the POi pin. 1: The high level is output on the POi pin. (i = 15 to 8)	R/W
b1	POD9	Output Data Register		R/W
b2	POD10	Output Data Register		R/W
b3	POD11	Output Data Register		R/W
b4	POD12	Output Data Register		R/W
b5	POD13	Output Data Register		R/W
b6	POD14	Output Data Register		R/W
b7	POD15	Output Data Register		R/W

PPG0.PODRH stores pulse output values. For bits corresponding to pins that have been set for pulse output by PPG0.NDERH, the output trigger transfers the values in PPG0.NDRH to this register.

PODi Bit (Output Data Register) (i = 15 to 8)

When an output trigger is generated during PPG operation, the values of bits for which data transfer is enabled in the PPG0.NDERH register are transferred from the PPG0.NDRH register to this register. Writing from the CPU is impossible while any of the NDERi (i = 15 to 8) bits in PPG0.NDERH is 1. The initial output level for the pulse signal can be set when the value in the PPG0.NDERH register is 00h.

- PPG0.PODRL

Bit	Symbol	Bit Name	Description	R/W
b0	POD0	Output Data Register	0: The low level is output on the POi pin. 1: The high level is output on the POi pin. (i = 7 to 0)	R/W
b1	POD1	Output Data Register		R/W
b2	POD2	Output Data Register		R/W
b3	POD3	Output Data Register		R/W
b4	POD4	Output Data Register		R/W
b5	POD5	Output Data Register		R/W
b6	POD6	Output Data Register		R/W
b7	POD7	Output Data Register		R/W

PPG0.PODRL stores pulse output values. For bits corresponding to pins that have been set for pulse output by PPG0.NDERL, the output trigger transfers the values in PPG0.NDRL to this register.

PODi Bit (Output Data Register) (i = 7 to 0)

When an output trigger is generated during PPG operation, the values of bits for which data transfer is enabled in the PPG0.NDERL register are transferred from the PPG0.NDRL register to this register. Writing from the CPU is impossible while any of the NDERi (i = 7 to 0) bits in PPG0.NDERL is 1. The initial output level for the pulse signal can be set when the value in the PPG0.NDERL register is 00h.

Address(es): 0008 81FAh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG1.PODRH	POD31	POD30	POD29	POD28	POD27	POD26	POD25	POD24
Value after reset:	0	0	0	0	0	0	0	0

Address(es): 0008 81FBh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG1.PODRL	POD23	POD22	POD21	POD20	POD19	POD18	POD17	POD16
Value after reset:	0	0	0	0	0	0	0	0

- PPG1.PODRH

Bit	Symbol	Bit Name	Description	R/W
b0	POD24	Output Data Register	0: The low level is output on the POi pin. 1: The high level is output on the POi pin. (i = 31 to 24)	R/W
b1	POD25	Output Data Register		R/W
b2	POD26	Output Data Register		R/W
b3	POD27	Output Data Register		R/W
b4	POD28	Output Data Register		R/W
b5	POD29	Output Data Register		R/W
b6	POD30	Output Data Register		R/W
b7	POD31	Output Data Register		R/W

PPG1.PODRH stores pulse output values. For bits corresponding to pins that have been set for pulse output by PPG1.NDERH, the output trigger transfers the values in PPG1.NDRH to this register.

PODi Bit (Output Data Register) (i = 31 to 24)

When an output trigger is generated during PPG operation, the values of bits for which data transfer is enabled in the PPG1.NDERH register are transferred from the PPG1.NDRH register to this register. Writing from the CPU is impossible while any of the NDERi (i = 31 to 24) bits in PPG1.NDERH is 1. The initial output level for the pulse signal can be set when the value in the PPG1.NDERH register is 00h.

- PPG1.PODRL

Bit	Symbol	Bit Name	Description	R/W
b0	POD16	Output Data Register	0: The low level is output on the POi pin. 1: The high level is output on the POi pin. (i = 23 to 16)	R/W
b1	POD17	Output Data Register		R/W
b2	POD18	Output Data Register		R/W
b3	POD19	Output Data Register		R/W
b4	POD20	Output Data Register		R/W
b5	POD21	Output Data Register		R/W
b6	POD22	Output Data Register		R/W
b7	POD23	Output Data Register		R/W

PPG1.PODRL stores pulse output values. For bits corresponding to pins that have been set for pulse output by PPG1.NDERL, the output trigger transfers the values in PPG1.NDRL to this register.

PODi Bit (Output Data Register) (i = 23 to 16)

When an output trigger is generated during PPG operation, the values of bits for which data transfer is enabled in the PPG1.NDERL register are transferred from the PPG1.NDRL register to this register. Writing from the CPU is impossible while any of the NDERi (i = 23 to 16) bits in PPG1.NDERL is 1. The initial output level for the pulse signal can be set when the value in the PPG1.NDERL register is 00h.

28.2.4 Next Data Registers H (NDRH) Next Data Registers L (NDRL)

Address(es): 0008 81ECh, 0008 81EEh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.NDRH	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
Value after reset:	0	0	0	0	0	0	0	0

Address(es): 0008 81EDh, 0008 81EFh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.NDRL	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
Value after reset:	0	0	0	0	0	0	0	0

- PPG0.NDRH

PPG0.NDRH stores the next data for pulse output. The PPG0.NDRH address differs depending on whether pulse output groups have the same output trigger or different output triggers.

(1) When pulse output groups 2 and 3 have the same output trigger

If pulse output groups 2 and 3 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time.

- Pulse output groups 2 and 3: 0008 81ECh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR8	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRH.	R/W
b1	NDR9	Next Data Register		R/W
b2	NDR10	Next Data Register		R/W
b3	NDR11	Next Data Register		R/W
b4	NDR12	Next Data Register		R/W
b5	NDR13	Next Data Register		R/W
b6	NDR14	Next Data Register		R/W
b7	NDR15	Next Data Register		R/W

Note: The address (0008 81EEh) to which PPG0.NDRH address has not been assigned is read as FFh, and cannot be modified.

(2) When pulse output groups 2 and 3 have different output triggers

If pulse output groups 2 and 3 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses.

- Pulse output group 3: 0008 81ECh

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b4	NDR12	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRH.	R/W
b5	NDR13	Next Data Register		R/W
b6	NDR14	Next Data Register		R/W
b7	NDR15	Next Data Register		R/W

- Pulse output group 2: 0008 81EEh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR8	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRH.	R/W
b1	NDR9	Next Data Register		R/W
b2	NDR10	Next Data Register		R/W
b3	NDR11	Next Data Register		R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

- PPG0.NDRL

PPG0.NDRL stores the next data for pulse output. The PPG0.NDRL address differs depending on whether pulse output groups have the same output trigger or different output triggers.

(1) When pulse output groups 0 and 1 have the same output trigger

If pulse output groups 0 and 1 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time.

- Pulse output groups 0 and 1: 0008 81EDh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR0	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRL.	R/W
b1	NDR1	Next Data Register		R/W
b2	NDR2	Next Data Register		R/W
b3	NDR3	Next Data Register		R/W
b4	NDR4	Next Data Register		R/W
b5	NDR5	Next Data Register		R/W
b6	NDR6	Next Data Register		R/W
b7	NDR7	Next Data Register		R/W

Note: The address (0008 81EFh) to which PPG0.NDRL address has not been assigned is read as FFh, and cannot be modified.

(2) When pulse output groups 0 and 1 have different output triggers

If pulse output groups 0 and 1 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses.

- Pulse output group 1: 0008 81EDh

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b4	NDR4	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRL.	R/W
b5	NDR5	Next Data Register		R/W
b6	NDR6	Next Data Register		R/W
b7	NDR7	Next Data Register		R/W

- Pulse output group 0: 0008 81EFh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR0	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRL.	R/W
b1	NDR1	Next Data Register		R/W
b2	NDR2	Next Data Register		R/W
b3	NDR3	Next Data Register		R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Address(es): 0008 81FCh, 0008 81FEh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG1.NDRH	NDR31	NDR30	NDR29	NDR28	NDR27	NDR26	NDR25	NDR24
Value after reset:	0	0	0	0	0	0	0	0

Address(es): 0008 81FDh, 0008 81FFh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG1.NDRL	NDR23	NDR22	NDR21	NDR20	NDR19	NDR18	NDR17	NDR16
Value after reset:	0	0	0	0	0	0	0	0

- PPG1.NDRH

PPG1.NDRH stores the next data for pulse output. The PPG1.NDRH address differs depending on whether pulse output groups have the same output trigger or different output triggers.

(1) When pulse output groups 6 and 7 have the same output trigger

If pulse output groups 6 and 7 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time.

- Pulse output groups 6 and 7: 0008 81FCh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR24	Next Data Register	The output trigger specified by PPG1.PCR transfers the values in this register to the corresponding bits in PPG1.PODRH.	R/W
b1	NDR25	Next Data Register		R/W
b2	NDR26	Next Data Register		R/W
b3	NDR27	Next Data Register		R/W
b4	NDR28	Next Data Register		R/W
b5	NDR29	Next Data Register		R/W
b6	NDR30	Next Data Register		R/W
b7	NDR31	Next Data Register		R/W

Note: The address (0008 81FEh) to which PPG1.NDRH address has not been assigned is read as FFh, and cannot be modified.

(2) When pulse output groups 6 and 7 have different output triggers

If pulse output groups 6 and 7 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses.

- Pulse output group 7: 0008 81FCh

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b4	NDR28	Next Data Register	The output trigger specified by PPG1.PCR transfers the values in this register to the corresponding bits in PPG1.PODRH.	R/W
b5	NDR29	Next Data Register		R/W
b6	NDR30	Next Data Register		R/W
b7	NDR31	Next Data Register		R/W

- Pulse output group 6: 0008 81FEh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR24	Next Data Register	The output trigger specified by PPG1.PCR transfers the values in this register to the corresponding bits in PPG1.PODRH.	R/W
b1	NDR25	Next Data Register		R/W
b2	NDR26	Next Data Register		R/W
b3	NDR27	Next Data Register		R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

- PPG1.NDRL

PPG1.NDRL stores the next data for pulse output. The PPG1.NDRL address differs depending on whether pulse output groups have the same output trigger or different output triggers.

(1) When pulse output groups 4 and 5 have the same output trigger

If pulse output groups 4 and 5 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time.

- Pulse output groups 4 and 5: 0008 81FDh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR16	Next Data Register	The output trigger specified by PPG1.PCR transfers the values in this register to the corresponding bits in PPG1.PODRL.	R/W
b1	NDR17	Next Data Register		R/W
b2	NDR18	Next Data Register		R/W
b3	NDR19	Next Data Register		R/W
b4	NDR20	Next Data Register		R/W
b5	NDR21	Next Data Register		R/W
b6	NDR22	Next Data Register		R/W
b7	NDR23	Next Data Register		R/W

Note: The address (0008 81FFh) to which PPG1.NDRL address has not been assigned is read as FFh, and cannot be modified.

(2) When pulse output groups 4 and 5 have different output triggers

If pulse output groups 4 and 5 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses.

- Pulse output group 5: 0008 81FDh

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b4	NDR20	Next Data Register	The output trigger specified by PPG1.PCR transfers the values in this register to the corresponding bits in PPG1.PODRL.	R/W
b5	NDR21	Next Data Register		R/W
b6	NDR22	Next Data Register		R/W
b7	NDR23	Next Data Register		R/W

- Pulse output group 4: 0008 81FFh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR16	Next Data Register	The output trigger specified by PPG1.PCR transfers the values in this register to the corresponding bits in PPG1.PODRL.	R/W
b1	NDR17	Next Data Register		R/W
b2	NDR18	Next Data Register		R/W
b3	NDR19	Next Data Register		R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

28.2.5 PPG Output Control Register (PCR)

Address(es): PPG0.PCR 0008 81E6h, PPG1.PCR 0008 81F6h

	b7	b6	b5	b4	b3	b2	b1	b0
	G3CMS[1:0]		G2CMS[1:0]		G1CMS[1:0]		G0CMS[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

- PPG0.PCR

Bit	Symbol	Bit Name	Description	R/W
b1, b0	G0CMS[1:0]	Group 0 Compare Match Select	b1 b0 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3	R/W
b3, b2	G1CMS[1:0]	Group 1 Compare Match Select	b3 b2 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3	R/W
b5, b4	G2CMS[1:0]	Group 2 Compare Match Select	b5 b4 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3	R/W
b7, b6	G3CMS[1:0]	Group 3 Compare Match Select	b7 b6 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3	R/W

- PPG1.PCR

Bit	Symbol	Bit Name	Description	R/W
b1, b0	G0CMS[1:0]	Group 4 Compare Match Select	• When the PTRSL bit in PPG1.PTRSLR is set to 0. b1 b0 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3 • When the PTRSL bit in PPG1.PTRSLR is set to 1. b1 b0 0 0: Compare match in TPU0 0 1: Compare match in TPU1 1 0: Compare match in TPU2 1 1: Compare match in TPU3	R/W
b3, b2	G1CMS[1:0]	Group 5 Compare Match Select	• When the PTRSL bit in PPG1.PTRSLR is set to 0. b3 b2 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3 • When the PTRSL bit in PPG1.PTRSLR is set to 1. b3 b2 0 0: Compare match in TPU0 0 1: Compare match in TPU1 1 0: Compare match in TPU2 1 1: Compare match in TPU3	R/W

Bit	Symbol	Bit Name	Description	R/W
b5, b4	G2CMS[1:0]	Group 6 Compare Match Select	• When the PTRSL bit in PPG1.PTRSLR is set to 0. b5 b4 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3 • When the PTRSL bit in PPG1.PTRSLR is set to 1. b5 b4 0 0: Compare match in TPU0 0 1: Compare match in TPU1 1 0: Compare match in TPU2 1 1: Compare match in TPU3	R/W
b7, b6	G3CMS[1:0]	Group 7 Compare Match Select	• When the PTRSL bit in PPG1.PTRSLR is set to 0. b7 b6 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3 • When the PTRSL bit in PPG1.PTRSLR is set to 1. b7 b6 0 0: Compare match in TPU0 0 1: Compare match in TPU1 1 0: Compare match in TPU2 1 1: Compare match in TPU3	R/W

PPGn.PCR (n = 0, 1) selects pulse output trigger signals on a group-by-group basis. For details on output trigger selection, see section 28.2.6, PPG Output Mode Register (PMR).

28.2.6 PPG Output Mode Register (PMR)

Address(es): PPG0.PMR 0008 81E7h, PPG1.PMR 0008 81F7h

	b7	b6	b5	b4	b3	b2	b1	b0
	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV
Value after reset:	1	1	1	1	0	0	0	0

- PPG0.PMR

Bit	Symbol	Bit Name	Description	R/W
b0	G0NOV	Group 0 Non-Overlap	0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3)	R/W
b1	G1NOV	Group 1 Non-Overlap	0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3)	R/W
b2	G2NOV	Group 2 Non-Overlap	0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3)	R/W
b3	G3NOV	Group 3 Non-Overlap	0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3)	R/W
b4	G0INV	Group 0 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b5	G1INV	Group 1 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b6	G2INV	Group 2 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b7	G3INV	Group 3 Output Polarity Change	0: Inverted output 1: Direct output	R/W

- PPG1.PMR

Bit	Symbol	Bit Name	Description	R/W
b0	G0NOV	Group 4 Non-Overlap	• When the PPG1.PTRSLR.PTRSL bit is 0 0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3) • When the PPG1.PTRSLR.PTRSL bit is 1 0: Normal operation (Output values updated on compare match A in the selected TPUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected TPUn) (n = 0 to 3)	R/W
b1	G1NOV	Group 5 Non-Overlap	• When the PPG1.PTRSLR.PTRSL bit is 0 0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3) • When the PPG1.PTRSLR.PTRSL bit is 1 0: Normal operation (Output values updated on compare match A in the selected TPUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected TPUn) (n = 0 to 3)	R/W
b2	G2NOV	Group 6 Non-Overlap	• When the PPG1.PTRSLR.PTRSL bit is 0 0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3) • When the PPG1.PTRSLR.PTRSL bit is 1 0: Normal operation (Output values updated on compare match A in the selected TPUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected TPUn) (n = 0 to 3)	R/W
b3	G3NOV	Group 7 Non-Overlap	• When the PPG1.PTRSLR.PTRSL bit is 0 0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3) • When the PPG1.PTRSLR.PTRSL bit is 1 0: Normal operation (Output values updated on compare match A in the selected TPUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected TPUn) (n = 0 to 3)	R/W

Bit	Symbol	Bit Name	Description	R/W
b4	G0INV	Group 4 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b5	G1INV	Group 5 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b6	G2INV	Group 6 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b7	G3INV	Group 7 Output Polarity Change	0: Inverted output 1: Direct output	R/W

PPGn.PMR (n = 0, 1) selects the pulse output mode of the PPG on a group-by-group basis.
While inverted output is selected, a low-level pulse is output when the values in PPGn.PODRH and PPGn.PODRL are 1, and a high-level pulse is output when the values in PPGn.PODRH and PPGn.PODRL are 0.
In addition, when non-overlapping operation is selected, the PPG updates its output values on compare match A or B in an MTU3 or TPU channel that functions as an output trigger.
For details, see section 28.3.4, Non-Overlapping Pulse Output.

28.3 Operation

Figure 28.4 shows a schematic diagram of the PPG.
PPG pulse output is enabled when the corresponding bits in PPGn.NDERH and PPGn.NDERL (n = 0, 1) are set to 1 (data transfer is enabled).
An initial output value is determined by the initial settings in the corresponding PPGn.PODRH and PPGn.PODRL.
When the compare match event selected in PPGn.PCR occurs, the output values are updated by transfer of the values in the corresponding PPGn.NDRH and PPGn.NDRL to PPGn.PODRH and PPGn.PODRL, respectively.
Consecutive output of up to 16 bits of data is possible by writing new output data to PPGn.NDRH and PPGn.NDRL before the next compare match.

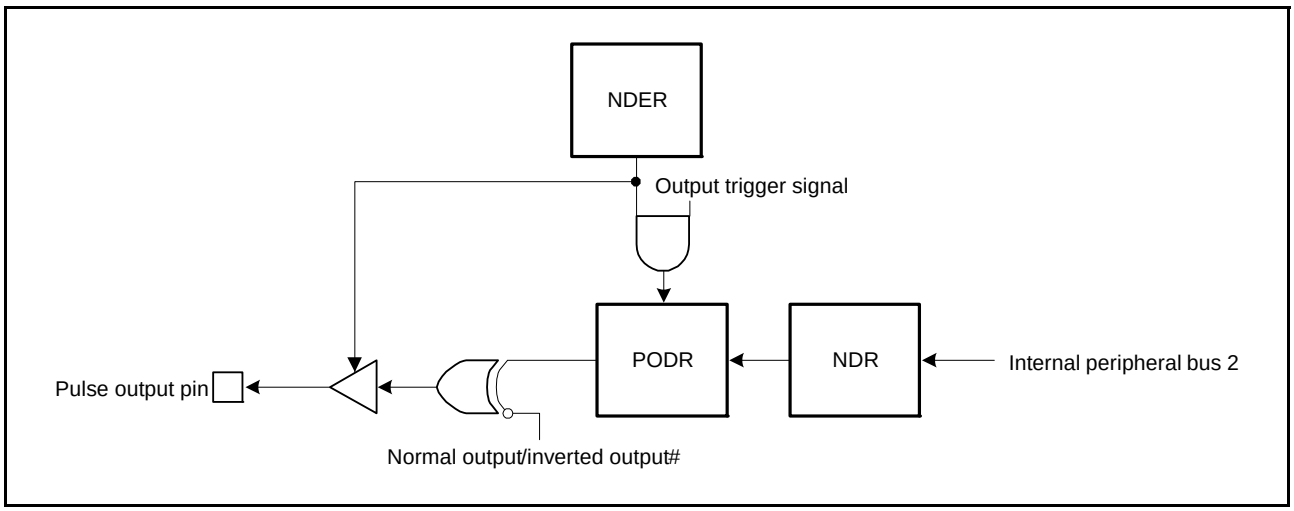


Figure 28.4 Schematic Diagram of PPG

28.3.1 Output Timing

When the selected compare match event occurs while pulse output is enabled, the values in PPGn.NDRH and PPGn.NDRL (n = 0, 1) are transferred to PPGn.PODRH and PPGn.PODRL, respectively, and then output on the corresponding pins.

Figure 28.5 shows the timing of the above operation. In this case, the timing when compare match A triggers normal output from groups 2 and 3 is shown.

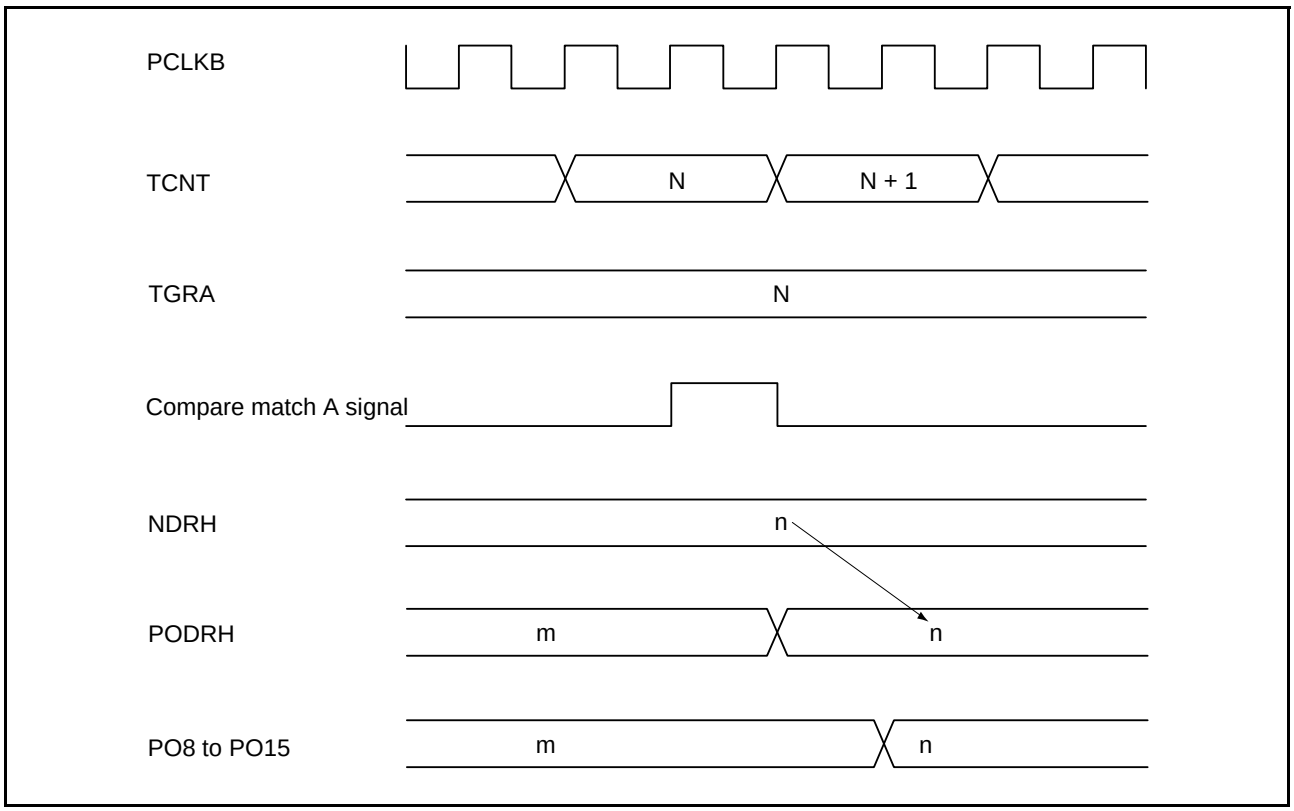
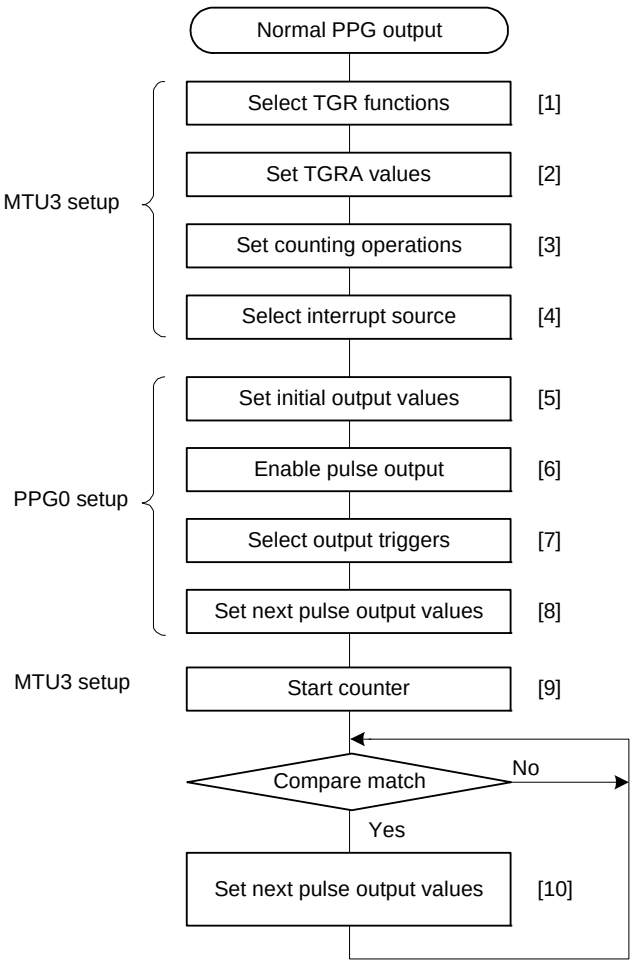


Figure 28.5 Timing of Transfer and Output of the Values in NDR (Example)

28.3.2 Sample Setup Procedure for Normal Pulse Output

Figure 28.6 and Figure 28.7 show sample procedures for setting normal pulse output.

(1) PPG0 Setting



[1] Use the TIOR register for MTU3 to set the TGRA register to an output compare register (toggle output).

[2] Set the PPG output trigger period.

[3] Select the counter clock source with the TCR.TPSC[2:0] bits. Select the counter clear source with the TCR.CCLR[2:0] bits.

[4] Enable the TGIA interrupt in TIER register. The DTC or DMAC can also be set up to transfer data to NDR.

[5] Set the initial output values in PODR register.

[6] Set the bits In NDER register for the pins to be used for pulse output to 1.

[7] Select the MTU3 compare match event to be used as the output trigger in PCR register.

[8] Set the next pulse output values in NDR register.

[9] Set the TSTR.CST bit to 1 to start the TCNT counter.

[10] At each TGIA interrupt, set the next output values in NDR register.

Figure 28.6 Sample Setup Procedure for Normal Pulse Output (PPG0 Setting)

(2) PPG1 Setting

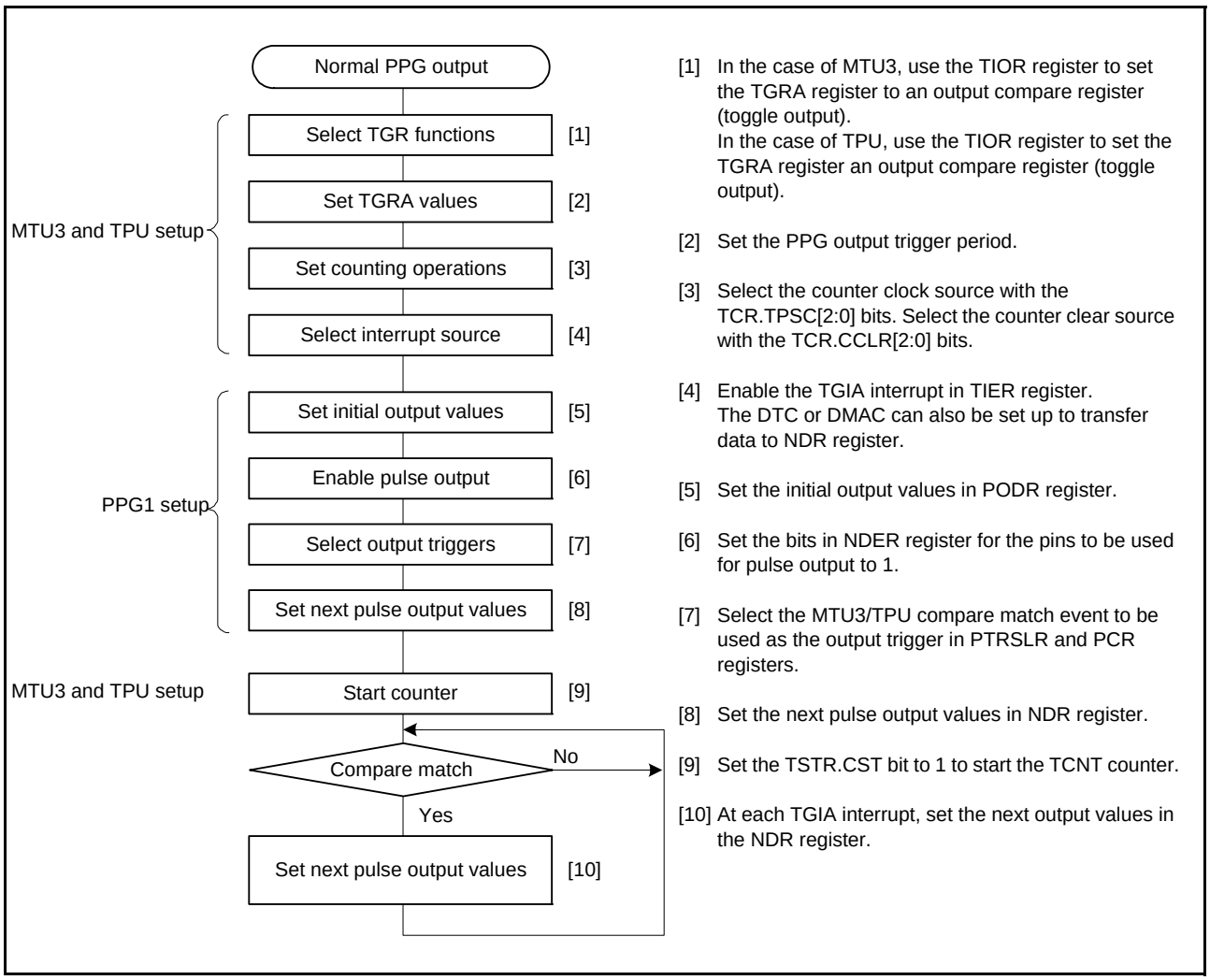


Figure 28.7 Sample Setup Procedure for Normal Pulse Output (PPG1 Setting)

28.3.3 Example of Normal Pulse Output (Example of Five-Phase Pulse Output)

Figure 28.8 shows an example in which pulse output from the PPG0 is used for cyclic five-phase pulse output.

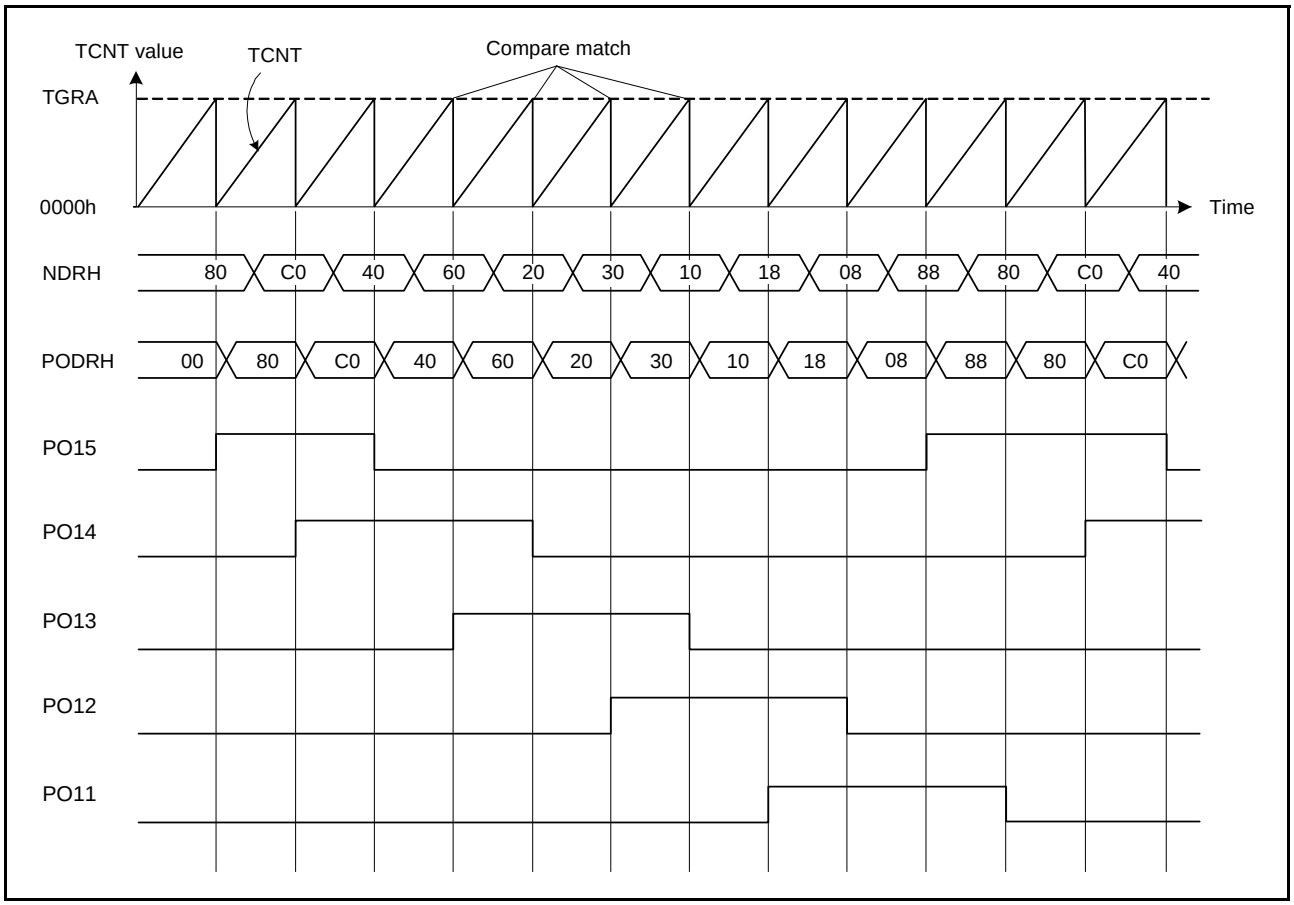


Figure 28.8 Example of Normal Pulse Output (Example of Five-Phase Pulse Output)

1. Set an output compare register of the MTUn.TGRA (n = 0 to 3) of MTU3 so that the corresponding compare match signal is the output trigger. Set a cycle in TGRA so that the counter will be cleared by compare match A. Set the MTUn.TIER.TGIEA bit to 1 to enable the compare match/input capture A (TGIAn) interrupt.
2. Write F8h to PPG0.NDERH, and set the G3CMS[1:0] and G2CMS[1:0] bits in PPG0.PCR to select the respective compare matches in the MTUn selected in the previous step to be the output triggers. Write output data 80h to PPG0.NDRH.
3. The timer counter in the MTU3 starts. When compare match A occurs, the values in PPG0.NDRH are transferred to PPG0.PODRH and output. The TGIAn interrupt handling routine writes the next output data C0h to PPG0.NDRH.
4. Five-phase pulse output (one or two phases active at a time) can be obtained subsequently by writing 40h, 60h, 20h, 30h, 10h, 18h, 08h, 88h... at successive TGIAn interrupts.
   If the DTC or DMAC is set for activation by the TGIAn interrupt, pulse output can be obtained without imposing a load on the CPU.

28.3.4 Non-Overlapping Pulse Output

During non-overlapping operation, data transfer from PPGn.NDRH and PPGn.NDRL (n = 0, 1) to PPGn.PODRH and PPGn.PODRL is performed as follows.

- On compare match A, the values in PPGn.NDRH and PPGn.NDRL are always transferred to PPGn.PODRH and PPGn.PODRL.
- On compare match B, data transfer proceeds for bits in PPGn.NDRH and PPGn.NDRL that have the value 0. It does not proceed for bits having the value 1.
  Figure 28.9 shows the non-overlapping pulse output operation.

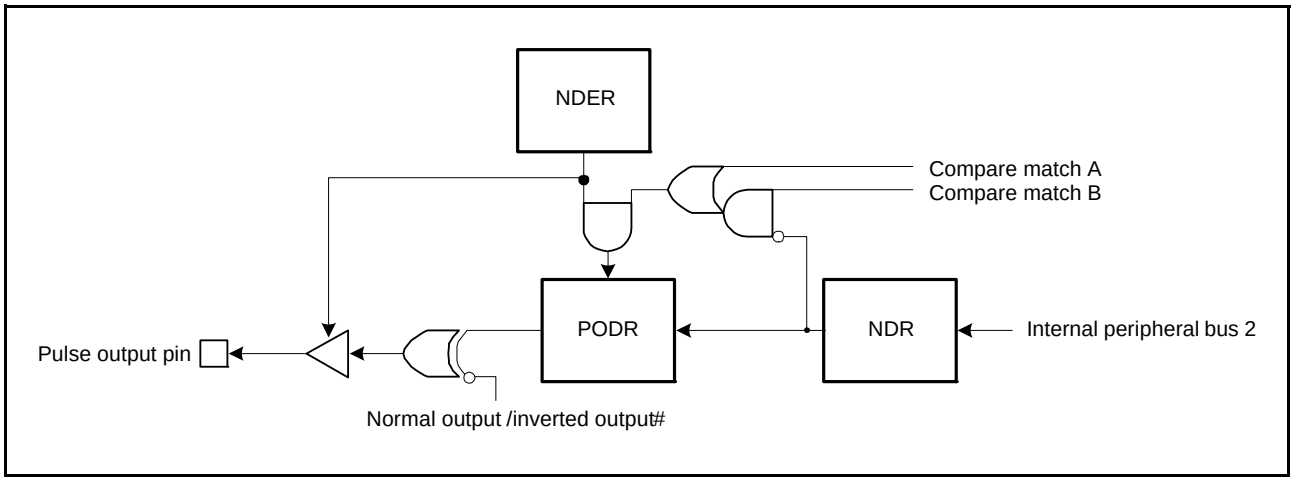


Figure 28.9 Non-Overlapping Pulse Output

Therefore, compare match B before compare match A allows 0-valued data to be transferred in advance of 1-valued data. Do not change the values in PPGn.NDRH and PPGn.NDRL during the interval from compare match B to compare match A (the non-overlap margin).
To transfer 0-valued data in advance of 1-valued data, write the next data to PPGn.NDRH and PPGn.NDRL from within the TGIA interrupt handling routine or by using a TGIA interrupt to activate transfer by the DTC or DMAC. In any case, the next data must be written before the next compare match B occurs.
Figure 28.10 shows the timing of the above procedure.

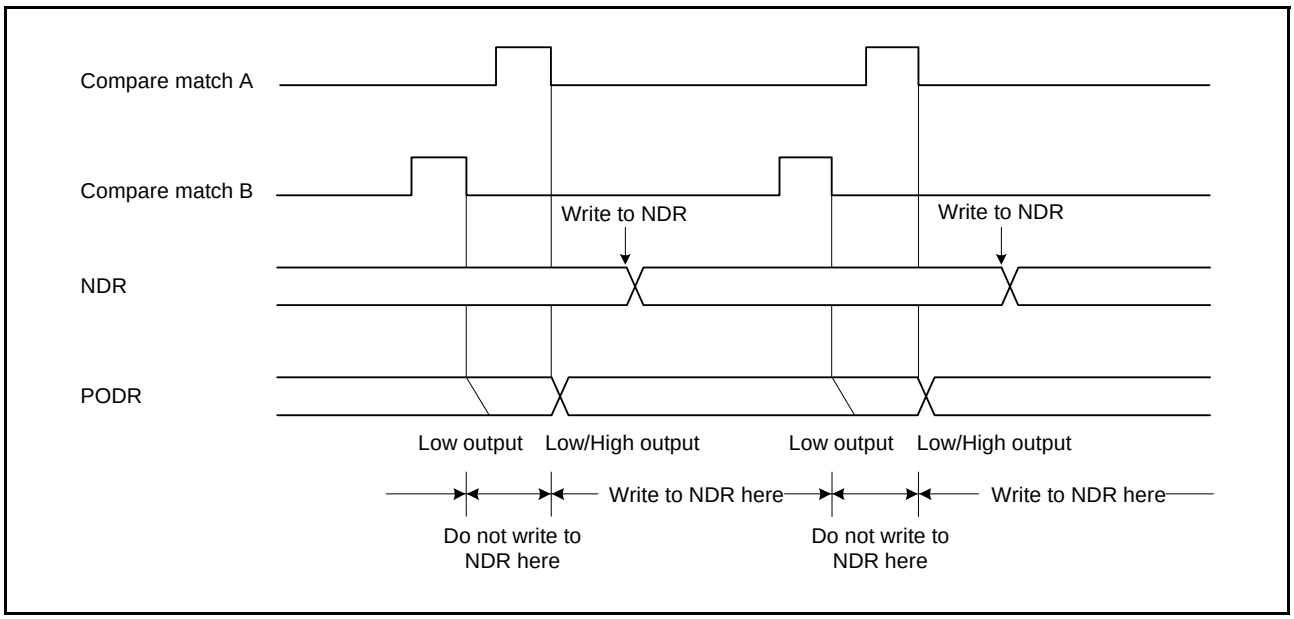


Figure 28.10 Non-Overlapping Operation and Write Timing to PPGn.NDRH and PPGn.NDRL

28.3.5 Sample Setup Procedure for Non-Overlapping Pulse Output

Figure 28.11 and Figure 28.12 show sample procedures for setting up non-overlapping pulse outputs.

(1) PPG0 Setting

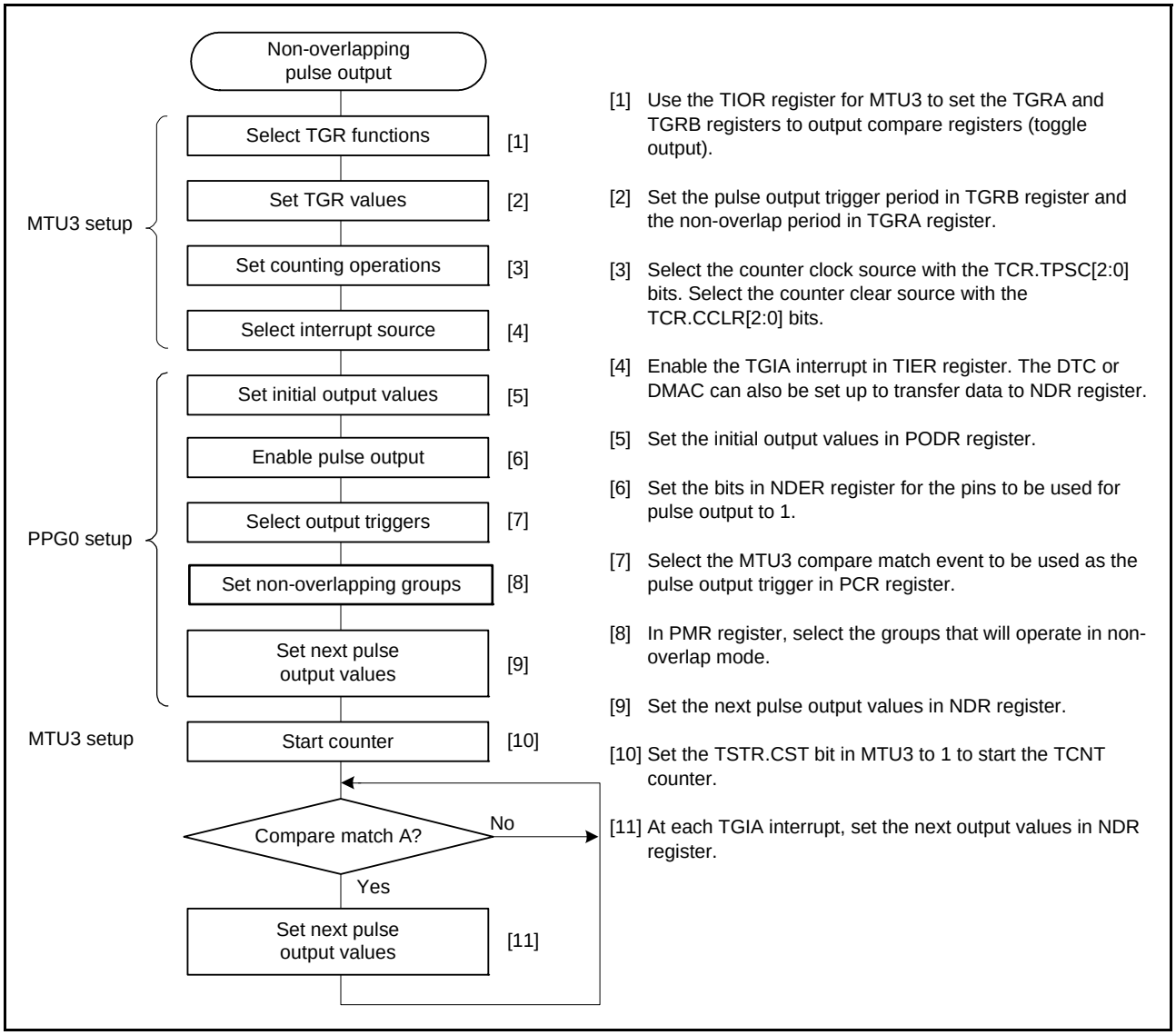


Figure 28.11 Sample Setup Procedure for Non-Overlapping Pulse Output (PPG0 Setting)

(2) PPG1 Setting

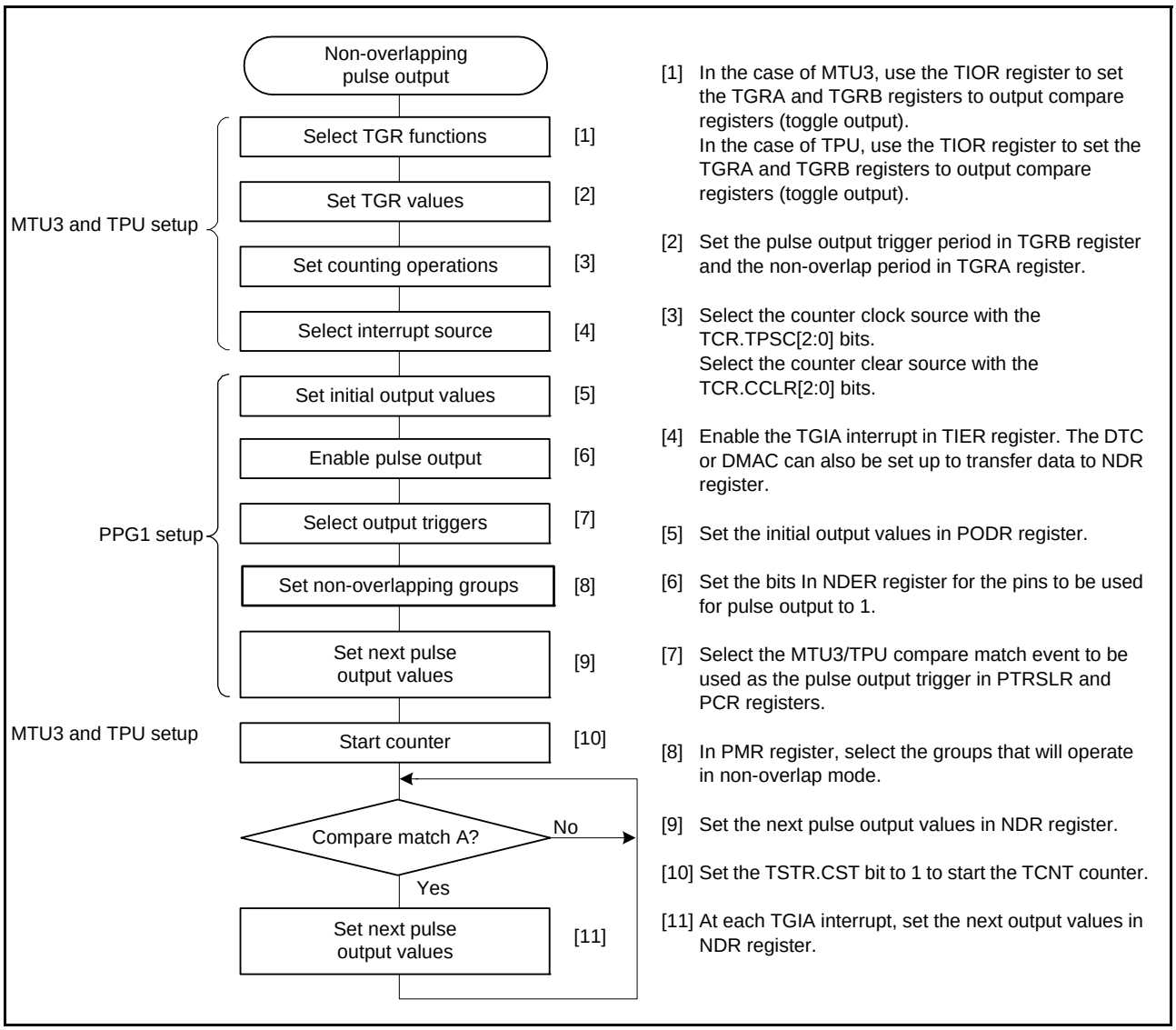


Figure 28.12 Sample Setup Procedure for Non-Overlapping Pulse Output (PPG1 Setting)

28.3.6 Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output)

Figure 28.13 shows an example in which pulse output from the PPG0 is used for four-phase complementary non-overlapping pulse output.

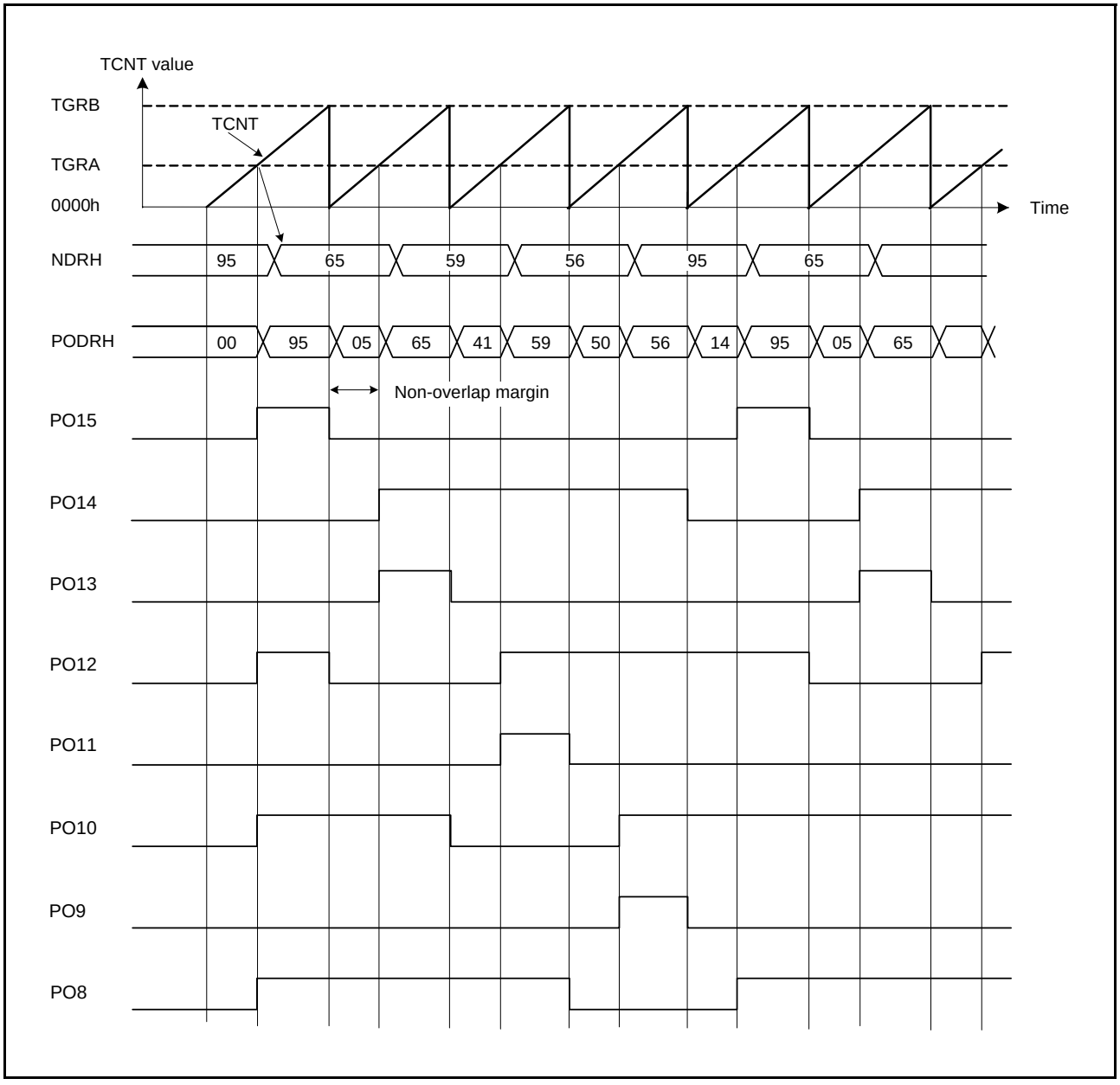


Figure 28.13 Example of Non-Overlapping Pulse Output (Four-Phase Complementary Non-Overlapping Output)

1. Set output compare registers of the MTUn.TGRA and MTUn.TGRB (n = 0 to 3) of MTU3 so that the corresponding compare match signals are the output triggers. Set the trigger period in TGRB and the non-overlap margin in TGRA, and set the counter to be cleared by compare match B. Set the MTUn.TIER.TGIEA bit to 1 to enable the compare match/input capture A (TGIAn) interrupt.
2. Write FFh in PPG0.NDERH, and set the G3CMS[1:0] and G2CMS[1:0] bits in PPG0.PCR to select the respective compare matches in the MTUn selected in the previous step to be the output triggers.
   Set the G3NOV and G2NOV bits in PPG0.PMR to 1 to select non-overlapping outputs. Write output data 95h in PPG0.NDRH.
3. The timer counter in the MTU3 starts. When a compare match with TGRB occurs, outputs change from high to low. When a compare match with TGRA occurs, outputs change from low to high (the change from low to high is delayed by the value set in TGRA).
   The TGIAn interrupt handling routine writes the next output data 65h in PPG0.NDRH.
4. Four-phase complementary non-overlapping pulse output can be obtained subsequently by writing 59h, 56h, 95h... at successive TGIAn interrupts.
   If the DTC or DMAC is set for activation by the TGIAn interrupt, pulse output can be obtained without imposing a load on the CPU.

28.3.7 Inverted Pulse Output

When the G3INV, G2INV, G1INV, and G0INV bits in PPG0.PMR are cleared to 0, the values that are the inverse of the respective values in PPG0.PODRH and PPG0.PODRL can be output.

Figure 28.14 shows the outputs when the G3INV and G2INV bits are cleared to 0 in addition to the settings in Figure 28.13.

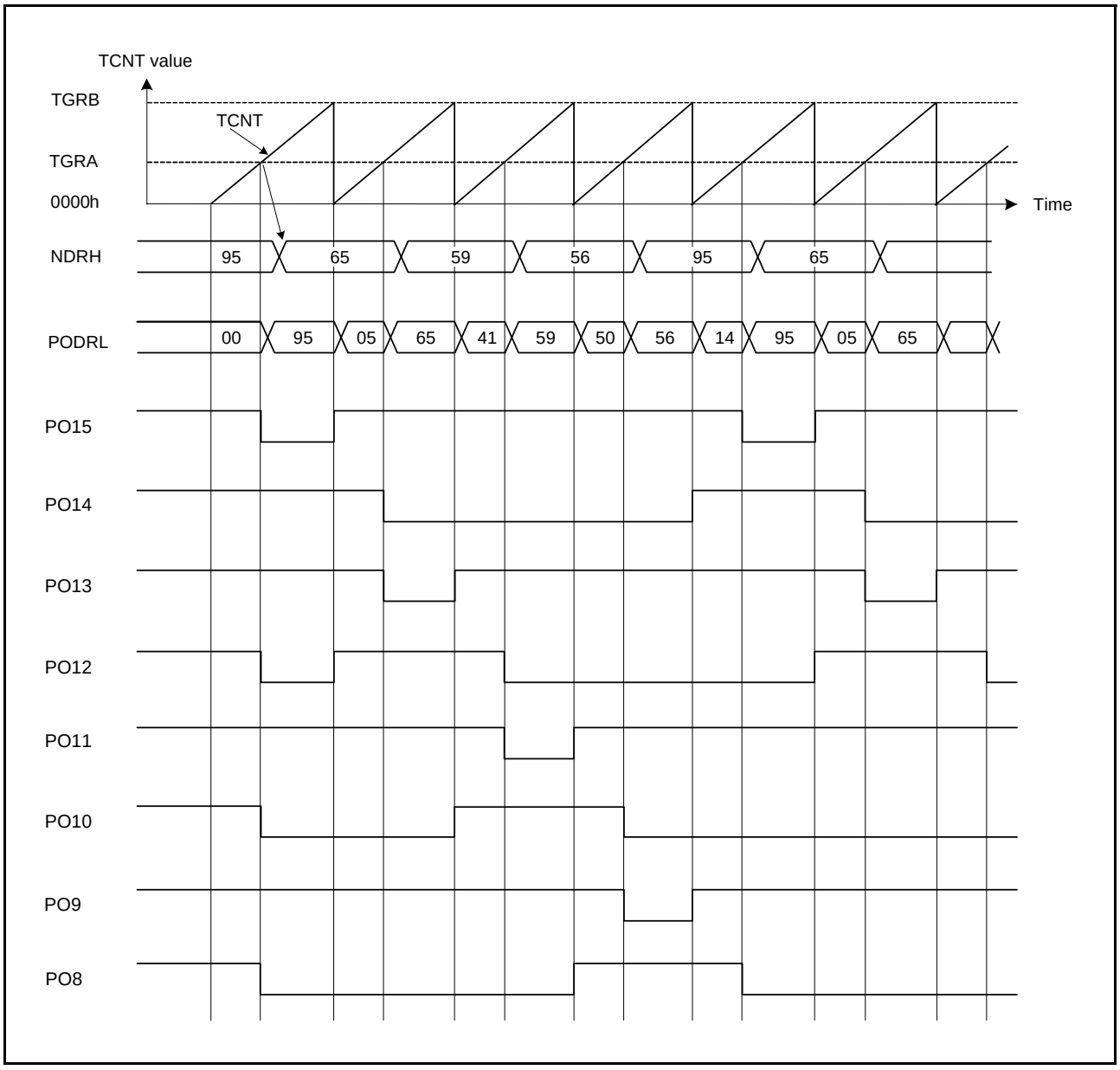


Figure 28.14 Inverted Pulse Output (Example)

28.3.8 Pulse Output Triggered by Input Capture

Pulse output from the PPG0 can be triggered by the MTU3 input capture as well as by compare match. When MTUn.TGRA (n = 0 to 3) functions as an input capture register in the MTU3 channel selected by PPG0.PCR, pulse output is triggered by the input capture signal.
Figure 28.15 shows the timing of pulse output triggered by input capture.

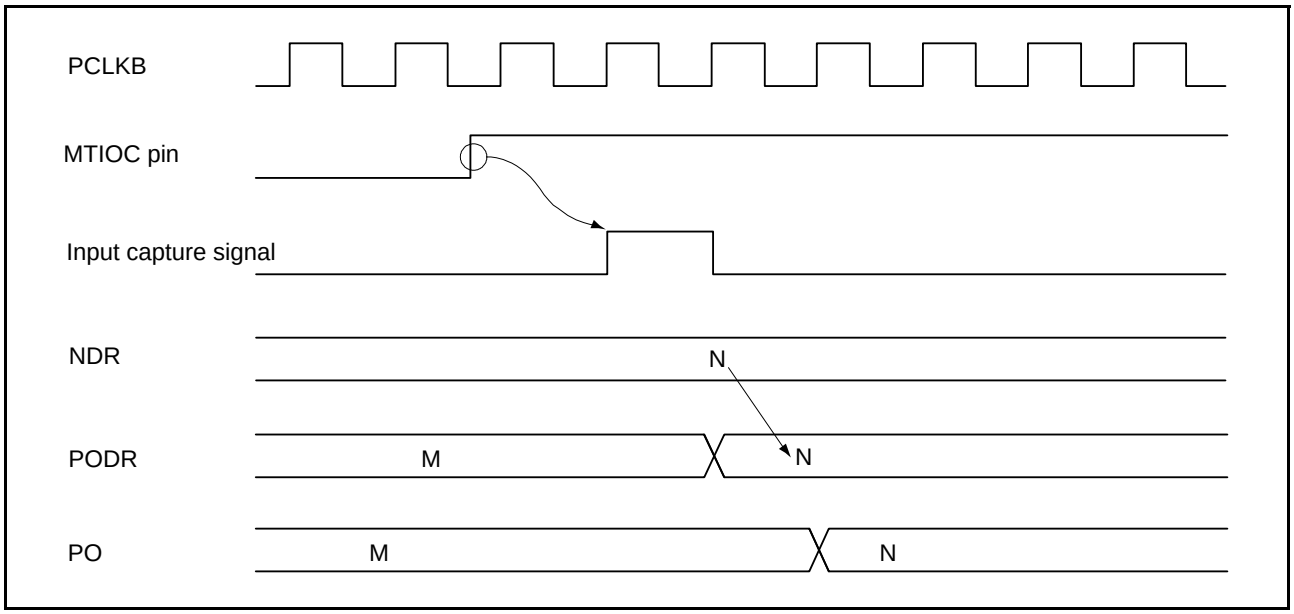


Figure 28.15 Timing of Pulse Output Triggered by Input Capture (Example)

28.4 Usage Note

28.4.1 Module-Stop Function Setting

Operation of the PPG can be disabled or enabled by the module stop control register. The initial setting is for operation of the PPG to be stopped. Register access is enabled by clearing module-stop state. For details, see section 11, Low Power Consumption.

29. 8-Bit Timer (TMR)

This MCU has two units (unit 0, unit 1) of an on-chip 8-bit timer (TMR) module that comprise two 8-bit counter channels, totaling four channels. The 8-bit timer module can be used to count external events and also be used as a multi-function timer in a variety of applications, such as generation of counter reset signal, interrupt requests, and pulse output with a desired duty cycle using a compare-match signal with two registers.
Unit 0 and unit 1 have the same functions, and can generate a baud rate clock for the SCI.
In this section, “PCLK” is used to refer to PCLKB.

29.1 Overview

Table 29.1 lists the specifications of the TMR. Table 29.2 lists the TMR functions.
Figure 29.1 shows a block diagram of the 8-bit timer module (unit 0), and Figure 29.2 shows that of the 8-bit timer module (unit 1).

Table 29.1 Specifications of TMR

Item	Description
Count clock	• Internal clock: PCLK/1, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, PCLK/8192 • External clock: external count clock
Number of channels	(8 bits × 2 channels) × 2 units
Compare match	• 8-bit mode (compare match A, compare match B) • 16-bit mode (compare match A, compare match B)
Counter clear	Selected by compare match A or B, or an external counter reset signal.
Timer output	Output pulses with a desired duty cycle or PWM output
Cascading of two channels	• 16-bit count mode 16-bit timer using TMR0 for the upper 8 bits and TMR1 for the lower 8 bits (TMR2 for the upper 8 bits and TMR3 for the lower 8 bits) • Compare match count mode TMR1 can be used to count TMR0 compare matches (TMR3 can be used to count TMR2 compare matches).
Interrupt sources	Compare match A, compare match B, and overflow
Event link function (Output)	Compare match A, compare match B, and overflow (TMR0 to TMR3)
Event link function (Input)	One of the following three operations proceeds in response to an event reception: (1) Counting start operation (TMR0 to TMR3) (2) Event counting operation (TMR0 to TMR3) (3) Counting restart operation (TMR0 to TMR3)
DTC activation	DTC can be activated by compare match A interrupts or compare match B interrupts.
A/D conversion start trigger of the A/D converter	Compare match A of TMR0 and TMR2
Capable of generating baud rate clock for SCI	Generates baud rate clock for SCI.*1
Low power consumption function	Each unit can be placed in a module stop state

Note 1. For details, see section 40, Serial Communications Interface (SCIg, SCIh).

Table 29.2 TMR Functions

Item		Unit 0			Unit 1		
Counter mode		8 Bits		16 Bits	8 Bits		16 Bits
Channel		TMR0	TMR1	TMR0 + TMR1	TMR2	TMR3	TMR2 + TMR3
Count clock		PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI0	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI1	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI1	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI2	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI3	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI3
Counter clear		TMR0.TCORA TMR0.TCORB TMRI0	TMR1.TCORA TMR1.TCORB TMRI1	TMR0.TCORA + TMR1.TCORA TMR0.TCORB + TMR1.TCORB TMRI0	TMR2.TCORA TMR2.TCORB TMRI2	TMR3.TCORA TMR3.TCORB TMRI3	TMR2.TCORA + TMR3.TCORA TMR2.TCORB + TMR3.TCORB TMRI2
Compare match	Compare match A	○	○	○	○	○	○
	Compare match B	○	○	○	○	○	○
Timer output	Low output	○	○	○	○	○	○
	High output	○	○	○	○	○	○
	Toggle output	○	○	○	○	○	○
DTC activation	Compare match A	○	○	○	○	○	○
	Compare match B	○	○	○	○	○	○
	TCNT overflow	—	—	—	—	—	—
Interrupt	Compare match A	CMIA0	CMIA1	CMIA0	CMIA2	CMIA3	CMIA2
	Compare match B	CMIB0	CMIB1	CMIB0	CMIB2	CMIB3	CMIB2
	TCNT overflow	OVI0	OVI1	OVI0	OVI2	OVI3	OVI2
Cascaded connection		TMR1 overflow	TMR0 compare match A	—	TMR3 overflow	TMR2 compare match A	—
A/D conversion start trigger of the A/D converter*1		○	—	○	○	—	○
SCI baud rate clock generation*2		○		—	○		—
ELC output event	Compare match A	○	○	○	○	○	○
	Compare match B	○	○	○	○	○	○
	TCNT overflow	○	○	○	○	○	○
ELC input event	Counting start	○	○	—	○	○	—
	Event counting	○	○	—	○	○	—
	Counting restart	○	○	—	○	○	—
Module stop setting*3		MSTPCRA.MSTPA5 bit (unit 0), MSTPCRA.MSTPA4 bit (unit 1)					

○: Possible

—: Impossible

Note 1. For details, see section 57, 12-Bit A/D Converter (S12ADC).
Note 2. For details, see section 40, Serial Communications Interface (SCIg, SCIh).
Note 3. For details, see section 11, Low Power Consumption.

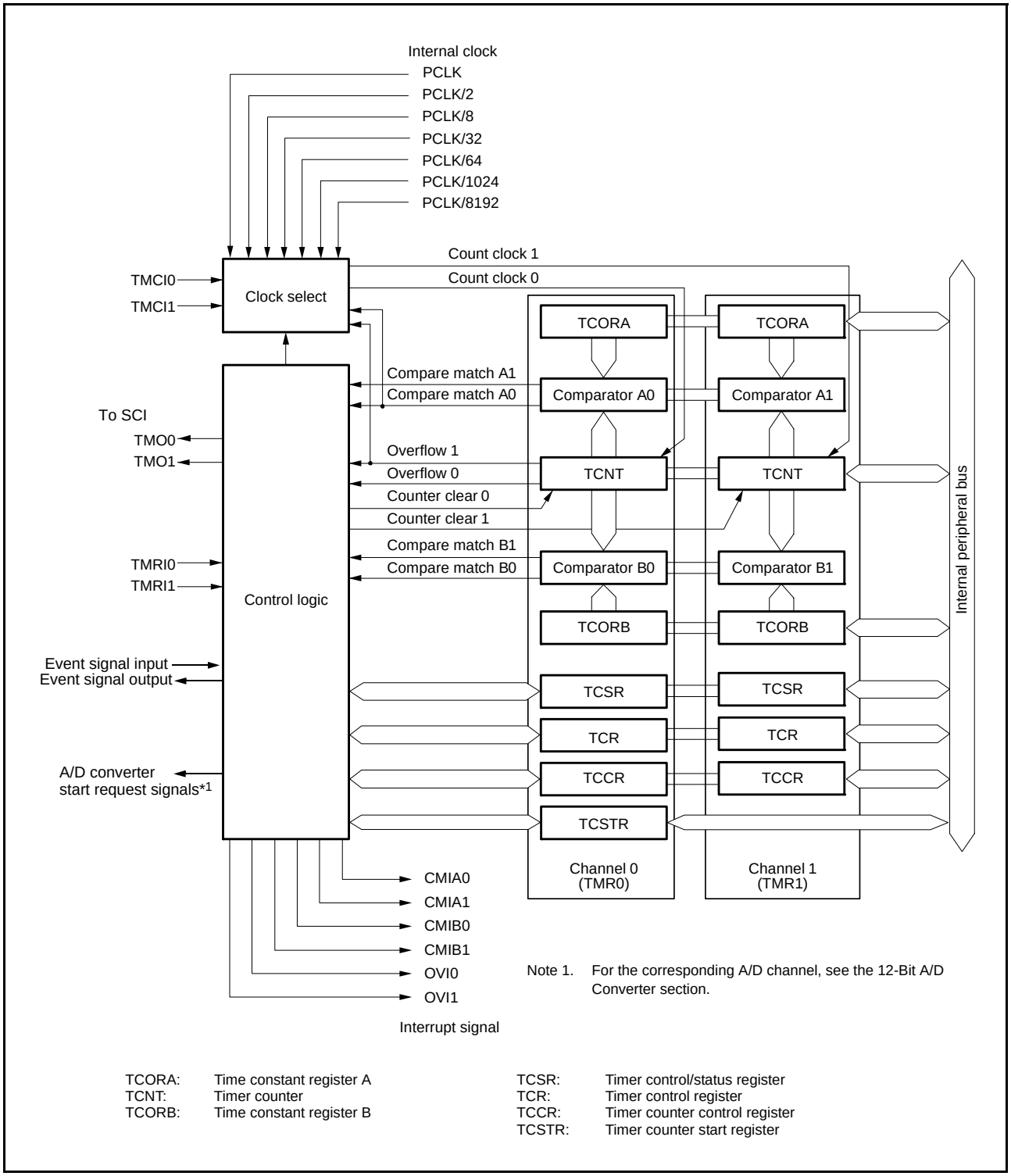


Figure 29.1 Block Diagram of TMR (Unit 0)

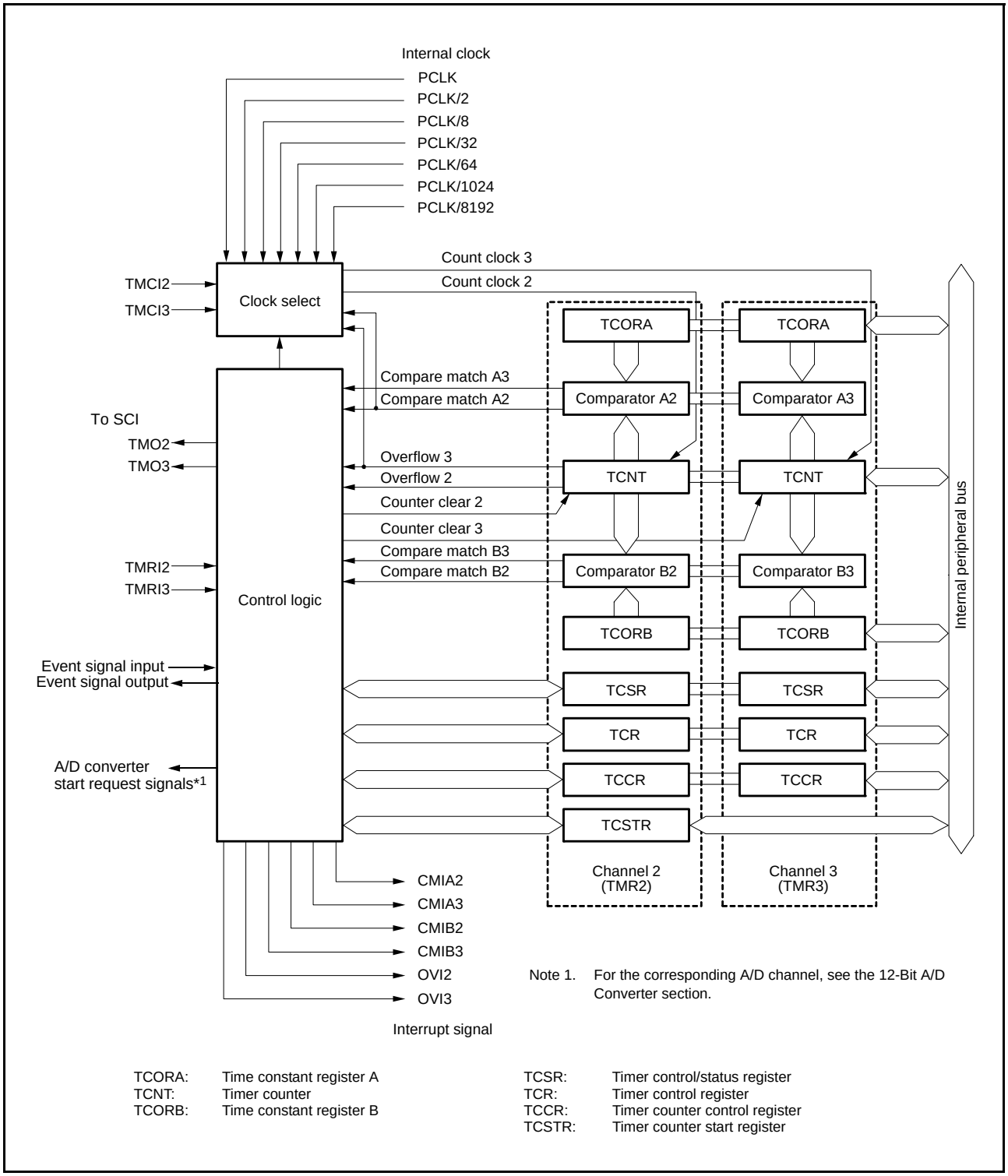


Figure 29.2 Block Diagram of TMR (Unit 1)

Table 29.3 lists the I/O pins of the TMR.

Table 29.3 Pin Configuration of TMR

Unit	Channel	Pin Name	I/O	Description
0	TMR0	TMO0	Output	Outputs compare match
		TMCI0	Input	Inputs external count clock
		TMRI0	Input	Inputs external counter reset
	TMR1	TMO1	Output	Outputs compare match
		TMCI1	Input	Inputs external count clock
		TMRI1	Input	Inputs external counter reset
1	TMR2	TMO2	Output	Outputs compare match
		TMCI2	Input	Inputs external count clock
		TMRI2	Input	Inputs external counter reset
	TMR3	TMO3	Output	Outputs compare match
		TMCI3	Input	Inputs external count clock
		TMRI3	Input	Inputs external counter reset

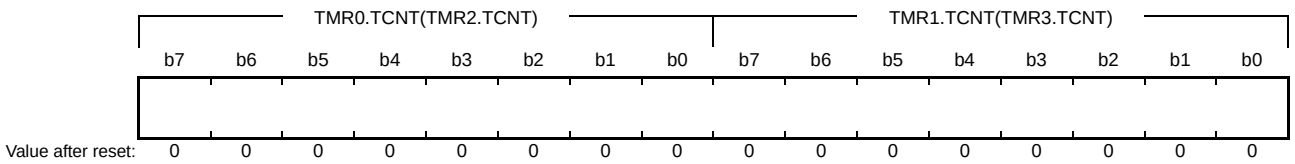
29.2 Register Descriptions

Table 29.4 Register Allocation for 16-Bit Access

Address	Upper 8 Bits	Lower 8 Bits
0008 8208h	TMR0.TCNT	TMR1.TCNT
0008 8204h	TMR0.TCORA	TMR1.TCORA
0008 8206h	TMR0.TCORB	TMR1.TCORB
0008 820Ah	TMR0.TCCR	TMR1.TCCR
0008 8218h	TMR2.TCNT	TMR3.TCNT
0008 8214h	TMR2.TCORA	TMR3.TCORA
0008 8216h	TMR2.TCORB	TMR3.TCORB
0008 821Ah	TMR2.TCCR	TMR3.TCCR

29.2.1 Timer Counter (TCNT)

Address(es): TMR0.TCNT 0008 8208h, TMR1.TCNT 0008 8209h, TMR2.TCNT 0008 8218h, TMR3.TCNT 0008 8219h



TCNT is an 8-bit readable/writable up-counter.
TMR0.TCNT and TMR1.TCNT (TMR2.TCNT and TMR3.TCNT) comprise a single 16-bit counter so they can be accessed together by a word transfer instruction.
The TCCR.CSS[1:0] and CKS[2:0] bits are used to select a count clock.
TCNT can be cleared by an external counter reset signal, compare match A, or compare match B. Which compare match to be used for clearing is selected by the TCR.CCLR[1:0] bits.
When TCNT overflows (its value changes from FFh to 00h), an overflow interrupt (low-level pulse) is output provided the interrupt request is enabled by the TCR.OVIE bit.
For details on the corresponding interrupt vector number, see section 15, Interrupt Controller (ICUA), and Table 29.6, TMR Interrupt Sources.

29.2.2 Time Constant Register A (TCORA)

Address(es): TMR0.TCORA 0008 8204h, TMR1.TCORA 0008 8205h, TMR2.TCORA 0008 8214h, TMR3.TCORA 0008 8215h

	TMR0.TCORA(TMR2.TCORA)								TMR1.TCORA(TMR3.TCORA)							
	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TCORA is an 8-bit readable/writable register.
TMR0.TCORA and TMR1.TCORA (TMR2.TCORA and TMR3.TCORA) comprise a single 16-bit register so they can be accessed together by a word transfer instruction.
The value in TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding compare match A is generated, and a compare match A interrupt (low-level pulse) is output provided the interrupt request is enabled by the TCR.CMIEA bit.
However, comparison is not performed during writing to TCORA. The timer output from the TMOn pin can be freely controlled by this compare match A and the settings of the TCSR.OSA[1:0] bits.

29.2.3 Time Constant Register B (TCORB)

Address(es): TMR0.TCORB 0008 8206h, TMR1.TCORB 0008 8207h, TMR2.TCORB 0008 8216h, TMR3.TCORB 0008 8217h

	TMR0.TCORB(TMR2.TCORB)								TMR1.TCORB(TMR3.TCORB)							
	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TCORB is an 8-bit readable/writable register.
TMR0.TCORB and TMR1.TCORB (TMR2.TCORB and TMR3.TCORB) comprise a single 16-bit register so they can be accessed together by a word transfer instruction.
The value in TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding compare match B is generated, and a compare match B interrupt (low-level pulse) is output provided the interrupt request is enabled by the TCR.CMIEB bit.
However, comparison is not performed during writing to TCORB. The timer output from the TMOn pin can be freely controlled by this compare match B and the settings of the TCSR.OSB[1:0] bits.

29.2.4 Timer Control Register (TCR)

Address(es): TMR0.TCR 0008 8200h, TMR1.TCR 0008 8201h, TMR2.TCR 0008 8210h, TMR3.TCR 0008 8211h

	b7	b6	b5	b4	b3	b2	b1	b0
	CMIEB	CMIEA	OVIE	CCLR[1:0]		—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4, b3	CCLR[1:0]	Counter Clear*1	b4 b3 0 0: Clearing is disabled 0 1: Cleared by compare match A 1 0: Cleared by compare match B 1 1: Cleared by the external counter reset signal (Select edge or level by the TMRIS bit in TCCR.)	R/W
b5	OVIE	Timer Overflow Interrupt Enable	0: Overflow interrupt requests (OVIn) are disabled 1: Overflow interrupt requests (OVIn) are enabled	R/W
b6	CMIEA	Compare Match Interrupt Enable A	0: Compare match A interrupt requests (CMIAn) are disabled 1: Compare match A interrupt requests (CMIAn) are enabled	R/W
b7	CMIEB	Compare Match Interrupt Enable B	0: Compare match B interrupt requests (CMIBn) are disabled 1: Compare match B interrupt requests (CMIBn) are enabled	R/W

Note 1. To use an external counter reset signal, set the corresponding pin function. For details, refer to section 22, I/O Ports and section 23, Multi-Function Pin Controller (MPC).

CCLR[1:0] Bits (Counter Clear)

Select the condition by which TCNT is cleared.

OVIE Bit (Timer Overflow Interrupt Enable)

Selects whether overflow interrupt requests (OVIn) issued by TCNT are enabled or disabled.

CMIEA Bit (Compare Match Interrupt Enable A)

Selects whether compare match A interrupt requests (CMIAn) that are issued when the value of TCORA corresponds to that of TCNT are enabled or disabled.

CMIEB Bit (Compare Match Interrupt Enable B)

Selects whether compare match B interrupt requests (CMIBn) that are issued when the value of TCORB corresponds to that of TCNT are enabled or disabled.

29.2.5 Timer Counter Control Register (TCCR)

Address(es): TMR0.TCCR 0008 820Ah, TMR1.TCCR 0008 820Bh, TMR2.TCCR 0008 821Ah, TMR3.TCCR 0008 821Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	TMRIS	—	—	CSS[1:0]		CKS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CKS[2:0]	Clock Select*1	See Table 29.5.	R/W
b4, b3	CSS[1:0]	Clock Source Select	See Table 29.5.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TMRIS	Timer Reset Detection Condition Select	0: Cleared at rising edge of the external counter reset signal 1: Cleared when the external counter reset signal is high	R/W

Note 1. To use an external count clock, set the corresponding pin function. For details, refer to section 22, I/O Ports and section 23, Multi-Function Pin Controller (MPC).

CKS[2:0] Bits (Clock Select)
CSS[1:0] Bits (Clock Source Select)

The CKS[2:0] and CSS[1:0] bits select a clock. For details, see Table 29.5.

TMRIS Bit (Timer Reset Detection Condition Select)

This bit is enabled when the TCR.CCLR[1:0] bits are 11b (cleared by external counter reset signal) and selects the condition for detecting counter reset (level or edge).

Table 29.5 Clock Input to TCNT and Count Condition

Channel	TCCR Register CSS[1:0] b4	b3	CKS[2:0] b2	b1	b0	Description
TMR0 (TMR2)	0	0	—	0	0	Clock input prohibited
					1	Uses external count clock. Counts at rising edge*1.
				1	0	Uses external count clock. Counts at falling edge*1.
					1	Uses external count clock. Counts at both rising and falling edges*1.
	0	1	0	0	0	Uses internal clock. Counts at PCLK.
					1	Uses internal clock. Counts at PCLK/2.
				1	0	Uses internal clock. Counts at PCLK/8.
					1	Uses internal clock. Counts at PCLK/32.
			1	0	0	Uses internal clock. Counts at PCLK/64.
					1	Uses internal clock. Counts at PCLK/1024.
				1	0	Uses internal clock. Counts at PCLK/8192.
					1	Clock input prohibited
	1	0	—	—	—	Setting prohibited
	1	1	—	—	—	Counts at TMR1.TCNT (TMR3.TCNT) overflow signal*2.
TMR1 (TMR3)	0	0	—	0	0	Clock input prohibited
					1	Uses external count clock. Counts at rising edge*1.
				1	0	Uses external count clock. Counts at falling edge*1.
					1	Uses external count clock. Counts at both rising and falling edges*1.
	0	1	0	0	0	Uses internal clock. Counts at PCLK.
					1	Uses internal clock. Counts at PCLK/2.
				1	0	Uses internal clock. Counts at PCLK/8.
					1	Uses internal clock. Counts at PCLK/32.
			1	0	0	Uses internal clock. Counts at PCLK/64.
					1	Uses internal clock. Counts at PCLK/1024.
				1	0	Uses internal clock. Counts at PCLK/8192.
					1	Clock input prohibited
	1	0	—	—	—	Setting prohibited
	1	1	—	—	—	Counts at TMR0.TCNT (TMR2.TCNT) compare match A*2.

Note 1. To use an external count clock, set the corresponding pin function. For details, see section 22, I/O Ports and section 23, Multi-Function Pin Controller (MPC).

Note 2. If the clock input of TMR0 (TMR2) is the overflow signal of the TMR1.TCNT (TMR3.TCNT) counter and that of TMR1 (TMR3) is the compare match signal of the TMR0.TCNT (TMR2.TCNT) counter, no TCNT count clock is generated. Do not use this setting.

29.2.6 Timer Control/Status Register (TCSR)

- TMR0.TCSR, TMR2.TCSR

Address(es): TMR0.TCSR 0008 8202h, TMR2.TCSR 0008 8212h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ADTE	OSB[1:0]		OSA[1:0]	
Value after reset:	x	x	x	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OSA[1:0]	Output Select A*1	b1 b0 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B *1	b3 b2 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b4	ADTE	A/D Trigger Enable*2	0: A/D conversion start request in response to compare match A is disabled. 1: A/D conversion start request in response to compare match A is enabled.	R/W
b7 to b5	—	Reserved	These bits are read as an undefined value. The write value should be 1.	R/W

Note 1. When the OSA[1:0] and OSB[1:0] bits are all 0, the output enable signal corresponding to the TMOn pin is negated and a request for high-impedance output is issued to the I/O port. Timer output pin is low until the first compare match occurs after a reset when either of the OSA[1:0] or OSB[1:0] bits are 1.

Note 2. For the corresponding A/D channel, see section 57, 12-Bit A/D Converter (S12ADC).

OSA[1:0] Bits (Output Select A)

These bits select a method of TMOn pin output when compare match A of TCORA and TCNT occurs.

OSB[1:0] Bits (Output Select B)

These bits select a method of TMOn pin output when compare match B of TCORB and TCNT occurs.

- TMR1.TCSR, TMR3.TCSR

Address(es): TMR1.TCSR 0008 8203h, TMR3.TCSR 0008 8213h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	OSB[1:0]		OSA[1:0]	
Value after reset:	x	x	x	1	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OSA[1:0]	Output Select A*1	b1 b0 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B *1	b3 b2 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b4	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7 to b5	—	Reserved	These bits are read as an undefined value. The write value should be 1.	R/W

Note 1. When the OSA[1:0] and OSB[1:0] bits are all 0, the output enable signal corresponding to the TMOn pin is negated and a request for high-impedance output is issued to the I/O port. Timer output pin is low until the first compare match occurs after a reset when either of the OSA[1:0] or OSB[1:0] bits are 1.

OSA[1:0] Bits (Output Select A)

These bits select a method of TMOn pin output when compare match A of TCORA and TCNT occurs.

OSB[1:0] Bits (Output Select B)

These bits select a method of TMOn pin output when compare match B of TCORB and TCNT occurs.

29.2.7 Timer Counter Start Register (TCSTR)

Address(es): TMR0.TCSTR 0008 820Ch, TMR1.TCSTR 0008 820Dh, TMR2.TCSTR 0008 821Ch, TMR3.TCSTR 0008 821Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TCS
Value after reset:	x	x	x	x	x	x	x	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	TCS	Timer Counter Status	0: Count stopped state in response to ELC. 1: Count start state in response to ELC.	R/W
b7 to b1	—	Reserved	These bits are read as an undefined value. The write value should be 0.	R/W

TCS Bit (Timer Counter Status)

The TCS bit is used to check the state of the timer count in response to ELC.

When this bit is read as 1, it shows the timer start state in response to ELC. When this bit is read as 0, it shows the timer stopped state in response to ELC.

This bit is cleared by writing 0. Do not write 1 to this bit.

The TCS bit is valid only when the count start operation is selected by the ELOPD register of the event controller (ELC). For details, see section 29.7, Link Operation by ELC, or section 21, Event Link Controller (ELC).

29.3 Operation

29.3.1 Pulse Output

Figure 29.3 shows an example of the 8-bit timer being used to generate a pulse output with a desired duty cycle.

1. Set the TCR.CCLR[1:0] bits to 01b (cleared by compare match A) so that TCNT is cleared at a compare match of TCORA.
2. Set the TCSR.OSA[1:0] bits to 10b (high output) and TCSR.OSB[1:0] bits to 01b (low output), causing the output to change to high at a compare match of TCORA and to low at a compare match of TCORB.

With these settings, the 8-bit timer provides pulses output at a cycle determined by TCORA with a pulse width determined by TCORB. No software intervention is required.
The timer output pin is low after the TCSR.OSA[1:0] or TCSR.OSB[1:0] bits are set until the first compare match occurs after a reset.

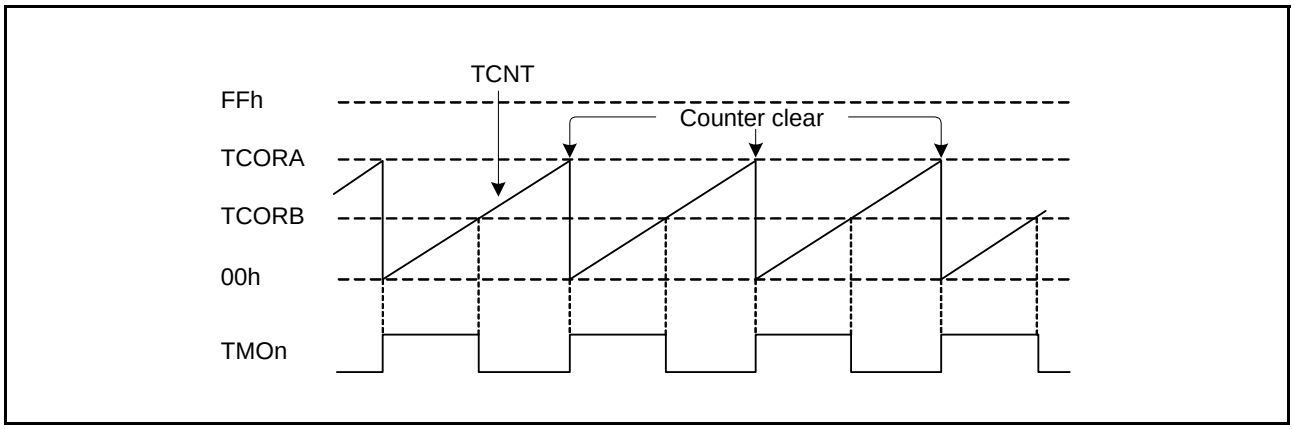


Figure 29.3 Example of Pulse Output (n = 0 to 3)

29.3.2 External Counter Reset Input

Figure 29.4 shows an example of the 8-bit timer being used to generate a pulse which is output after a desired delay time from a TMRIn input.

1. Set the TCR.CCLR[1:0] bits to 11b (cleared by external counter reset signal) and set the TMRIS bit in TCCR to 1 (cleared when the external counter reset signal is high) so that TCNT is cleared at the high level input of the TMRIn signal.
2. Set the TCSR.OSA[1:0] bits to 10b (high output) and the TCSR.OSB[1:0] bits to 01b (low output), causing the output to change to high at a compare match of TCORA and to low at a compare match of TCORB.

With these settings, the 8-bit timer provides pulses output at a desired delay time from a TMRIn input determined by TCORA and with a pulse width determined by TCORB and TCORA.

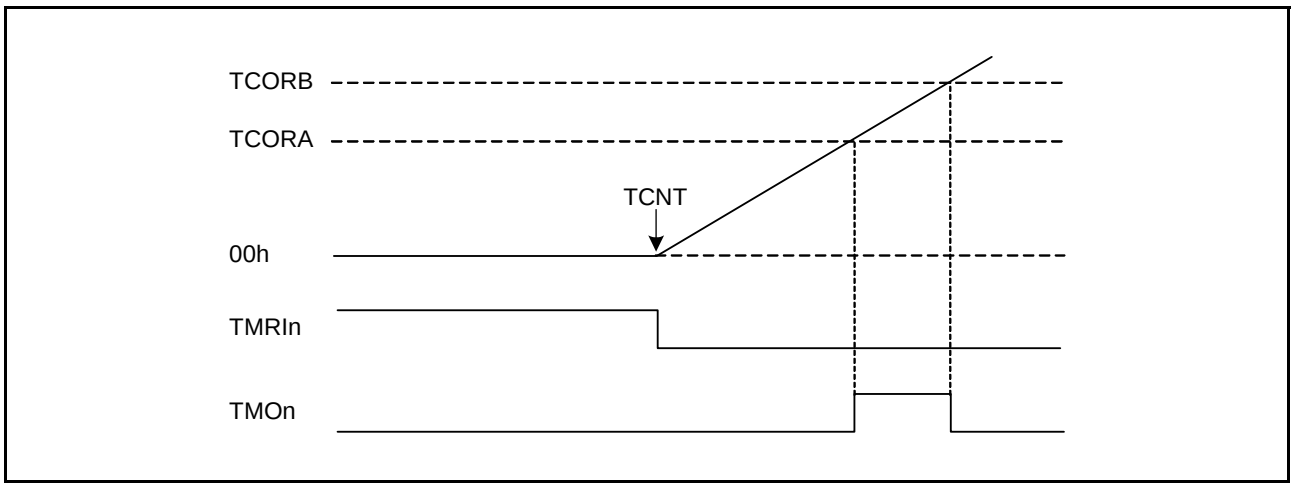


Figure 29.4 Example of External Counter Reset Signal Input (n = 0 to 3)

29.4 Operation Timing

29.4.1 TCNT Count Timing

Figure 29.5 shows the count timing of TCNT for internal clock. Figure 29.6 shows the count timing of TCNT for external clock.

Note that the external clock pulse width must be at least 1.5 PCLK cycles for increment at a single edge, and at least 2.5 PCLK cycles for increment at both edges. The counter will not increment correctly if the pulse width is less than these values.

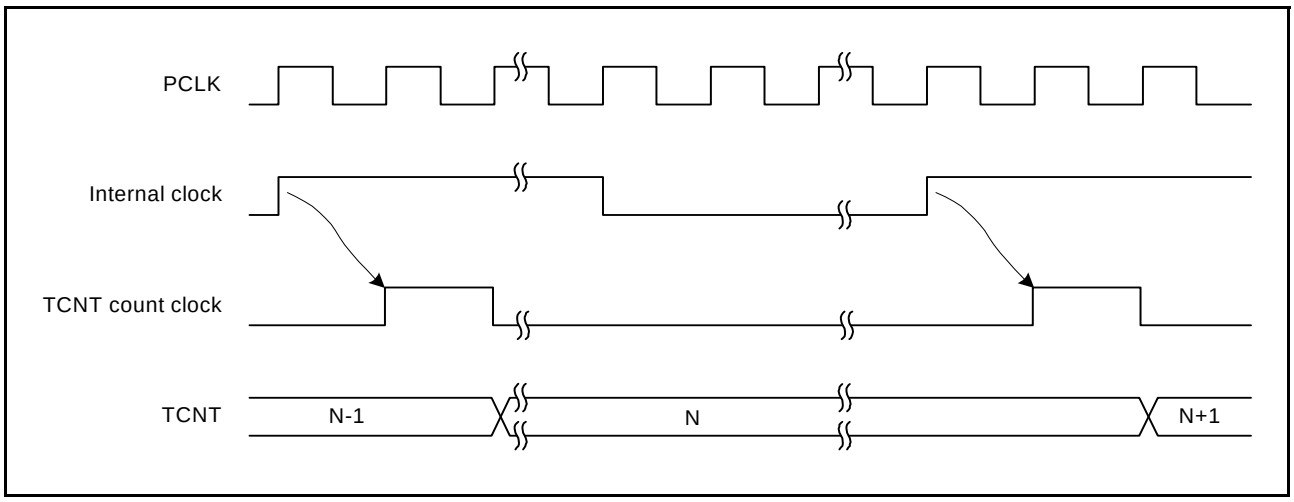


Figure 29.5 Count Timing for Internal Clock

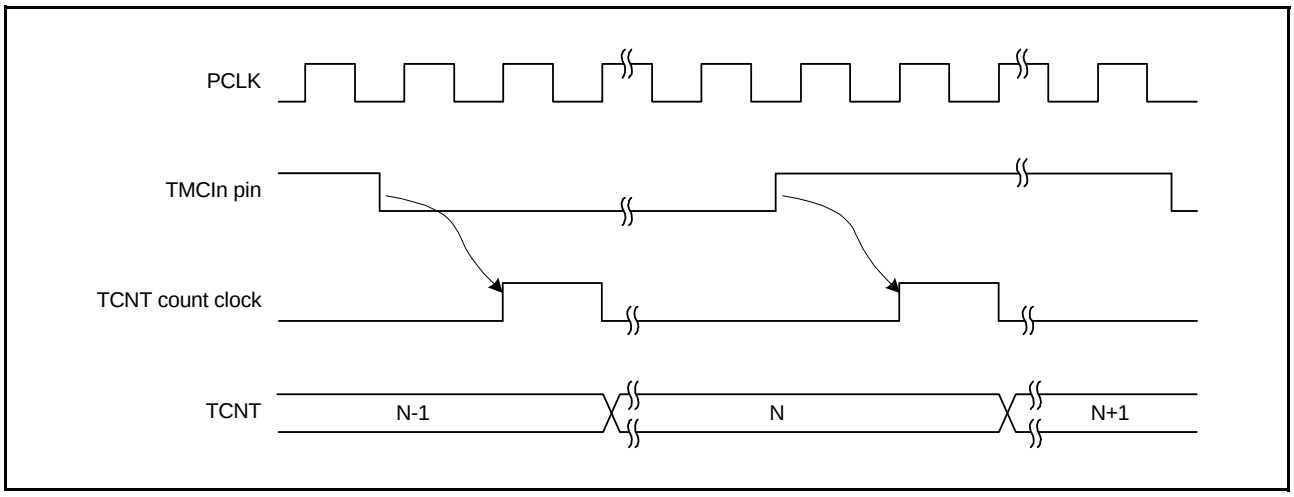


Figure 29.6 Count Timing for External Clock (at Both Edges)

29.4.2 Timing of Interrupt Signal Output on a Compare Match

A compare match refers to a match between the value of the TCORA or TCORB register and the TCNT, and a compare match interrupt signal is output at this time if the interrupt request is enabled. The compare match is generated in the last cycle in which the values match (at the time at which the value counted by TCNT to produce the match is updated). Accordingly, after a match between TCNT and the TCORA or TCORB register is detected, the compare match is not actually generated until the next cycle of the TCNT count clock. Figure 29.7 shows the timing of output of the interrupt signal.
For the corresponding interrupt vector number, see section 15, Interrupt Controller (ICUA) and Table 29.6.

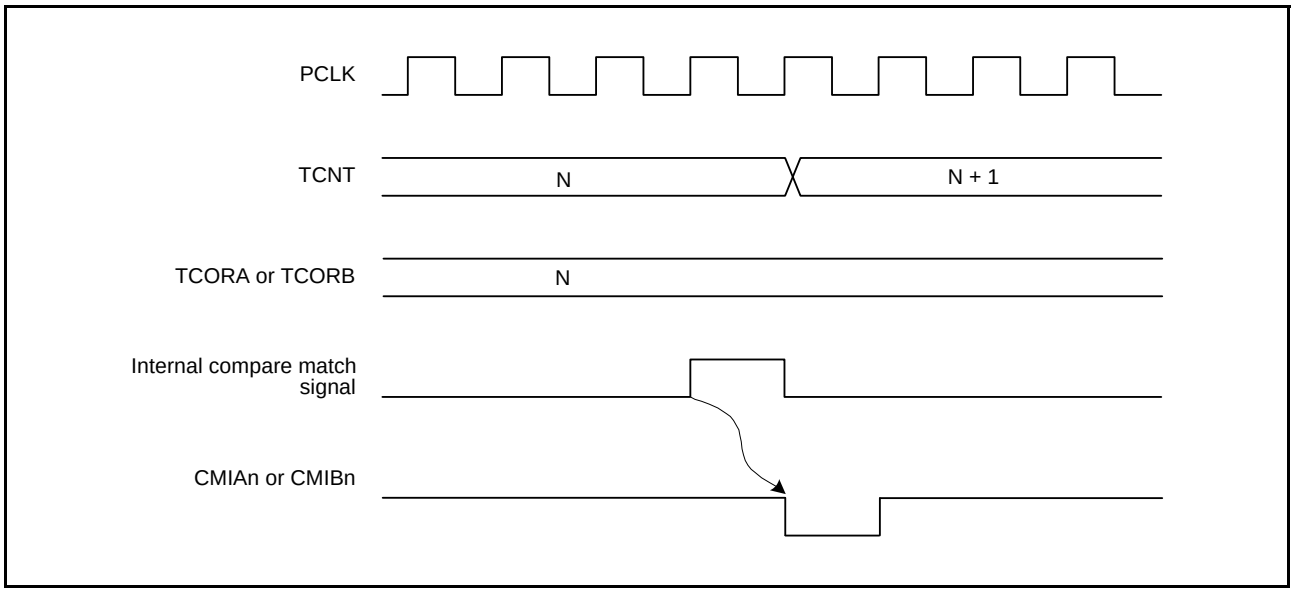


Figure 29.7 Timing of Interrupt Flag Setting to 1 at Compare Match (n = 0 to 3)

29.4.3 Timing of Timer Output Signal at Compare Match

When a compare match signal is generated, the output value specified by the TCSR.OSA[1:0] and OSB[1:0] bits is output on the timer output pin (TMOn).
Figure 29.8 shows the timing when the timer output is toggled by the compare match A signal.

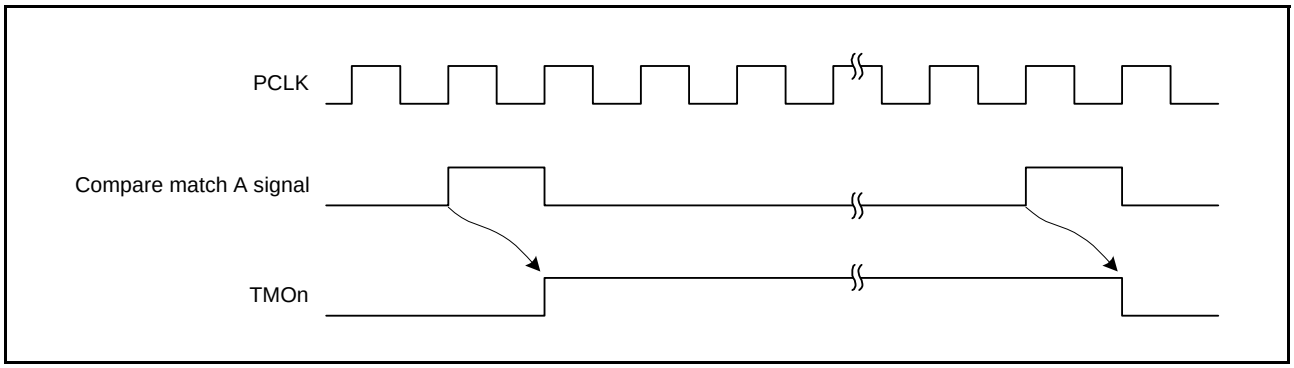


Figure 29.8 Timing of Timer Output Signal at Compare Match A Signal (n = 0 to 3)

29.4.4 Timing of Counter Clear by Compare Match

TCNT is cleared when compare match A or B occurs, depending on the settings of the TCR.CCLR[1:0] bits.
Figure 29.9 shows the timing of this operation.

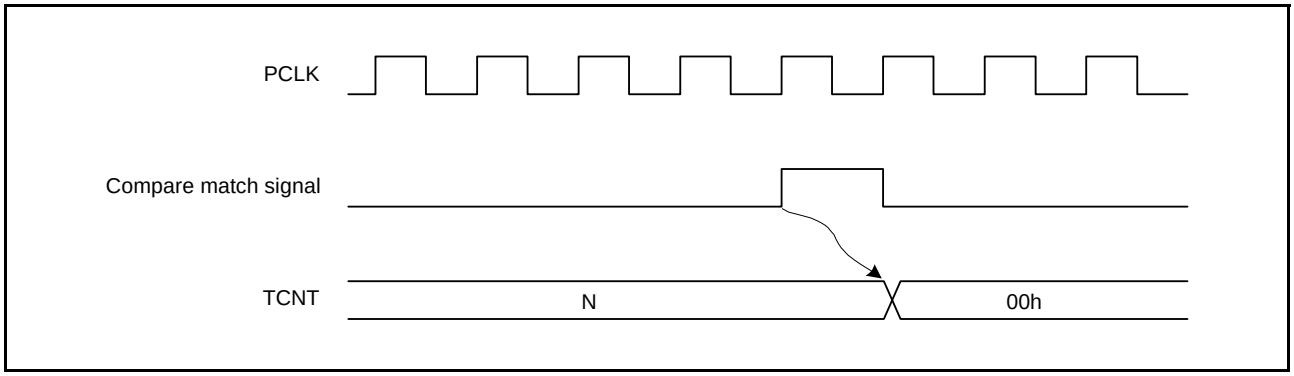


Figure 29.9 Timing of Counter Clear by Compare Match

29.4.5 Timing of the External Reset for TCNT

TCNT is cleared at the rising edge or high level of an external counter reset signal, depending on the settings of the TCR.CCLR[1:0] bits. At least 2 PCLK cycles are required from a reset input to clearing of TCNT.
Figure 29.10 and Figure 29.11 show the timing of this operation.

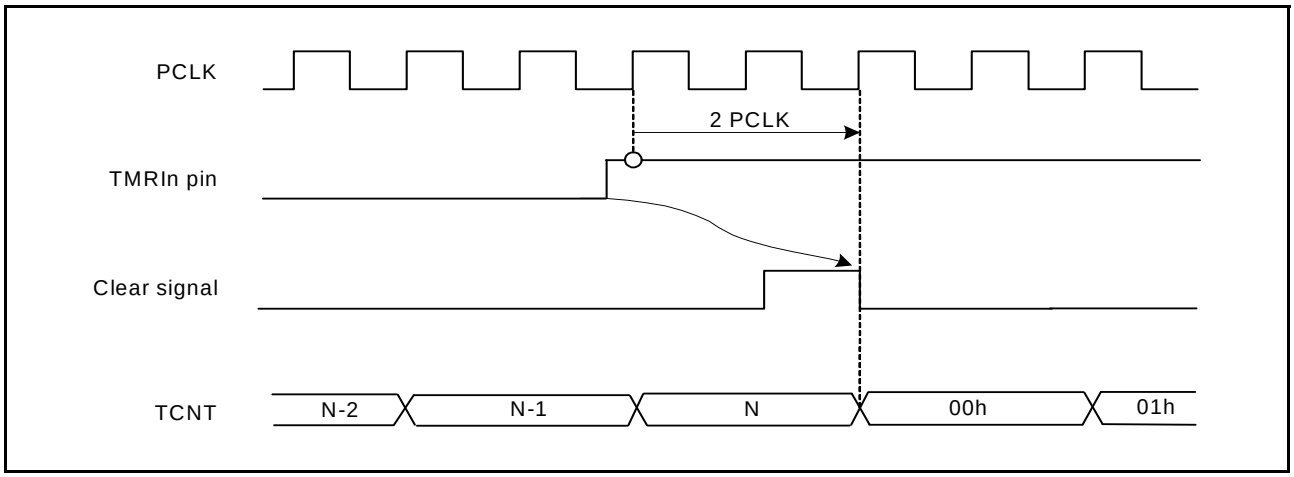


Figure 29.10 Clear Timing by External Counter Reset Signal (Rising Edge)

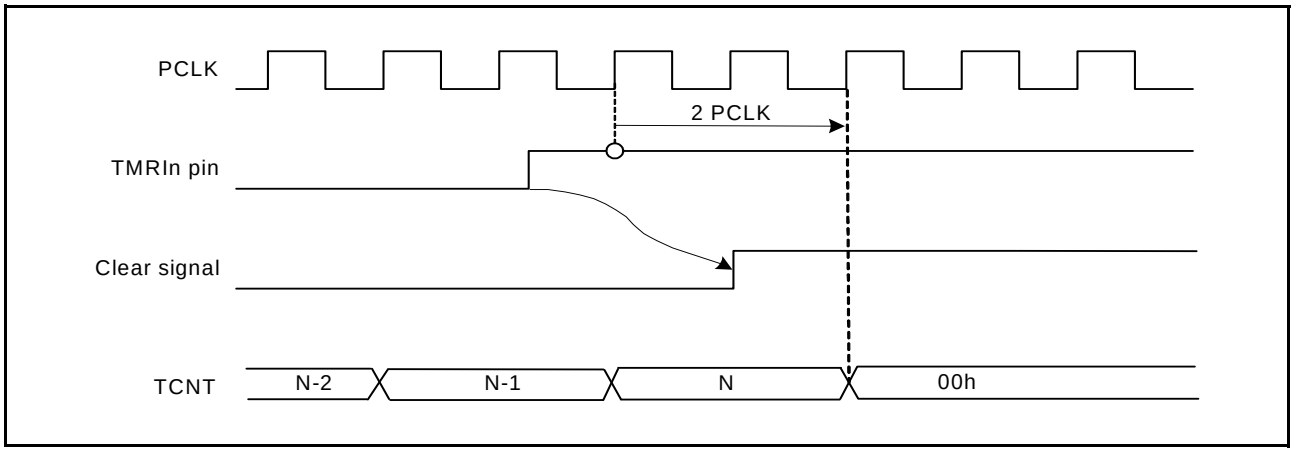


Figure 29.11 Clear Timing by External Counter Reset Signal (High Level)

29.4.6 Timing of Interrupt Signal Output on an Overflow

When TCNT overflows (changes from FFh to 00h), an overflow interrupt signal is output if this interrupt request is enabled.

Figure 29.12 shows the timing of output of the interrupt signal.

For the corresponding interrupt vector number, see section 15, Interrupt Controller (ICUA) and Table 29.6.

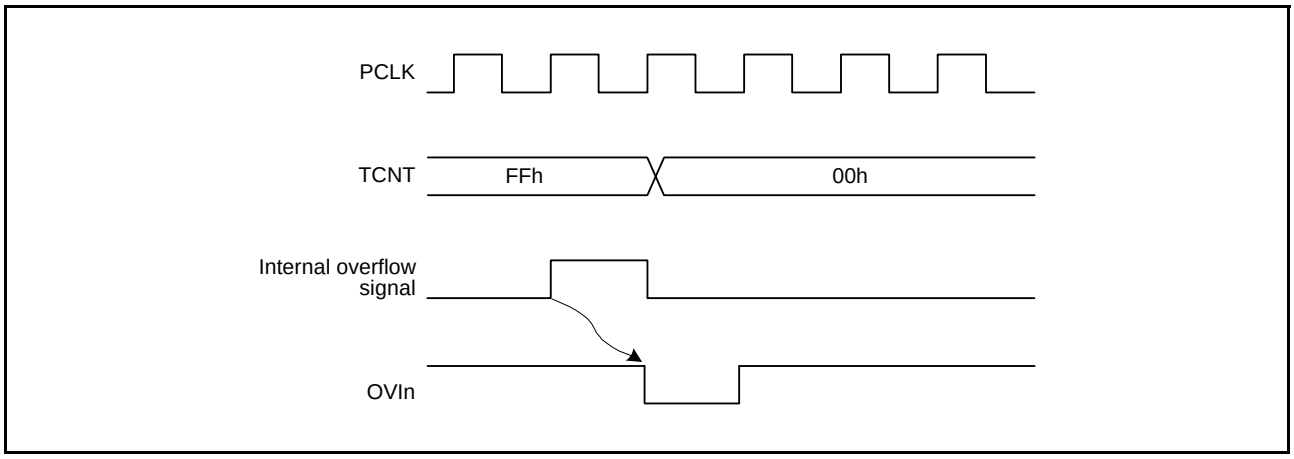


Figure 29.12 Timing of Overflow Interrupt Flag Setting to 1 (n = 0 to 3)

29.5 Operation with Cascaded Connection

If the CSS[1:0] bits in either TMR0.TCCR or TMR1.TCCR are set to 11b, the TMR of the two channels are cascaded. With this configuration, a single 16-bit timer could be used (16-bit counter mode) or compare matches of TMR0 could be counted by TMR1 (compare match count mode).

Supplementary information: This section describes unit 0. The operation of unit 1 with cascaded connection is the same as unit 0.

29.5.1 16-Bit Count Mode

When the TMR0.TCCR.CSS[1:0] bits are set to 11b, the timer functions as a single 16-bit timer with TMR0 occupying the upper 8 bits and TMR1 occupying the lower 8 bits.

(1) Counter Clear Specification

- The settings of the TMR0.TCR.CCLR[1:0] bits become effective for the 16-bit counter. If the TMR0.TCR.CCLR[1:0] bits have been set for counter clear at compare match, the 16-bit counter (TMR0.TCNT and TMR1.TCNT together) is cleared when a 16-bit compare match event occurs. The 16-bit counter (TMR0.TCNT and TMR1.TCNT together) is cleared even if counter clear by the TMRI0 pin has been set.
- The settings of the TMR1.TCR.CCLR[1:0] bits are ignored.

(2) Pin Output

- Control of output from the TMO0 pin by the TMR0.TCSR.OSA[1:0] and OSB[1:0] bits is in accordance with the 16-bit compare match conditions.
- Control of output from the TMO1 pin by the TMR1.TCSR.OSA[1:0] and OSB[1:0] bits is in accordance with the lower 8-bit compare match conditions.

29.5.2 Compare Match Count Mode

When the TMR1.TCCR.CSS[1:0] bits are set to 11b, TMR1.TCNT counts the number of occurrences of compare match A for TMR0. TMR0 and TMR1 are controlled independently. Conditions such as generation of interrupts, output from the TMOn (n = 0, 1) pin, and counter clear are in accordance with the settings for each channel.

29.6 Interrupt Sources

29.6.1 Interrupt Sources and DTC Activation

There are three interrupt sources for TMRn: CMIAn, CMIBn, and OVIn. Their interrupt sources and priorities are listed in Table 29.6.
It is also possible to activate the DTC by means of CMIAn and CMIBn interrupts.

Table 29.6 TMR Interrupt Sources

Name	Interrupt Sources	DTC Activation
CMIA0	TMR0.TCORA compare match	Possible
CMIB0	TMR0.TCORB compare match	Possible
OVI0	TMR0.TCNT overflow	Not possible
CMIA1	TMR1.TCORA compare match	Possible
CMIB1	TMR1.TCORB compare match	Possible
OVI1	TMR1.TCNT overflow	Not possible
CMIA2	TMR2.TCORA compare match	Possible
CMIB2	TMR2.TCORB compare match	Possible
OVI2	TMR2.TCNT overflow	Not possible
CMIA3	TMR3.TCORA compare match	Possible
CMIB3	TMR3.TCORB compare match	Possible
OVI3	TMR3.TCNT overflow	Not possible

29.6.2 Startup of the A/D Converter

The compare match A of TMR0 and TMR2 allows the A/D converter*1 to be started.
An A/D conversion start request is issued to the A/D converter in response to a generation of compare match A when the TMRn.TCSR.ADTE bit is 1 (i.e., when an A/D conversion request in response to compare match A is enabled). In this case, the conversion trigger for the 8-bit timer should be selected in the A/D converter to start A/D conversion.

Note 1. For the corresponding unit of the A/D converter, see section 57, 12-Bit A/D Converter (S12ADC).

Table 29.7 Startup of A/D Converter

Module Symbol	Unit	Target	A/D Startup Request	A/D Conversion Start Request
S12AD	0	Between TMR0.TCORA and TMR0.TCNT	Compare match	TMTRG0AN_0
S12AD1	1	Between TMR2.TCORA and TMR2.TCNT		TMTRG0AN_1

29.7 Link Operation by ELC

29.7.1 Event Signal Output to ELC

The TMR uses the event link controller (ELC) to perform link operation to the previously specified module using the interrupt request signal as the event signal. The TMR outputs compare match A, compare match B, and overflow signals as event signals. Channels that can be used in this way are TMR0 to TMR3.
The event signal can be output regardless of the setting of the corresponding interrupt request enable bits (TMRn.TCR.OVIE, TMRn.TCR.CMIEA, and TMRn.TCR.CMIEB (n = 0 to 3)). For details, see section 21, Event Link Controller (ELC).
The event output function can be used for the cascaded operation.

29.7.2 TMR Operation when Receiving an Event Signal from ELC

The TMR can perform either of the following operations upon the event previously specified by the ELSRn register of the ELC. However, the ELC does not support the cascaded operation.

(1) Count Start

When the TMR count start operation is selected by the ELOPD register of the ELC and the event specified by ELSRn occurs, the TCSTR.TCS bit is set to 1, starting the TMR count operation. After the TMR count start operation is selected by the ELOPD register of the ELC, use the TCCR.CKS[2:0] and CSS[1:0] bits to select the count source.
If the specified event occurs while the TCS bit is 1, the event is ignored.
Write 0 to the TCSTR.TCS bit to stop counting.
When the count start event is input in the count stopped state, the TMR starts counting again according to the CKS[2:0] and CSS[1:0] bits.
The TCS bit is valid only when the ELOPD.TMR0MD[1:0] and ELOPD.TMR2MD[1:0] bits of the ELC select the count start operation.

(2) Event Count

When the TMR event count operation is selected by the ELOPD register of the ELC and the event specified by ELSRn occurs, the events are counted as the count source regardless of the TCCR.CKS[2:0] and CSS[1:0] bit settings. Reading the counter value returns the number of events that have been actually input.

(3) Count Restart

When the TMR count restart operation is selected by the ELOPD register of the ELC and the event specified by ELSRn occurs, the TCNT counter value is modified to the initial value. If the CKS[2:0] and CSS[1:0] bit settings are not disabling the clock input, the count operation is continued.

29.7.3 Notes on Operating TMR According to an Event Signal from ELC

The following describes the notes on operating the TMR using the event link feature.

(1) Count Start

When the event specified by ELSRn occurs during the write cycle to the TCSTR.TCS bit, the cycle is not completed; setting 1 according to the event occurrence takes priority.

(2) Event Count

When the event specified by ELSRn occurs during the write cycle to the TCNT, the cycle is not completed; event count operation according to the event occurrence takes priority.

(3) Count Restart

When the event specified by ELSRn occurs during the write cycle to the TCNT, the cycle is not completed; count value initialization according to the event occurrence takes priority.

29.8 Usage Notes

29.8.1 Module Stop State Setting

Operation of the TMR can be disabled or enabled by using the module stop control registers. The initial setting is for halting of TMR operation. Register access becomes possible after release from the module stop state. For details, see section 11, Low Power Consumption.

29.8.2 Notes on Setting Cycle

If the compare match is selected for counter clear, TCNT is cleared at the last PCLK in the cycle in which the value of TCNT matches with that of TCORA or TCORB. TCNT updates the counter value at this last state. Therefore, the counter frequency is obtained by the following formula (f: Counter frequency, PCLK: Operating frequency, N: TCORA and TCORB register setting value).

$$f = PCLK / (N + 1)$$

29.8.3 Conflict between TCNT Write and Counter Clear

If a counter clear signal is generated concurrently with CPU write to TCNT, the clear takes priority and the write is not performed as shown in Figure 29.13.

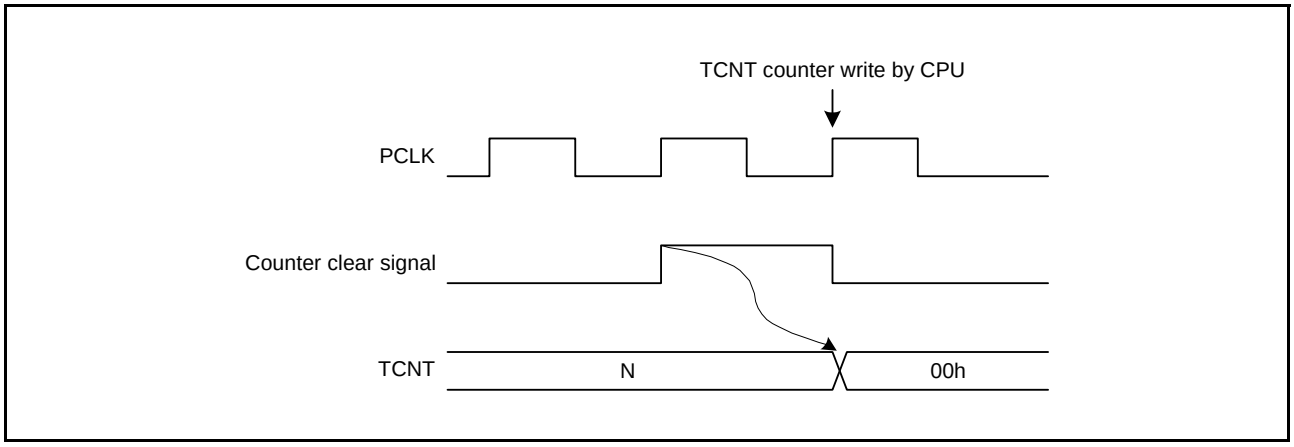


Figure 29.13 Conflict between TCNT Write and Counter Clear

29.8.4 Conflict between TCNT Write and Increment

Even if a counting-up signal is generated concurrently with CPU write to TCNT, the counting-up is not performed and the write takes priority as shown in Figure 29.14.

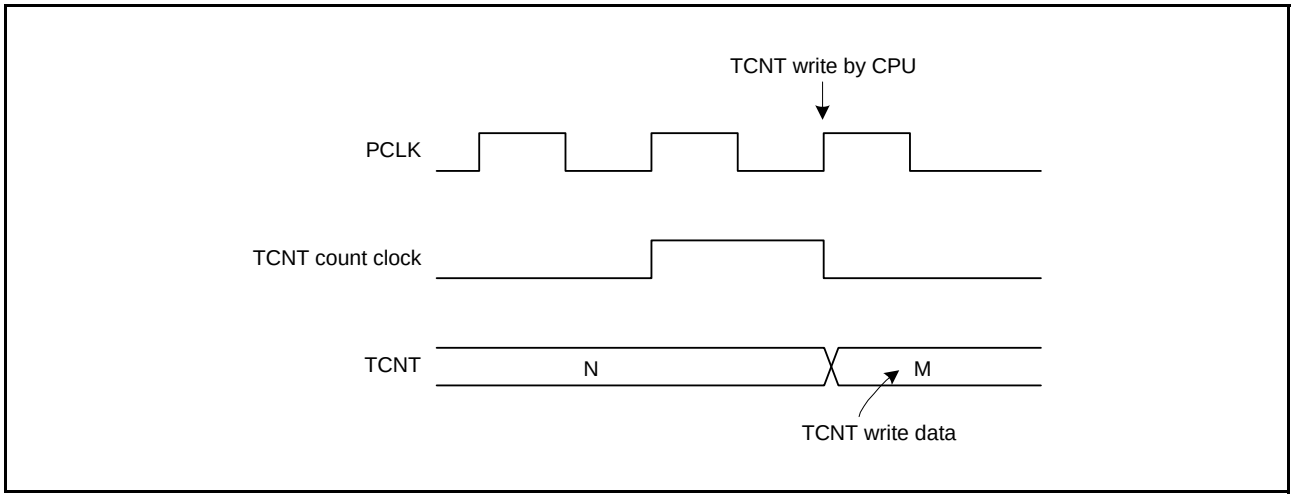


Figure 29.14 Conflict between TCNT Write and Increment

29.8.5 Conflict between TCORA or TCORB Write and Compare Match

Even if a compare match signal is generated simultaneously with CPU write to TCORA or TCORB as shown in Figure 29.15, the write takes priority and the compare match signal does not reach High level.

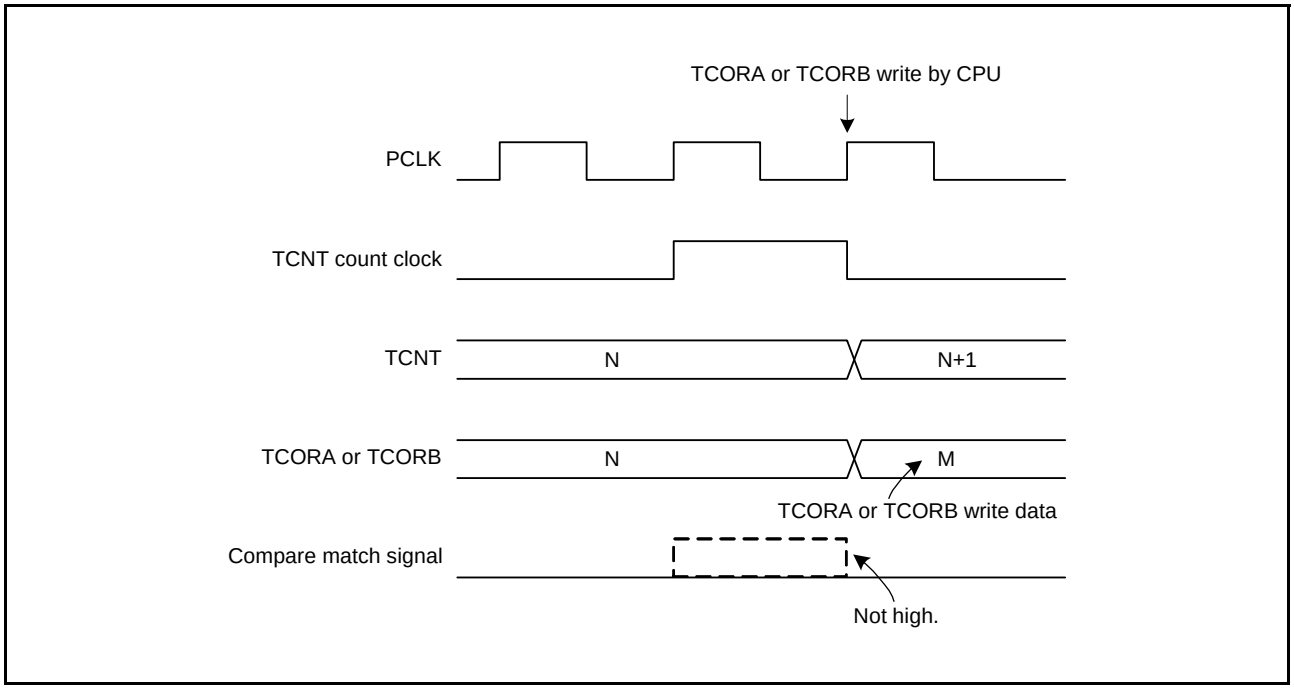


Figure 29.15 Conflict between TCORA or TCORB Write and Compare Match

29.8.6 Conflict between Compare Matches A and B

If compare match events A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output methods high for compare match A and compare match B, as listed in Table 29.8.

Table 29.8 Timer Output Priorities

Output Setting	Priority
Toggle output	High
High output	↑
Low output	
No change	Low

29.8.7 Switching of Internal Clocks and TCNT Operation

TCNT may be incremented erroneously depending on when the internal clock is switched. Table 29.9 lists the relationship between the timing at which the internal clock is switched (by writing to the TCCR.CKS[2:0] bits) and the operation of TCNT.

When TCNT count clock is generated from an internal clock, the rising edge of the internal clock pulse are always monitored. If the signal levels of the clocks before and after switching change from low to high as shown in No. 2 in Table 29.9, the change is considered as an edge. Therefore, a TCNT count clock is generated and TCNT is incremented. The erroneous increment of TCNT can also happen when switching between internal and internal clocks.

Table 29.9 Switching of Internal Clocks and TCNT Operation (1/2)

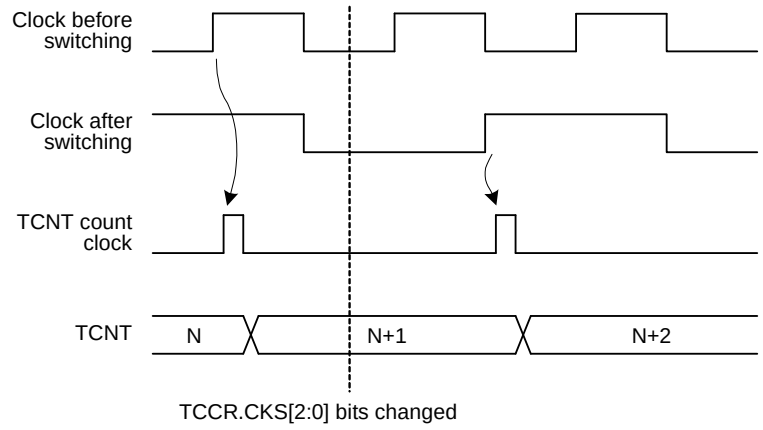
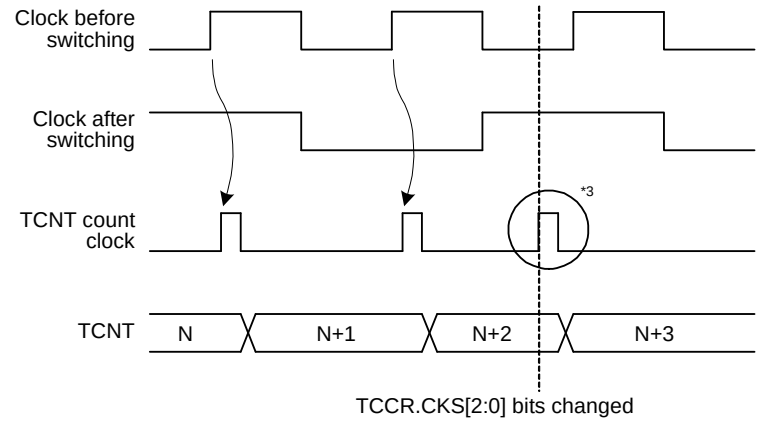
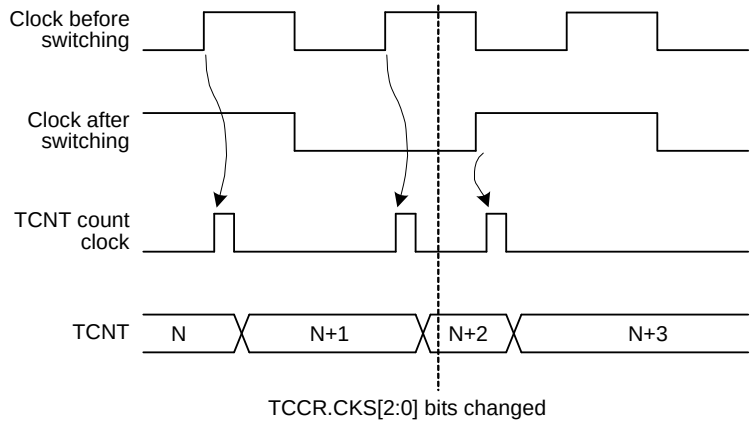
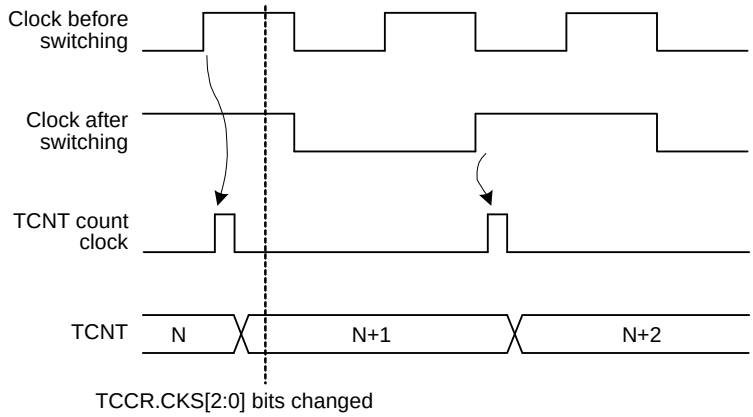
No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Counter Operation
1	Switching from low to low*1	
2	Switching from low to high*2	

Table 29.9 Switching of Internal Clocks and TCNT Operation (2/2)

No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Counter Operation
3	Switching from high to low*4	
4	Switching from high to high	

Note 1. Includes switching from low to stop, and from stop to low.
Note 2. Includes switching from stop to high.
Note 3. Generated because the change of the signal levels is considered as an edge; TCNT counter is incremented.
Note 4. Includes switching from high to stop.

29.8.8 Clock Source Setting with Cascaded Connection

If 16-bit counter mode and compare match count mode are specified at the same time, count clocks for TMR0.TCNT and TMR1.TCNT (TMR2.TCNT and TMR3.TCNT) are not generated, and the counter stops. Do not specify 16-bit counter mode and compare match count mode simultaneously.

29.8.9 Continuous Output of Compare Match Interrupt Signal

When TCORA or TCORB is set to 00h, PCLK/1 is set as the internal clock, and compare match is set as the counter clear source, the TCNT counter remains 00h and is not updated, and a compare match interrupt signal is output continuously to form a flat signal level.
At this time, the interrupt controller cannot detect the second and subsequent interrupts.
Figure 29.16 shows operation timing when the compare match interrupt signal is continuously output.

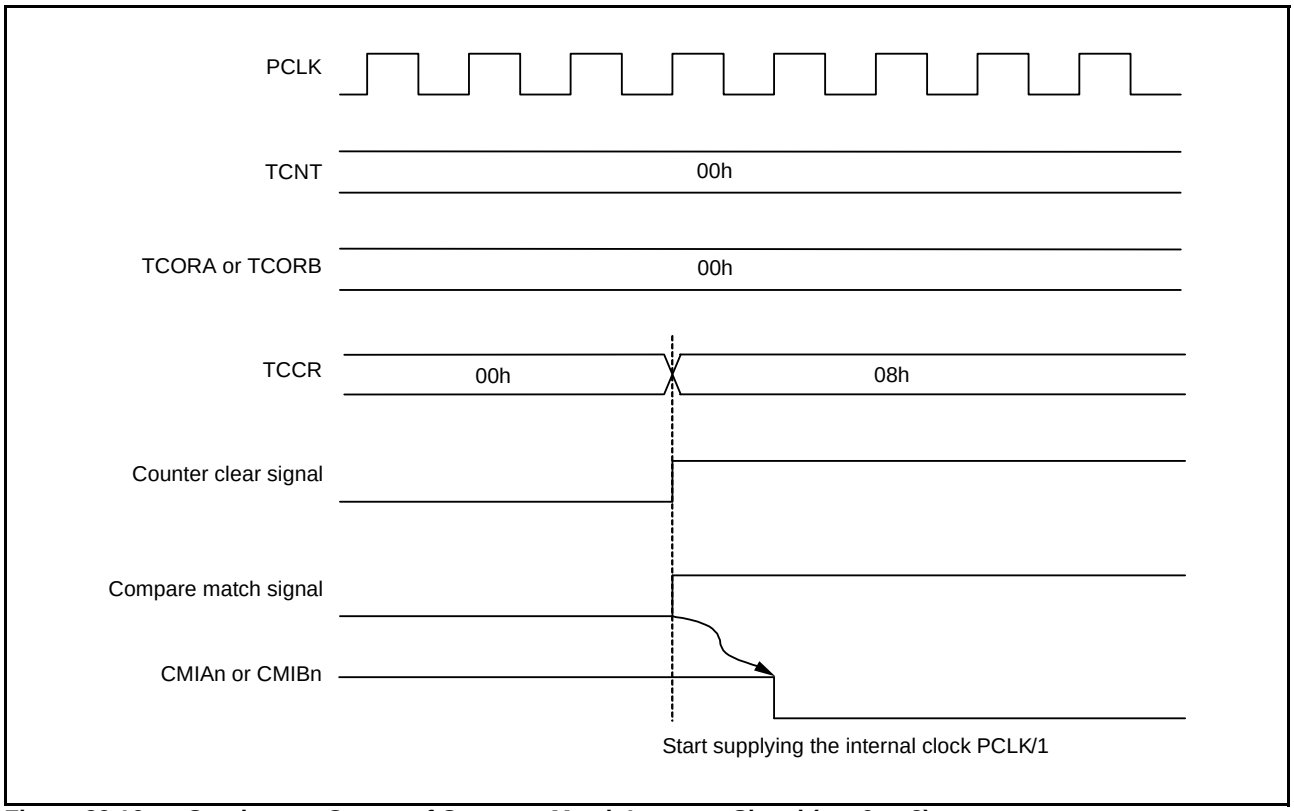


Figure 29.16 Continuous Output of Compare Match Interrupt Signal (n = 0 to 3)

30. Compare Match Timer (CMT)

This MCU has two on-chip compare match timer (CMT) units (unit 0 and unit 1), each consisting of a two-channel 16-bit timer (i.e., a total of four channels). The CMT has a 16-bit counter, and can generate interrupts at set intervals.
In this section, “PCLK” is used to refer to PCLKB.

30.1 Overview

Table 30.1 lists the specifications for the CMT.
Figure 30.1 shows a block diagram of the CMT (unit 0). A two-channel CMT constitutes a unit. Unit 0 and unit 1 are the same in terms of specifications. Compare match timer start register 0 (CMSTR0) and compare match interrupts (CMI0 and CMI1) of unit 0 correspond to compare match timer start register 1 (CMSTR1) and compare match interrupts (CMI2 and CMI3) of unit 1.

Table 30.1 CMT Specifications

Item	Description
Count clocks	• Four frequency dividing clocks One clock from PCLK/8, PCLK/32, PCLK/128, and PCLK/512 can be selected for each channel.
Interrupt	A compare match interrupt can be requested for each channel.
Event link function (output)	An event signal is output upon a CMT1 compare match.
Event link function (input)	Linking to the specified module is possible. CMT1 count start, event counter, or count restart operation is possible.
Low power consumption function	Each unit can be placed in a module stop state.

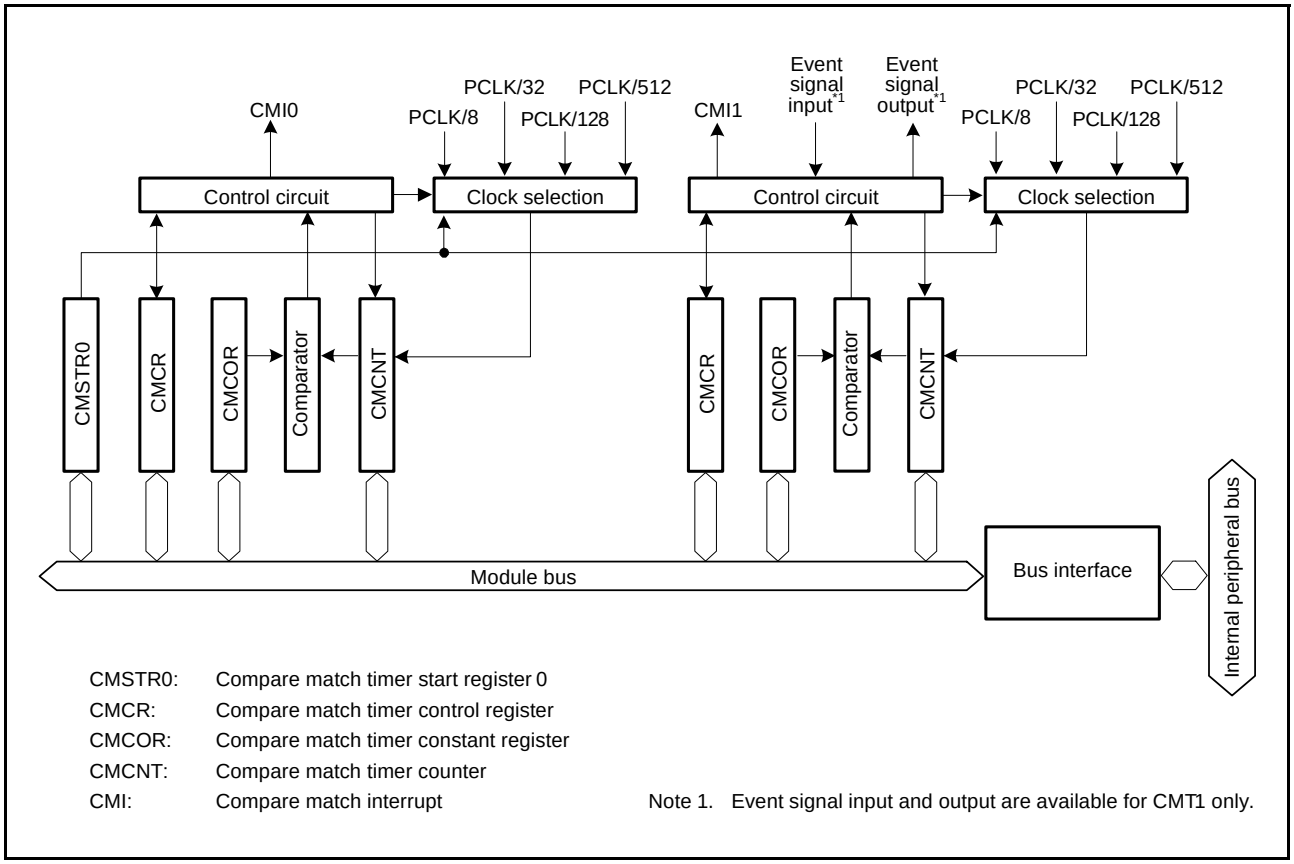


Figure 30.1 CMT (Unit 0) Block Diagram

30.2 Register Descriptions

30.2.1 Compare Match Timer Start Register 0 (CMSTR0)

Address(es): 0008 8000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR1	STR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STR0	Count Start 0	0: CMT0.CMCNT count is stopped. 1: CMT0.CMCNT count is started.	R/W
b1	STR1	Count Start 1	0: CMT1.CMCNT count is stopped. 1: CMT1.CMCNT count is started.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

30.2.2 Compare Match Timer Start Register 1 (CMSTR1)

Address(es): 0008 8010h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR3	STR2
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STR2	Count Start 2	0: CMT2.CMCNT count is stopped. 1: CMT2.CMCNT count is started.	R/W
b1	STR3	Count Start 3	0: CMT3.CMCNT count is stopped. 1: CMT3.CMCNT count is started.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

30.2.3 Compare Match Timer Control Register (CMCR)

Address(es): CMT0.CMCR 0008 8002h, CMT1.CMCR 0008 8008h, CMT2.CMCR 0008 8012h, CMT3.CMCR 0008 8018h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	CMIE	—	—	—	—	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK/8 0 1: PCLK/32 1 0: PCLK/128 1 1: PCLK/512	R/W
b5 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CMIE	Compare Match Interrupt Enable	0: Compare match interrupt (CMIn) disabled 1: Compare match interrupt (CMIn) enabled	R/W
b7	—	Reserved	This bit is read as undefined. The write value should be 1.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CKS[1:0] Bits (Clock Select)

These bits select the count source from four frequency dividing clocks obtained by dividing the peripheral module clock (PCLK).

When the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up on the clock selected with the CKS[1:0] bits.

CMIE Bit (Compare Match Interrupt Enable)

The CMIE bit enables or disables compare match interrupt (CMIn) (n = 0 to 3) generation when the CMCNT counter and the CMCOR register values match.

30.2.4 Compare Match Counter (CMCNT)

Address(es): CMT0.CMCNT 0008 8004h, CMT1.CMCNT 0008 800Ah, CMT2.CMCNT 0008 8014h, CMT3.CMCNT 0008 801Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CMCNT counter is a readable/writable up-counter.
When an frequency dividing clock is selected by the CMCR.CKS[1:0] bits and the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up using the selected clock.
When the value in the CMCNT counter and the value in the CMCOR register match, the CMCNT counter is set to 0000h. At the same time, a compare match interrupt (CMIn) (n = 0 to 3) is generated.

30.2.5 Compare Match Constant Register (CMCOR)

Address(es): CMT0.CMCOR 0008 8006h, CMT1.CMCOR 0008 800Ch, CMT2.CMCOR 0008 8016h, CMT3.CMCOR 0008 801Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The CMCOR register is a readable/writable register to set a value for compare match with the CMCNT counter.

30.3 Operation

30.3.1 Periodic Count Operation

When an frequency dividing clock is selected by the CMCR.CKS[1:0] bits and the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up using the selected clock.
When the value in the counter and the value in the register match, a compare match interrupt (CMIn) (n = 0 to 3) is generated. The CMCNT counter then starts counting up again from 0000h. Figure 30.2 shows the operation of the CMCNT counter.

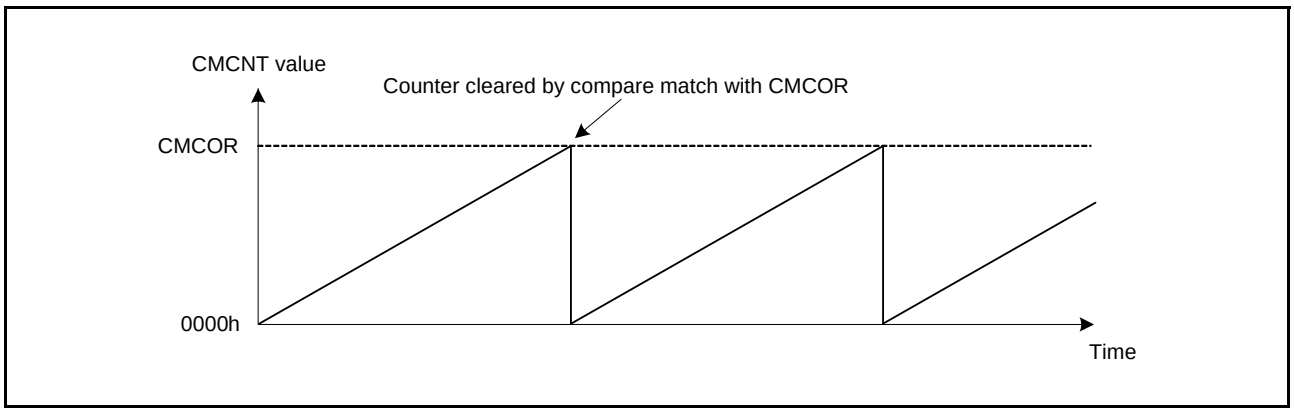


Figure 30.2 CMCNT Counter Operation

30.3.2 CMCNT Count Timing

As the count clock to be input to the CMCNT counter, one of four frequency dividing clocks (PCLK/8, PCLK/32, PCLK/128, and PCLK/512) obtained by dividing the peripheral module clock (PCLK) can be selected with the CMCR.CKS[1:0] bits. Figure 30.3 shows the timing of the CMCNT counter.

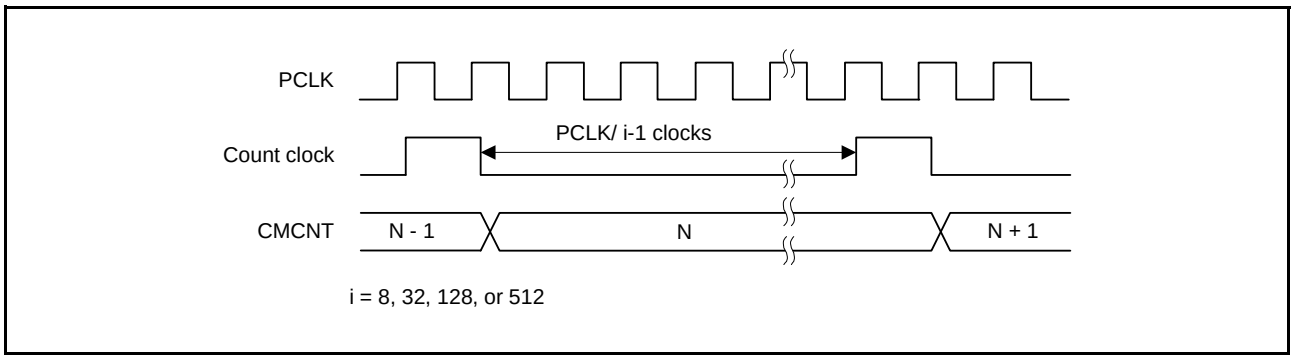


Figure 30.3 CMCNT Count Timing

30.4 Interrupts

30.4.1 Interrupt Sources

The CMT has channels and each of them to which a different vector address is allocated has a compare match interrupt (CMIn) (n = 0 to 3). When a compare match interrupt occurs, the corresponding interrupt request is output.
When the interrupt request is used to generate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 15, Interrupt Controller (ICUA).

Table 30.2 CMT Interrupt Sources

Name	Interrupt Sources	DTC Activation	DMAC Activation
CMI0	Compare match in CMT0	Possible	Possible
CMI1	Compare match in CMT1	Possible	Possible
CMI2	Compare match in CMT2	Possible	Possible
CMI3	Compare match in CMT3	Possible	Possible

30.4.2 Timing of Compare Match Interrupt Generation

When the CMCNT counter and the CMCOR register match, a compare match interrupt (CMIn) (n = 0 to 3) is generated. A compare match signal is generated at the last state in which the values match (the timing when the CMCNT counter updates the matched count value). That is, after a match between the CMCOR register and the CMCNT counter, the compare match signal is not generated until the next the CMCNT counter input clock.
Figure 30.4 shows the timing of a compare match interrupt.

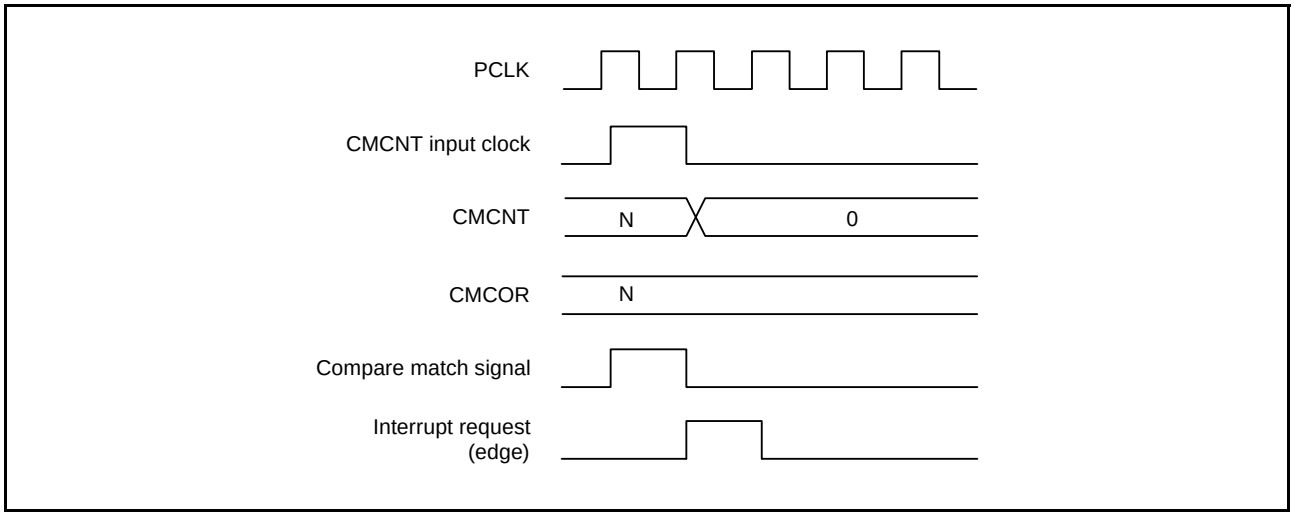


Figure 30.4 Timing of a Compare Match Interrupt

30.5 Link Operations by ELC

30.5.1 Event Signal Output to ELC

The CMT uses the event link controller (ELC) to perform link operation to a preset module using the interrupt request signal as the event signal. The CMT outputs the event signal upon a CMT1 compare match.
The event signal can be output regardless of the setting of the corresponding interrupt request enable bit (CMTn.CMCR.CMIE).

30.5.2 CMT Operation When Receiving an Event Signal from ELC

The CMT can perform either of the following operations upon the event preset by the ELSR7 register of the ELC.

(1) Count Start

When the CMT count start operation is selected by the ELOPC register of the ELC and the event specified by ELSR7 occurs, the CMSTR0.STR1 bit is set to 1, starting the CMT count operation.
However, if the specified event occurs while the CMSTR0.STR1 bit is 1, the event is ignored.

(2) Event Count

When the CMT event count operation is selected by the ELOPC register of the ELC and the event specified by ELSR7 occurs with the CMSTR0.STR1 bit being 1, the events are counted as the count source regardless of the CMT1.CMCR.CKS[1:0] bit setting. Reading the counter value returns the number of events that have been actually input.

(3) Count Restart

When the CMT count restart operation is selected by the ELOPC register of the ELC and the event specified by ELSR7 occurs, the CMT1.CMCNT counter value is modified to the initial value. If the CMSTR0.STR1 bit is 1 here, the count operation can be continued.

30.5.3 Notes on Operating CMT According to an Event Signal from ELC

The following describes the notes on operating the CMT using the event link feature.

(1) Count Start

When the event specified by ELSR7 occurs during the write cycle to the CMSTR0.STR1 bit, the cycle is not completed; setting 1 according to the event occurrence takes priority.

(2) Event Count

When the event specified by ELSR7 occurs during the write cycle to the CMT1.CMCNT, the cycle is not completed; event count operation according to the event occurrence takes priority.

(3) Count Restart

When the event specified by ELSR7 occurs during the write cycle to the CMT1.CMCNT, the cycle is not completed; count value initialization according to the event occurrence takes priority.

30.6 Usage Notes

30.6.1 Setting the Module Stop Function

The CMT can be enabled or disabled using the module stop control register. After a reset, the CMT is in the module stop state. The registers can be accessed by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

30.6.2 Conflict between CMCNT Counter Writing and Compare Match

When the compare match signal is generated while writing to the CMCNT counter, clearing the CMCNT counter has priority over writing to it. In this case, the CMCNT counter is not written to. Figure 30.5 shows the timing to clear the CMCNT counter.

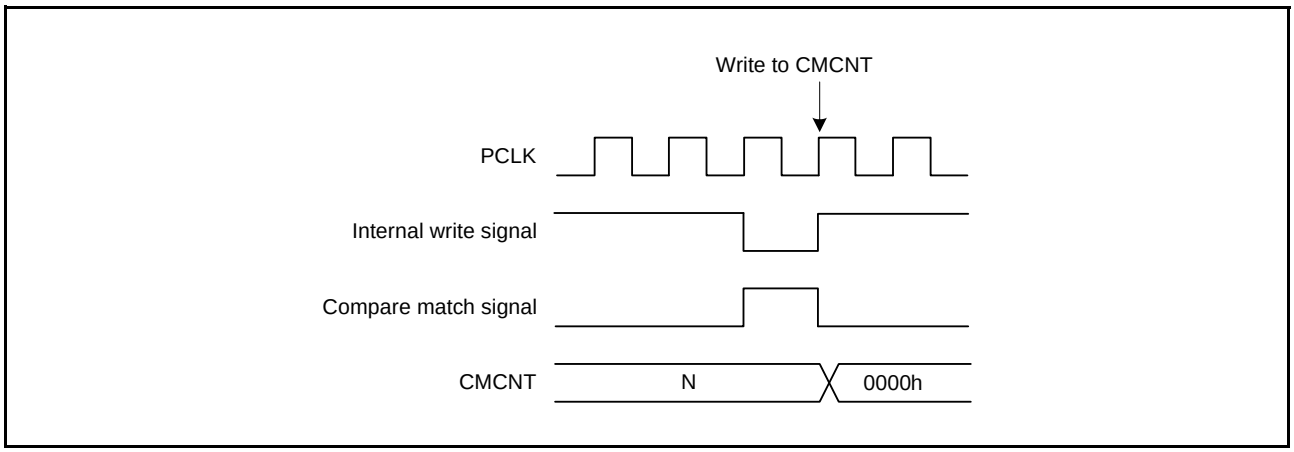


Figure 30.5 Conflict between CMCNT Counter Writing and Compare Match

30.6.3 Conflict between CMCNT Counter Writing and Incrementing

If writing to the counter and the incrementing conflict, the writing has priority over the incrementing. Figure 30.6 shows the timing to write the CMCNT counter.

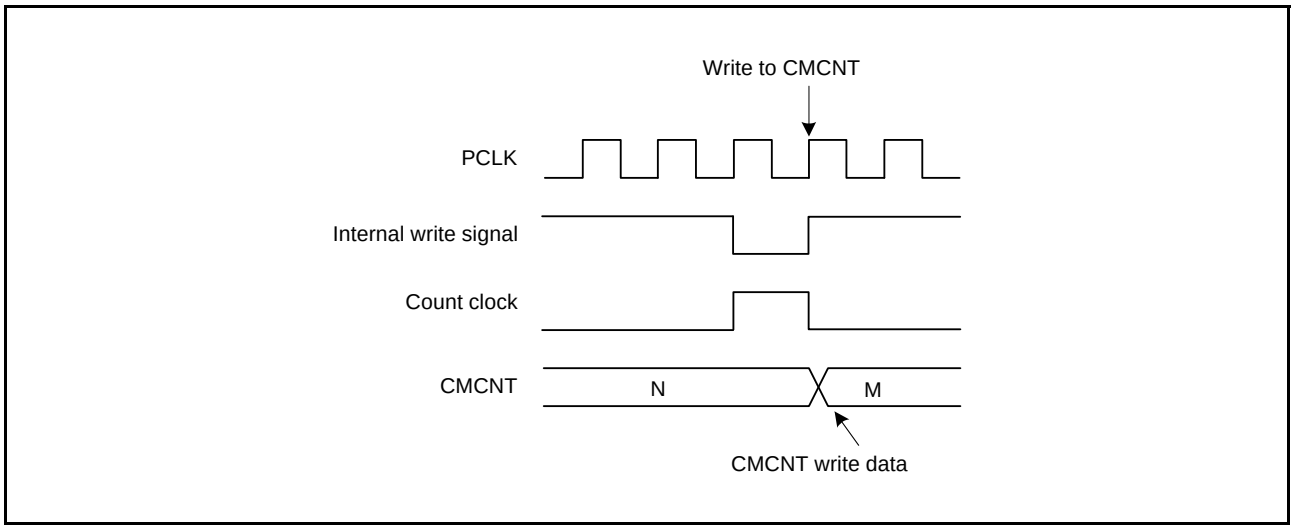


Figure 30.6 Conflict between CMCNT Counter Writing and Incrementing

31. Compare Match Timer W (CMTW)

This MCU includes two units (unit 0 and unit 1) with one channel of 32-bit compare match timer W (CMTW).
CMTW has a 32-bit counter and can generate interrupts each time a set period elapses.
In this section, “PCLK” is used to refer to PCLKB.

31.1 Overview

Table 31.1 shows the specifications of the CMTW.
Figure 31.1 shows a block diagram of the CMTW0 and Figure 31.2 shows a block diagram of the CMTW1.

Table 31.1 CMTW Specifications

Item	Function
Number of channels	Two channels (unit 0, unit 1)
Timer counter	16-bit/32-bit selectable up-counter The counter returns to 0000 0000h after a compare match.
Prescaler	Four dividing clocks are output. Selectable from any of PCLK/8, PCLK/32, PCLK/128, and PCLK/512
Input capture	Up to two input capture input signals available.
Output compare	Up to two output compare output signals available.
Compare match	One compare match available (no output compare output pin used).
Interrupts	Compare match interrupt Input capture 0 and 1 interrupts Output compare 0 and 1 interrupts
Event link function (output) (Unit 0)	Compare match
Event link function (input) (Unit 0)	One of the following operations is enabled after an event is accepted: • Count start operation • Event count operation • Count restart operation
Low power consumption function	Each unit can be placed in a module stop state.

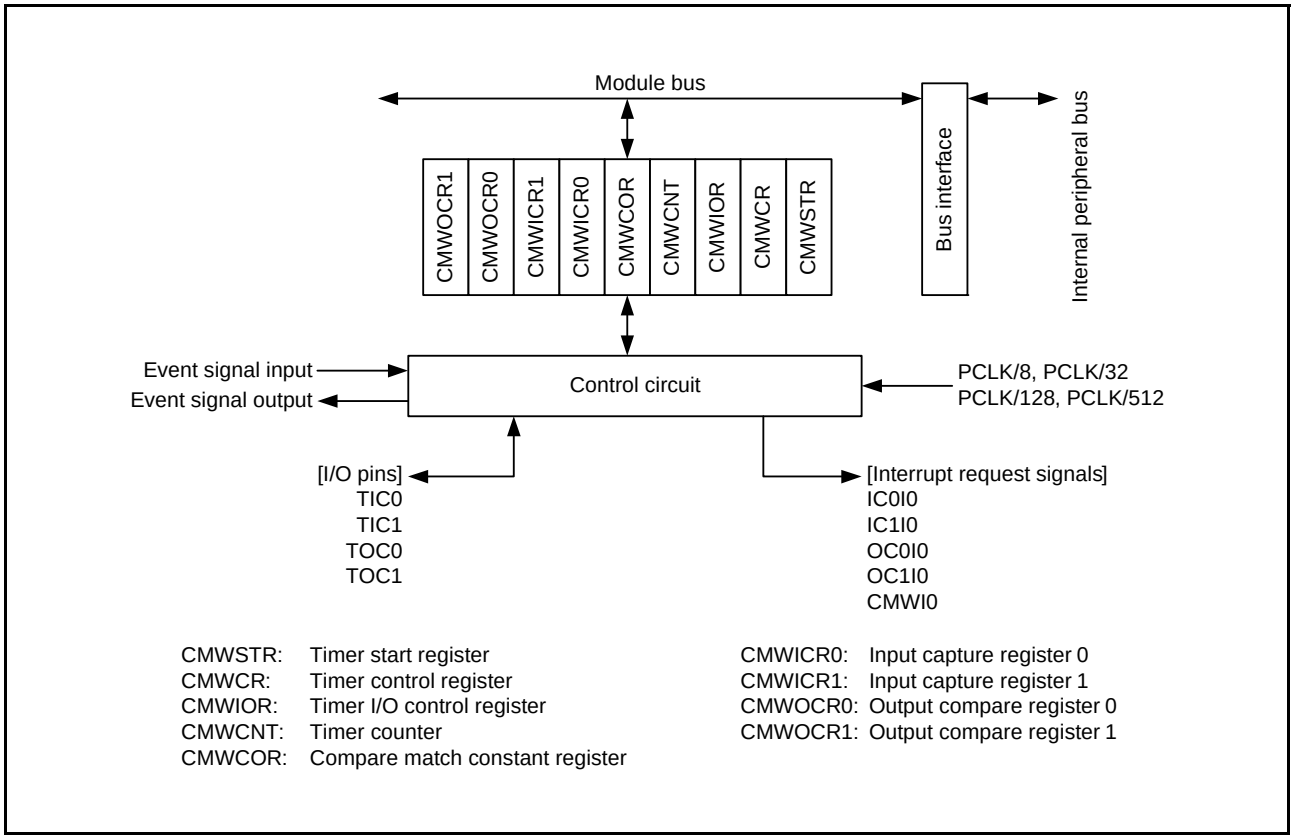


Figure 31.1 CMTW0 Block Diagram

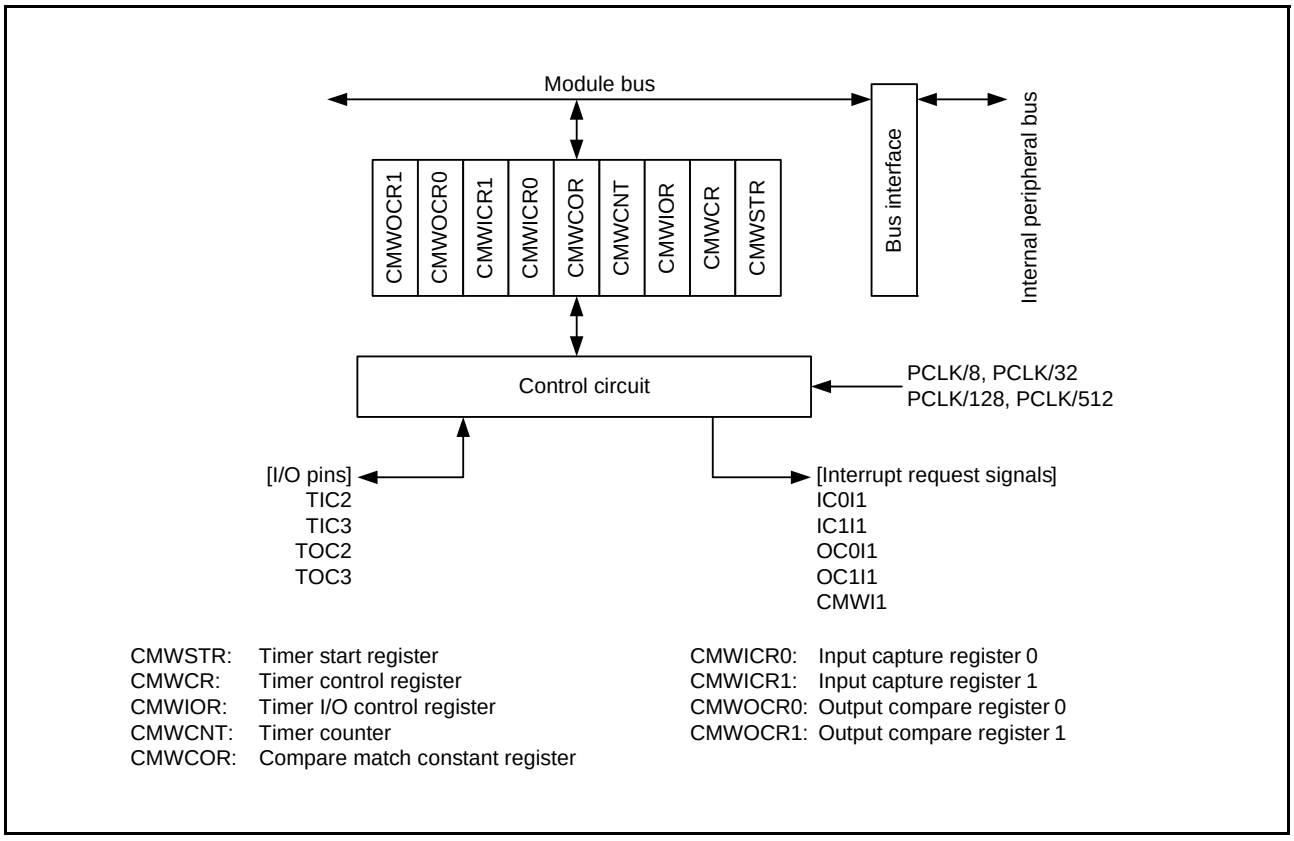


Figure 31.2 CMTW1 Block Diagram

Table 31.2 shows the CMTW pin configuration.

Table 31.2 CMTW Pin Configuration

Unit	Pin Name	I/O	Description
CMTW0	TIC0	Input	Input capture input for the CMTW0.CMWICR0 register
	TIC1	Input	Input capture input for the CMTW0.CMWICR1 register
	TOC0	Output	Output compare output for the CMTW0.CMWOCR0 register
	TOC1	Output	Output compare output for the CMTW0.CMWOCR1 register
CMTW1	TIC2	Input	Input capture input for the CMTW1.CMWICR0 register
	TIC3	Input	Input capture input for the CMTW1.CMWICR1 register
	TOC2	Output	Output compare output for the CMTW1.CMWOCR0 register
	TOC3	Output	Output compare output for the CMTW1.CMWOCR1 register

31.2 Register Descriptions

31.2.1 Timer Start Register (CMWSTR)

Address(es): CMTW0.CMWSTR 0009 4200h, CMTW1.CMWSTR 0009 4280h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STR	Counter Start	0: CMWCNT counter count is stopped. (The value immediately before count operation stops is retained and the count operation is stopped.) 1: CMWCNT counter count is started.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

STR Bit (Counter Start)

Specifies whether the timer counter operates or is stopped. The relevant prescaler operates or is stopped according to the settings of the STR bit.

31.2.2 Timer Control Register (CMWCR)

Address(es): CMTW0.CMWCR 0009 4204h, CMTW1.CMWCR 0009 4284h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CCLR[2:0]			—	—	—	CMS	—	OC1IE	OC0IE	IC1IE	IC0IE	CMWIE	—	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK/8 0 1: PCLK/32 1 0: PCLK/128 1 1: PCLK/512	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	CMWIE	Compare Match Interrupt Request Enable	0: Interrupt request (CMWI) disabled 1: Interrupt request (CMWI) enabled	R/W
b4	IC0IE	Input Capture 0 Interrupt Request Enable	0: Interrupt request (IC0I) disabled 1: Interrupt request (IC0I) enabled	R/W
b5	IC1IE	Input Capture 1 Interrupt Request Enable	0: Interrupt request (IC1I) disabled 1: Interrupt request (IC1I) enabled	R/W
b6	OC0IE	Output Compare 0 Interrupt Request Enable	0: Interrupt request (OC0I) disabled 1: Interrupt request (OC0I) enabled	R/W
b7	OC1IE	Output Compare 1 Interrupt Request Enable	0: Interrupt request (OC1I) disabled 1: Interrupt request (OC1I) enabled	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9	CMS	Timer Counter Size	0: 32 bits 1: 16 bits	R/W
b12 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b13	CCLR[2:0]	Counter Clear	b15 b14 b13 0 0 0: CMWCNT counter cleared by CMWCOR register compare match 0 0 1: Clearing of CMWCNT counter disabled 0 1 0: Clearing of CMWCNT counter disabled 0 1 1: Clearing of CMWCNT counter disabled 1 0 0: CMWCNT counter cleared by CMWICR0 register input capture 1 0 1: CMWCNT counter cleared by CMWICR1 register input capture 1 1 0: CMWCNT counter cleared by CMWOCR0 register compare match 1 1 1: CMWCNT counter cleared by CMWOCR1 register compare match	R/W

The CMWCR register should be set while the timer counter (CMWCNT) operation is stopped.

CKS[1:0] Bits (Clock Select)

Select the clock to be input to the CMWCNT counter among four internal clocks obtained by dividing the peripheral module clock (PCLK). When the CMWSTR.STR bit is set to 1, the CMWCNT counter starts counting based on the clock selected with the CMWCR.CKS[1:0] bits.

CMWIE Bit (Compare Match Interrupt Request Enable)

Enables or disables compare match interrupt request (CMWI) generation when the CMWCNT counter and the CMWCOR register values match.

IC0IE Bit (Input Capture 0 Interrupt Request Enable)

Enables or disables input capture 0 interrupt request (IC0I) generation when input capture is generated in the CMWICR0 register.

IC1IE Bit (Input Capture 1 Interrupt Request Enable)

Enables or disables input capture 1 interrupt request (IC1I) generation when input capture is generated in the CMWICR1 register.

OC0IE Bit (Output Compare 0 Interrupt Request Enable)

Enables or disables output compare 0 interrupt request (OC0I) generation when the CMWCNT counter and the CMWOCR0 register values match.

OC1IE Bit (Output Compare 1 Interrupt Request Enable)

Enables or disables output compare 1 interrupt request (OC1I) generation when the CMWCNT counter and CMWOCR1 register values match.

CMS Bit (Timer Counter Size)

Selects either 16 or 32 bits as the size of the timer counter (CMWCNT). The size selected with the CMS bit is valid in the CMWCOR, CMWICR0, CMWICR1, CMWOCR0, and CMWOCR1 registers.

CCLR[2:0] Bits (Counter Clear)

Select the CMWCNT counter clearing source.

31.2.3 Timer I/O Control Register (CMWIOR)

Address(es): CMTW0.CMWIOR 0009 4208h, CMTW1.CMWIOR 0009 4288h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMWE	—	OC1E	OC0E	OC1[1:0]		OC0[1:0]		—	—	IC1E	IC0E	IC1[1:0]		IC0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	IC0[1:0]	Input Capture 0 Control	b1 b0 0 0: Input capture at the rising edge 0 1: Input capture at the falling edge 1 0: Input capture at both edges 1 1: Setting prohibited	R/W
b3, b2	IC1[1:0]	Input Capture 1 Control	b3 b2 0 0: Input capture at the rising edge 0 1: Input capture at the falling edge 1 0: Input capture at both edges 1 1: Setting prohibited	R/W
b4	IC0E	Input Capture 0 Enable	0: Input capture 0 operation disabled 1: Input capture 0 operation enabled	R/W
b5	IC1E	Input Capture 1 Enable	0: Input capture 1 operation disabled 1: Input capture 1 operation enabled	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	OC0[1:0]	Output Compare 0 Control	b9 b8 0 0: Retains the output value.*1 0 1: Initially outputs low and toggles the output value upon compare match. 1 0: Initially outputs high and toggles the output value upon compare match. 1 1: Setting prohibited	R/W
b11, b10	OC1[1:0]	Output Compare 1 Control	b11 b10 0 0: Retains the output value.*1 0 1: Initially outputs low and toggles the output value upon compare match. 1 0: Initially outputs high and toggles the output value upon compare match. 1 1: Setting prohibited	R/W
b12	OC0E	Output Compare 0 Enable	0: Output compare 0 operation disabled 1: Output compare 0 operation enabled	R/W
b13	OC1E	Output Compare 1 Enable	0: Output compare 1 operation disabled 1: Output compare 1 operation enabled	R/W
b14	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15	CMWE	Compare Match Enable	0: Compare match operation disabled 1: Compare match operation enabled	R/W

Note 1. After reset, low is output until the CMWIOR register is set.

31.2.4 Timer Counter (CMWCNT)

Address(es): CMTW0.CMWCNT 0009 4210h, CMTW1.CMWCNT 0009 4290h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CMWCNT counter is a readable/writable up-counter.
Before starting count operation, the timer control register (CMWCR) should be set. When the 16-bit counter size is selected with the CMS bit in the timer control register (CMWCR), bits 15 to 0 in this register are valid. When writing to the CMWCNT counter, write data in 32-bit units with the upper bits set to 0000h. The CMWCNT counter can only be accessed in longword units.
When the CMWSTR.STR bit is set to 1, the CMWCNT counter starts counting. When the CMWSTR.STR bit is set to 0, the CMWCNT counter retains the value immediately before a stop of counting and stops counting.

31.2.5 Compare Match Constant Register (CMWCOR)

Address(es): CMTW0.CMWCOR 0009 4214h, CMTW1.CMWCOR 0009 4294h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The CMWCOR register is a readable/writable register that specifies the time up to a compare match between the timer counter (CMWCNT) value and CMWCOR value. When the 16-bit counter size is selected with the CMS bit in the timer control register (CMWCR), bits 15 to 0 in this register are valid. When writing to the CMWCOR register, write data in 32-bit units with the upper bits set to 0000h. The CMWCOR register can only be accessed in longword units. To detect an overflow, set the CMWCOR register value to FFFF FFFFh (32-bit count operation) or 0000 FFFFh (16-bit count operation). When the CMWCNT counter is set to 0, a compare match interrupt request (CMWI) can be used as an overflow detection signal.

31.2.6 Input Capture Registers 0 and 1 (CMWICR0 and CMWICR1)

Address(es): CMTW0.CMWICR0 0009 4218h, CMTW0.CMWICR1 0009 421Ch, CMTW1.CMWICR0 0009 4298h, CMTW1.CMWICR1 0009 429Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CMWICR register is a read-only register in which the CMWCNT value is stored when an input capture is generated. When the 16-bit counter size is selected with the CMS bit in the timer control register (CMWCR), bits 15 to 0 in these registers are valid. Writing to these registers is invalid. The CMWICR register can only be accessed in longword units.

31.2.7 Output Compare Registers 0 and 1 (CMWOCR0 and CMWOCR1)

Address(es): CMTW0.CMWOCR0 0009 4220h, CMTW0.CMWOCR1 0009 4224h, CMTW1.CMWOCR0 0009 42A0h, CMTW1.CMWOCR1 0009 42A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The CMWOCR register is a readable/writable register that set the value to be compared when an output compare is generated.
When the 16-bit counter size is selected with the CMS bit in the timer control register (CMWCR), bits 15 to 0 of these registers become valid. When writing to these registers, write data in 32-bit units with the upper bits set to 0000h.
The CMWOCR register can only be accessed in longword units. The initial value of CMWOCR0 and CMWOCR1 registers is FFFF FFFFh.

31.3 Operation

When the CMWCR register is set and then the STR bit in CMWSTR is set to 1, the CMTW starts count operation. Setting the CMWSTR.STR bit to 0 enables the CMWCNT counter to retain the value immediately before a stop of counting and stop counting. Setting the CMWIOR register enables using the compare match function, input capture input function, and output compare output function.

31.3.1 Period Count Operation

When the counter clock is selected by using the CMWCR.CKS[1:0] bits and the CMWSTR.STR bit is set to 1, the CMWCNT counter starts counting cycles of the selected clock. When clearing of the counter is selected by the CMWCR.CCLR[2:0] bits and the counter clearing source is generated, the CMWCNT counter is cleared to 0000 0000h and continues counting. When clearing of the counter is not selected, an overflow is generated when FFFF FFFFh changes to 0000 0000h during 32-bit count operation and 0000 FFFFh changes to 0000 0000h during 16-bit count operation, and the CMWCNT counter continues counting.

31.3.2 Compare Match Function

When the values of the CMWCNT counter and CMWCOR register match, the CMWCNT counter is cleared to 0000 0000h. At this time, a compare match interrupt request (CMWI) is generated. The CMWCNT counter restarts counting from 0000 0000h.
To enable overflow detection, the CMWCOR register value should be set to FFFF FFFFh (when the counter size is 32 bits) or 0000 FFFFh (when the counter size is 16 bits). When the values of the CMWCNT counter and CMWCOR register match, the CMWCNT counter is cleared to 0000 0000h. In this case, the compare match interrupt request (CMWI) is generated. The CMWCNT counter then restarts counting from 0000 0000h.
Figure 31.3 shows an example of procedure for setting compare match operation.

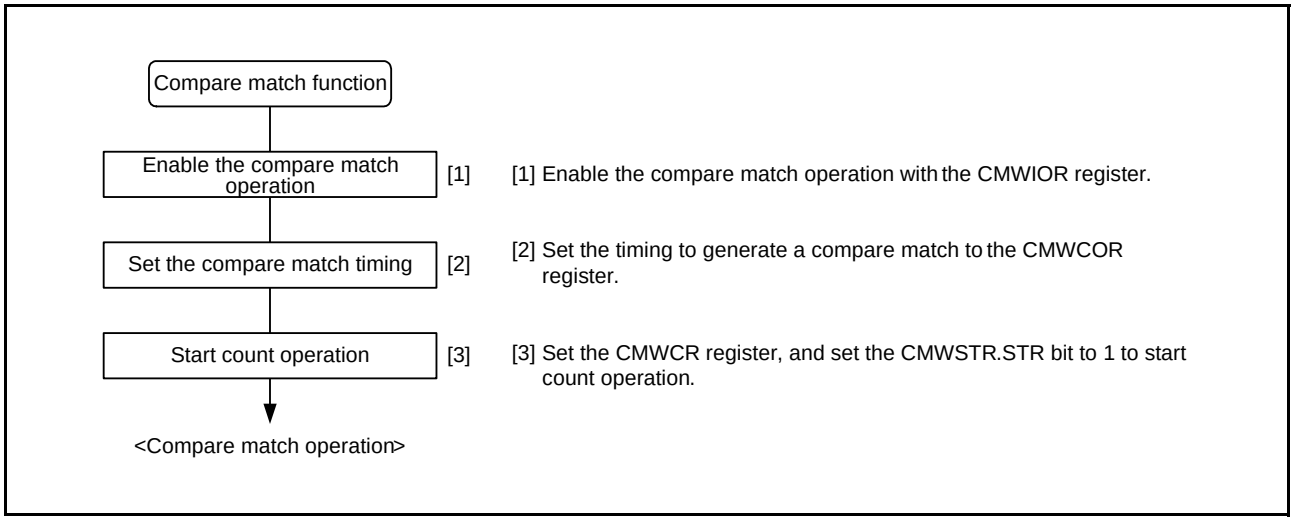


Figure 31.3 Procedure for Setting Compare Match Operation

Figure 31.4 shows an example when compare match with CMWCOR is set as a counter clearing source.

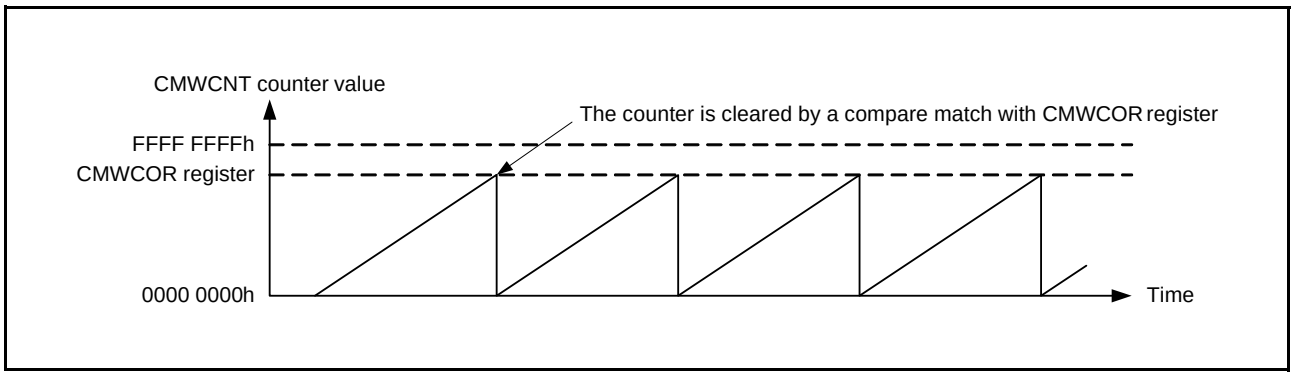


Figure 31.4 Example of Compare Match Operation

Figure 31.5 shows an example when CMWCOR is set to FFFF FFFFh and an overflow is detected.

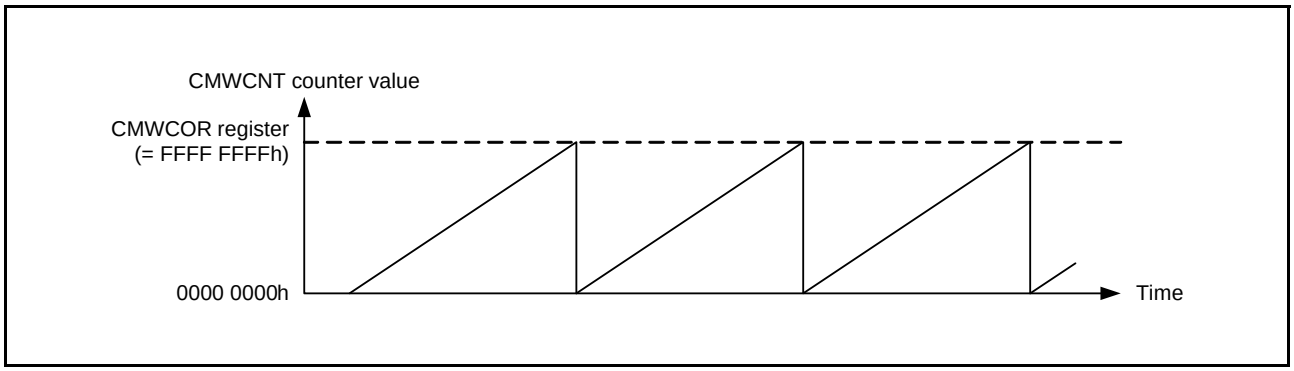


Figure 31.5 Example of Compare Match Operation (Overflow Detected)

31.3.3 Output Compare Function

The output compare function can be used for toggle waveform output. When the CMWCNT counter value matches either of the values of the CMWOCR0 or CMWOCR1 register, the output compare interrupt request (OC0I or OC1I) is generated. Figure 31.6 shows an example of procedure for setting output compare operation.

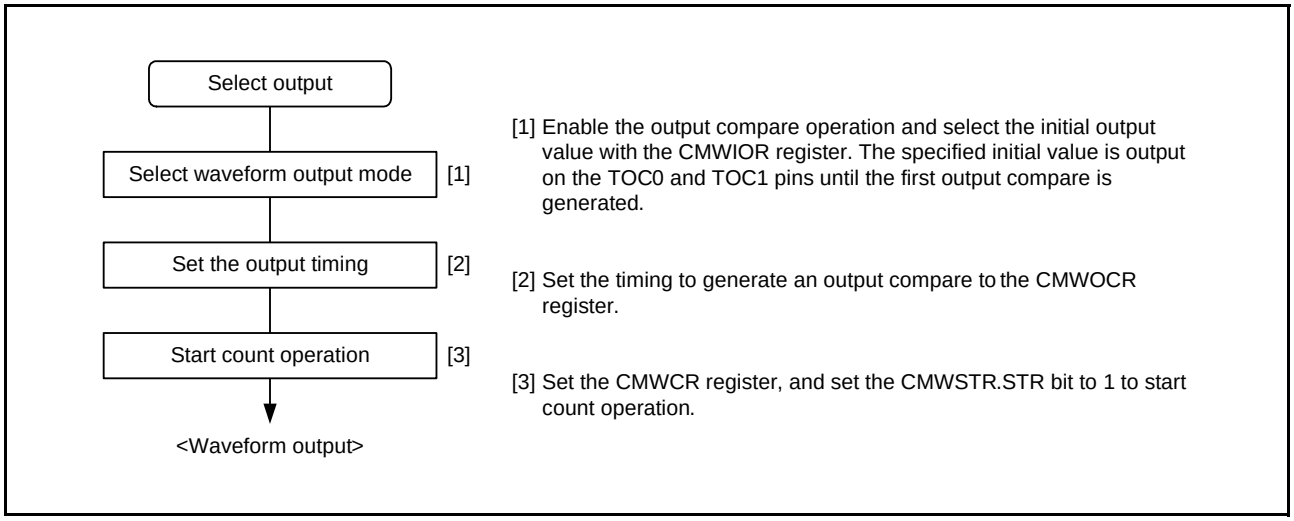


Figure 31.6 Procedure for Setting Output Compare Operation

Figure 31.7 shows an example of toggle waveform output from the TOC0 and TOC1 pins when the counter is set to be cleared by compare match with the CMWOCR1 register.

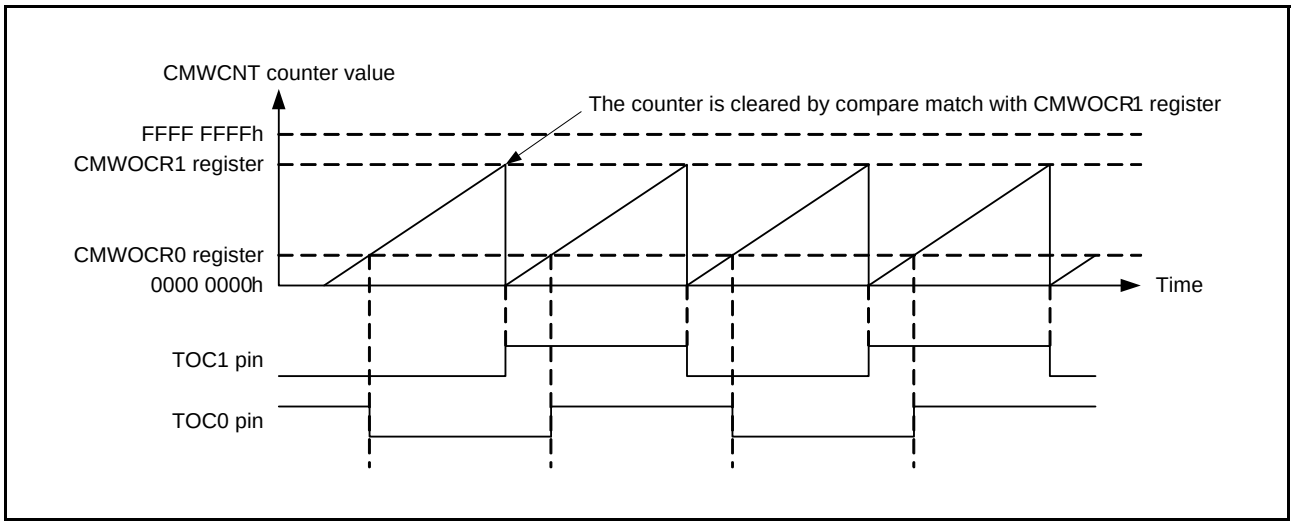


Figure 31.7 Example of Output Compare Operation (Unit 0)

31.3.4 Input Capture Function

Through detecting the edge on the TIC0 and TIC1 pin input, the CMWCNT counter value can be transferred to the CMWICR0 and CMWICR1 registers, respectively. The edges to be detected can be selected from among the rising edge alone, falling edge alone, and both the rising and falling edges. When the CMWCNT counter value is transferred to the CMWICR0 or CMWICR1 register using the input capture operation, an input capture interrupt request (IC0I or IC1I) is generated. Figure 31.8 shows an example of procedure for setting input capture operation.

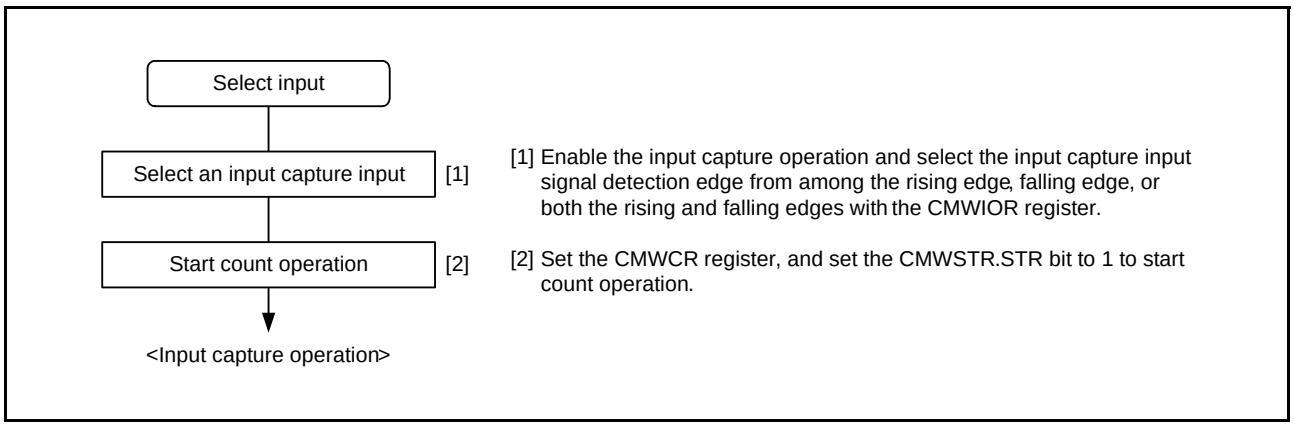


Figure 31.8 Procedure for Setting Input Capture Operation

Figure 31.9 shows an example in which both edges are selected for the TIC0 pin input capture detection edge and the falling edge for the TIC1 pin, and the CMWCNT counter is cleared by a CMWICR1 register input capture.

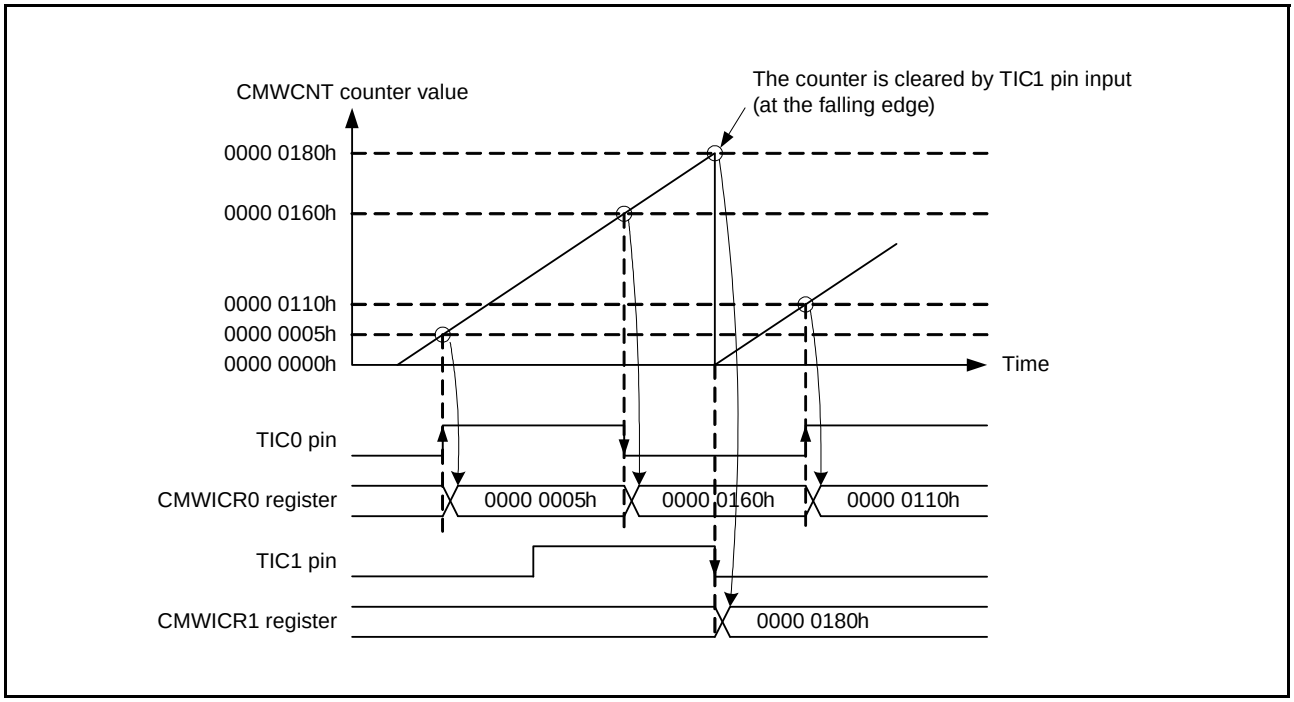


Figure 31.9 Example of Input Capture Operation (Unit 0)

31.3.5 Counter Size

With the CMTW, either 16 or 32 bits can be selected as the counter size by using the CMWCR.CMS bit.
When the counter is used as a 16-bit counter, set the value of the CMWCOR register in 32-bit units with the upper 16 bits set to 0000h. 0000 FFFFh should be set to detect an overflow. Similarly, set the values of the CMWOCR0 and CMWOCR1 registers in 32-bit units with the upper 16 bits set to 0000h. Read the CMWOCR0 and CMWOCR1 registers in 32-bit units. The upper 16 bits can be read as 0000h.

31.3.6 Count Timing of CMWCNT Counter

By setting the CMWCR.CKS[1:0] bits, one of four clocks (PCLK/8, PCLK/32, PCLK/128, and PCLK/512) obtained by dividing the peripheral module clock (PCLK) can be selected as the counter clock to be input to the CMWCNT counter. Figure 31.10 shows the timing.

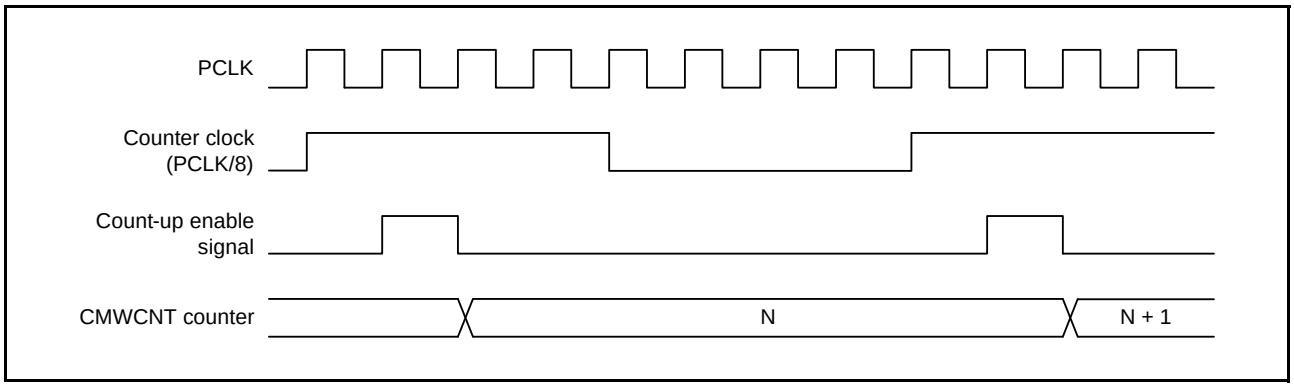


Figure 31.10 Count Timing (PCLK/8)

31.3.7 Output Compare Output Timing

A compare match signal is generated in the last state in which the CMWOCR register and CMWCNT counter values match (the CMWCNT counter value is updated immediately after the state). The compare match signal is generated if the CMWCNT count-up enable signal is input after a match between the CMWOCR register and CMWCNT counter values. When a compare match signal is generated, output of the output compare pin (TOC pin) is toggled.
Figure 31.11 shows output compare output timing.

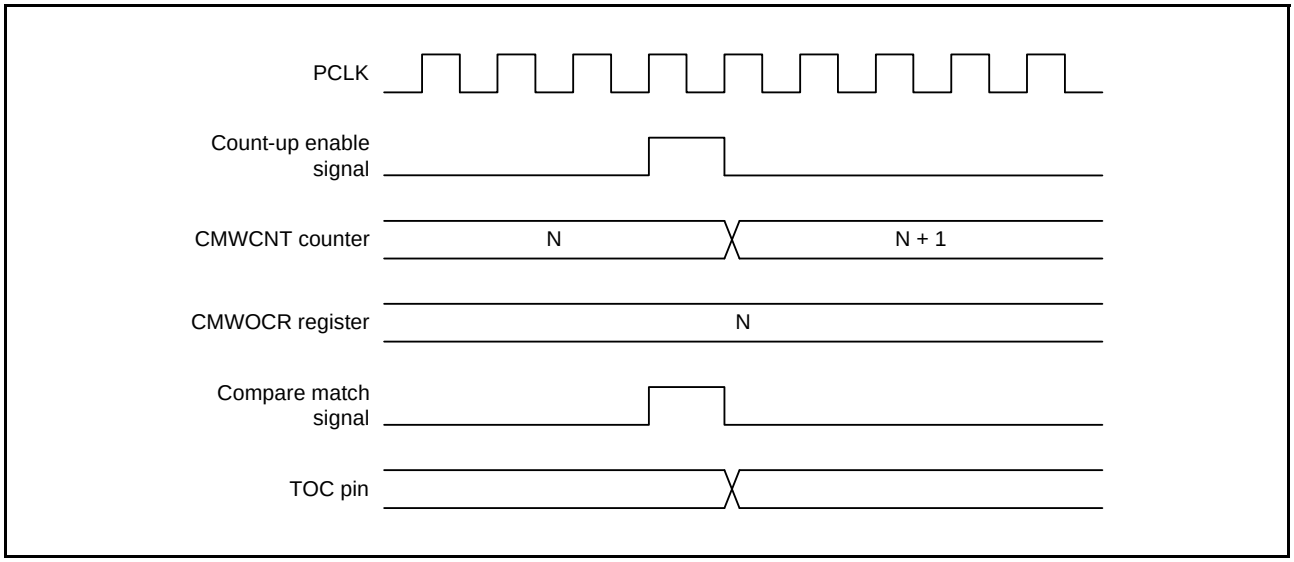


Figure 31.11 Output Compare Output Timing

31.3.8 Input Capture Timing

Figure 31.12 shows the timing of input capture operation at both edges.
When the edge of the TIC0 and TIC1 pins is detected, the CMWCNT counter value is transferred to the CMWICR0 and CMWICR1 registers, respectively.

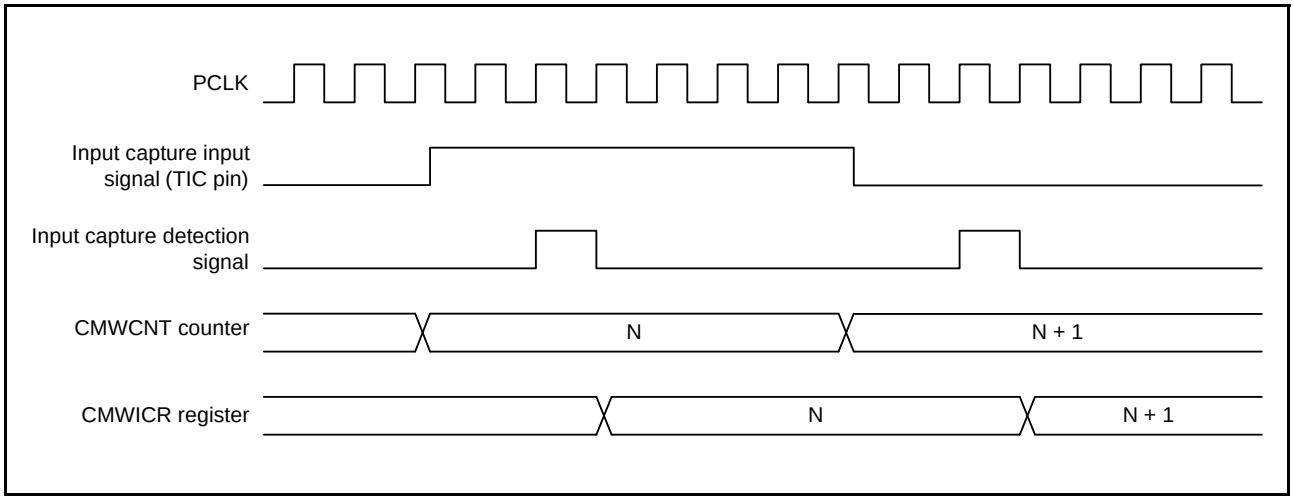


Figure 31.12 Input Capture Timing (Unit 0, Both-Edge Detection)

31.4 Interrupts

31.4.1 CMTW Interrupt Sources and DTC/DMAC Transfer Requests

The CMTW has five interrupt sources: two input capture interrupt requests (IC0I and IC1I), two output compare interrupt requests (OC0I and OC1I), and a compare match interrupt request (CMWI).

Table 31.3 shows the interrupt sources. The interrupt sources can be enabled or disabled using the IC0IE, IC1IE, OC0IE, OC1IE, and CMWIE bits in CMWCR and are separately generated to the interrupt controller.

Each interrupt request can activate the direct memory access controller (DMAC) or data transfer controller (DTC). When the DMAC is used for data transfer, an interrupt request is not generated to the CPU. For generating an interrupt request to the CPU during data transfer using the DTC, refer to section 20, Data Transfer Controller (DTCa).

Table 31.3 CMTW Interrupt Sources

Unit	Name	Interrupt Request	Interrupt Request Enable Bit	DMAC/DTC Activation
CMTW0	CMWI0	Compare match of CMTW0.CMWCOR register	CMTW0.CMWCR.CMWIE	Possible
	IC0I0	Input capture of CMTW0.CMWICR0 register	CMTW0.CMWCR.IC0IE	Possible
	IC1I0	Input capture of CMTW0.CMWICR1 register	CMTW0.CMWCR.IC1IE	Possible
	OC0I0	Output compare of CMTW0.CMWOCR0 register	CMTW0.CMWCR.OC0IE	Possible
	OC1I0	Output compare of CMTW0.CMWOCR1 register	CMTW0.CMWCR.OC1IE	Possible
CMTW1	CMWI1	Compare match of CMTW1.CMWCOR register	CMTW1.CMWCR.CMWIE	Possible
	IC0I1	Input capture of CMTW1.CMWICR0 register	CMTW1.CMWCR.IC0IE	Possible
	IC1I1	Input capture of CMTW1.CMWICR1 register	CMTW1.CMWCR.IC1IE	Possible
	OC0I1	Output compare of CMTW1.CMWOCR0 register	CMTW1.CMWCR.OC0IE	Possible
	OC1I1	Output compare of CMTW1.CMWOCR1 register	CMTW1.CMWCR.OC1IE	Possible

31.4.2 Timing of Compare Match Interrupt Generation

When the values of the CMWCNT counter and CMWCOR register match, a compare match interrupt request (CMWI) is generated. The compare match signal is generated at the end of the cycle where the values matched (i.e. when the CMWCNT counter is updated from the matching counter value).The compare match signal, therefore, is not generated until a count-up enable signal is generated after the values of the CMWCNT counter and CMWCOR register have matched. Figure 31.13 shows the timing of compare match interrupt generation.

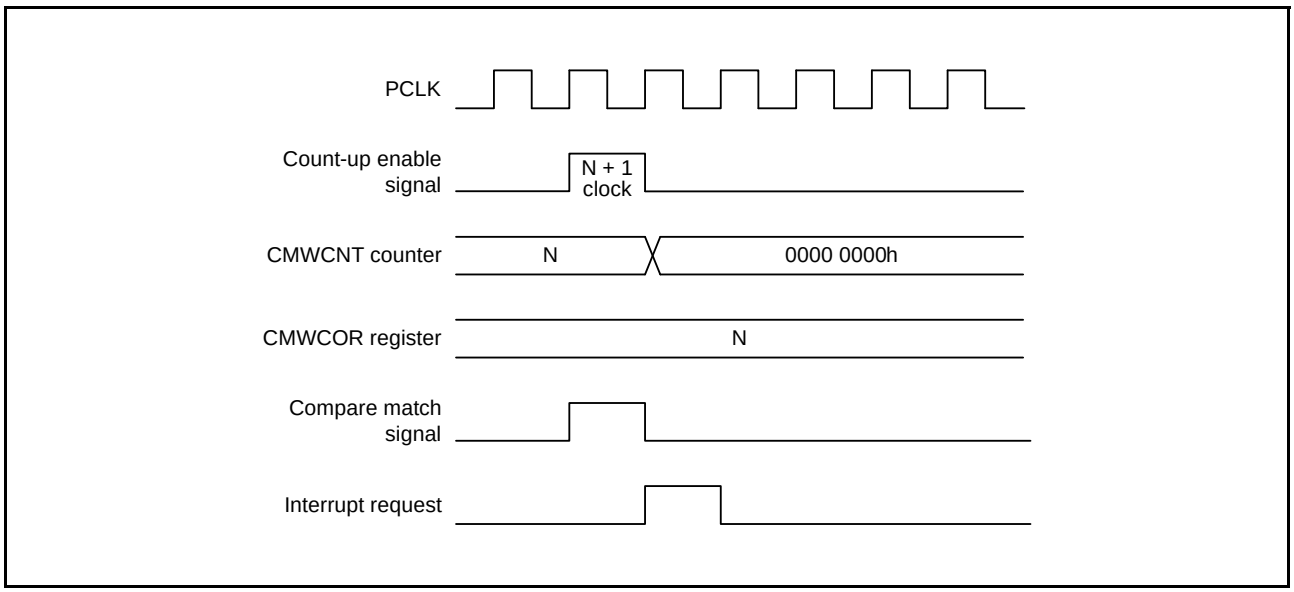


Figure 31.13 Timing of Compare Match Interrupt Generation

(a) Timing of Output Compare Interrupt Generation

Figure 31.14 shows the timing of output compare interrupt generation.

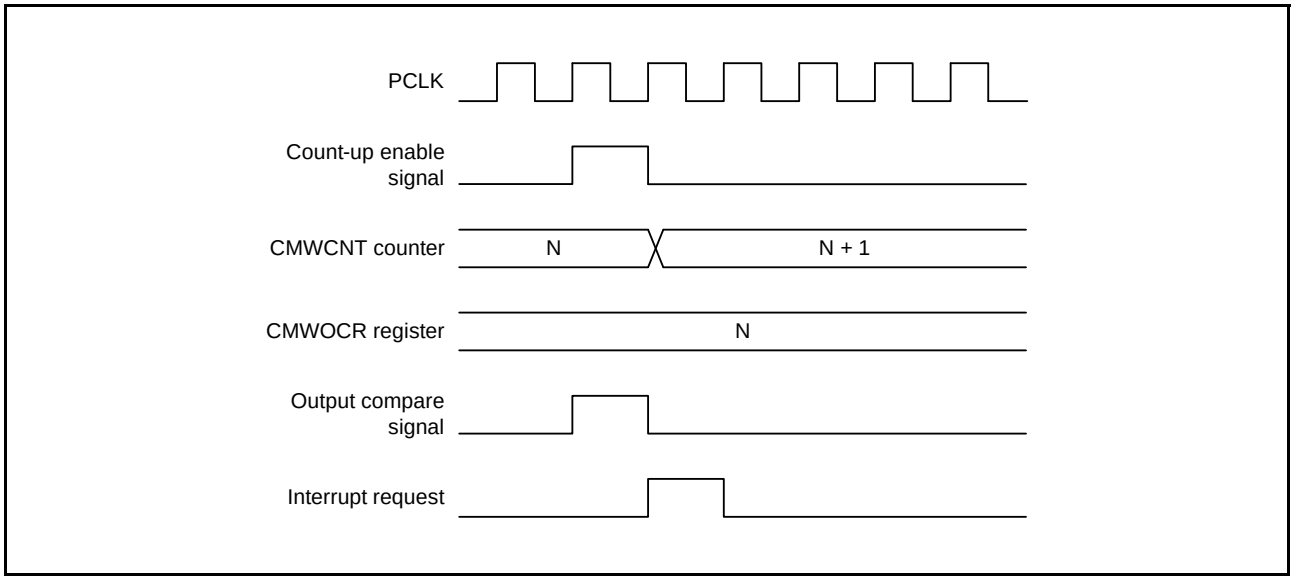


Figure 31.14 Timing of Output Compare Interrupt Generation

(b) Timing of Input Capture Interrupt Generation

Figure 31.15 shows the timing of input capture interrupt generation.

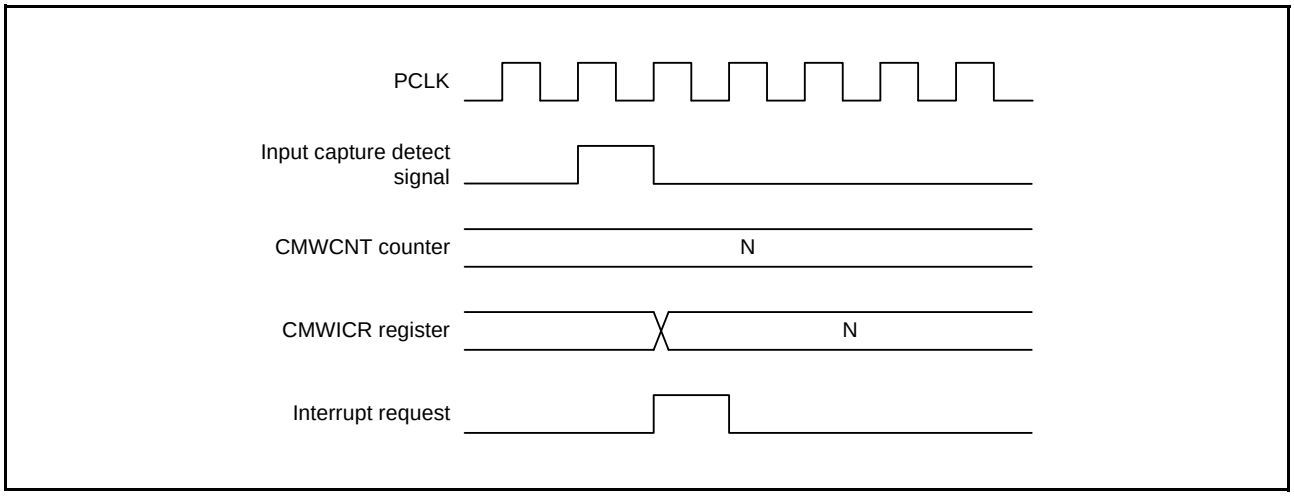


Figure 31.15 Timing of Input Capture Interrupt Generation

31.5 Link Operations by ELC

31.5.1 Event Signal Output to ELC

The CMTW uses the event link controller (ELC) to perform link operation to a preset module using the interrupt request signal as the event signal.
The CMTW outputs the event signal upon a compare match. The corresponding channel is channel 0. The event signal can be output regardless of the corresponding interrupt request enable bit (CMWCR.CMWIE).
For details, refer to section 21, Event Link Controller (ELC).

(1) Compare Match Event

In response to a compare match, the CMTW simultaneously issues an interrupt request and a compare match event signal to the ELC. The event signal is issued regardless of the settings of the corresponding interrupt request enable bit (CMWCR.CMWIE bit).

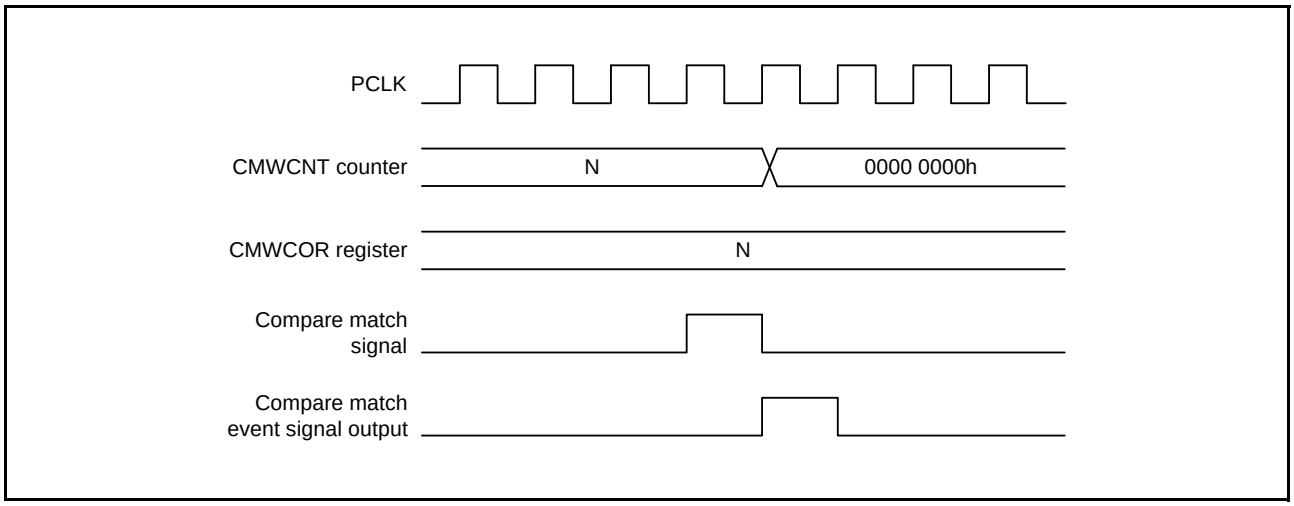


Figure 31.16 Timing of Issuing a Compare Match Event Signal

31.5.2 CMTW Operation When Receiving an Event Signal from ELC

The CMTW can perform any of the following operations according to the event preset by the ELSRn register of the ELC.

(1) Count Start

The CMTW count start operation is selected by the ELOPH register of the ELC. If the event specified by the ELSRn register occurs, the CMWSTR.STR bit is set to 1, starting the CMTW count operation.
However, if the specified event occurs while the CMWSTR.STR bit is 1, the event is ignored.

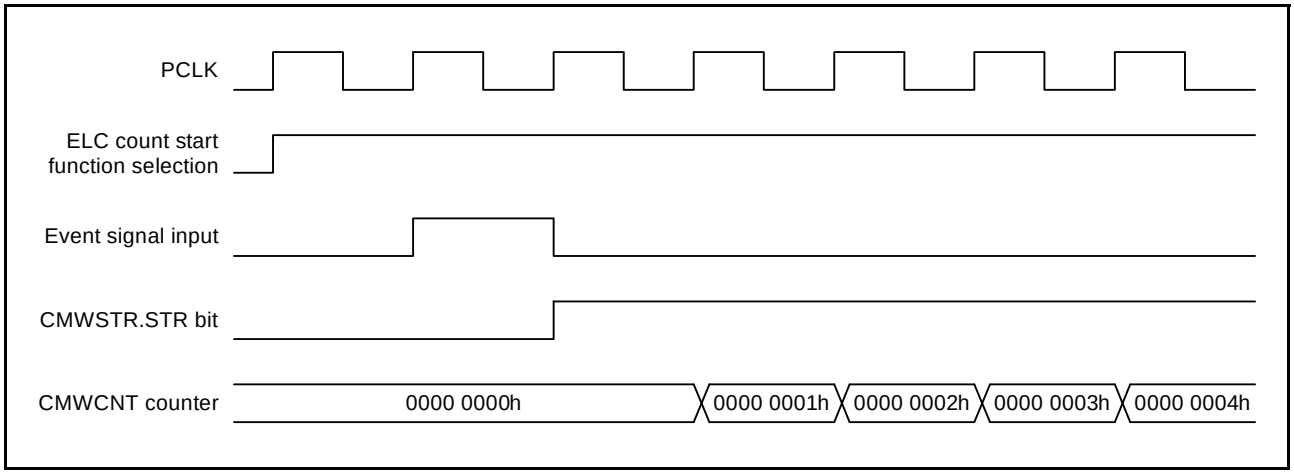


Figure 31.17 Count Start Operation on Acceptance of the Event Signal

(2) Event Count

The CMTW event count operation is selected by the ELOPH register of the ELC. If the event specified by the ELSRn register occurs when the CMWCR.STR bit is 1, the event is counted as the count source regardless of the CMWCR.CKS[1:0] bit setting.
Reading the counter value returns the number of events that have been actually input.

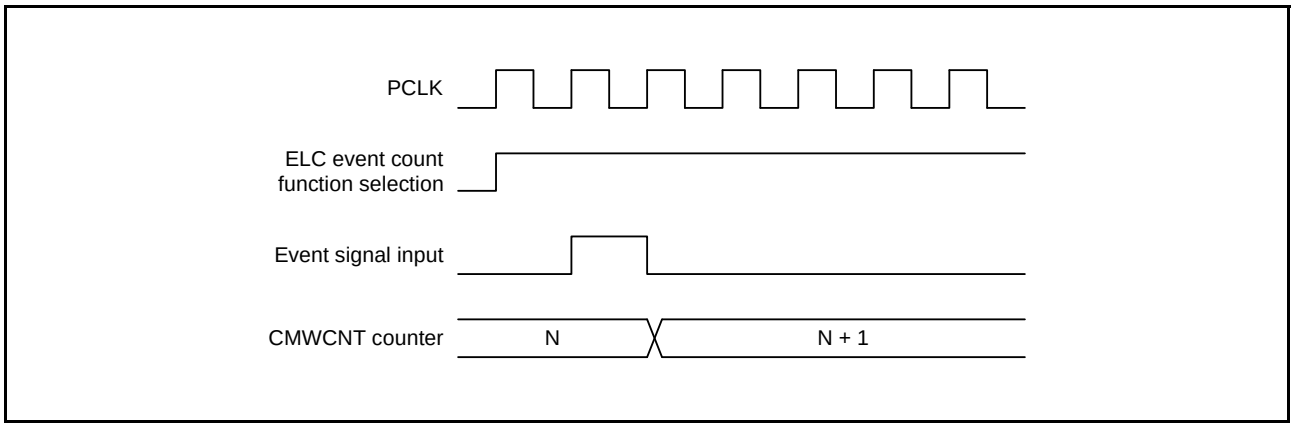


Figure 31.18 Event Count Operation on Acceptance of the Event Signal

(3) Count Restart

The CMTW count restart operation is selected by the ELOPH register of the ELC. If the event specified by the ELSRn register occurs, the CMWCNT counter value is set to 0. If the CMWSTR.STR bit is 1, the count operation can be continued.

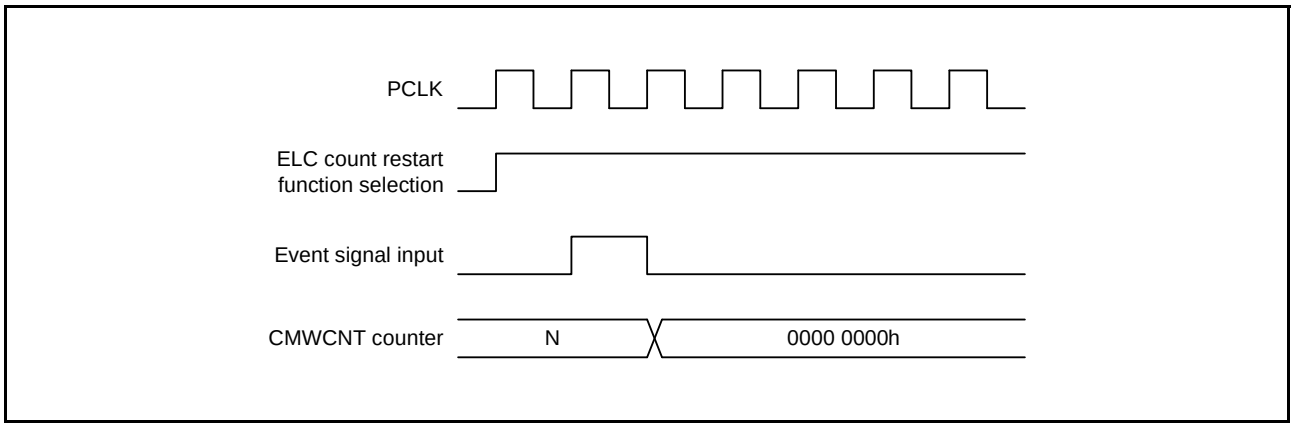


Figure 31.19 Count Restart Operation on Acceptance of the Event Signal

31.5.3 Conflict between Event Link Operation and Register Access

The followings are the notes on using CMTW for event link operations.
Table 31.4 lists count operations when event link operation and register access conflict.

Table 31.4 Count Operations When Event Link Operation and Register Access Conflict

Event Link Operation	Register Access	CMWCNT Counter Status	Operation to be Performed
Count start	Writing to the CMWSTR.STR bit	Stopped state	Count start
		Compare match	Count start
		Counting up	Count start
Event count	Writing to CMWCNT counter	—	Event count
	Writing to CMWCOR register	Compare match	Compare match
Count restart	Writing to CMWCNT counter	Other than compare match	Count restart
	Writing to CMWCNT counter	Compare match	Compare match
	(No access to registers)	Compare match	Compare match
(No events)	Writing to CMWCNT counter	Compare match	Output of compare match interrupt request Writing to CMWCNT counter
		Counting up	Writing to CMWCNT counter
	Writing to CMWCOR register	Compare match	Compare match
	Writing to CMWOCR register	Compare match	Compare match
	Reading from CMWCNT counter	Counting up	Counting up and reading of the previous value

(1) Count Start

When writing to the STR bit in the CMWSTR register and acceptance of the event signal are in contention, writing to the STR bit does not proceed since setting of the STR bit to 1 in response to the event takes priority.

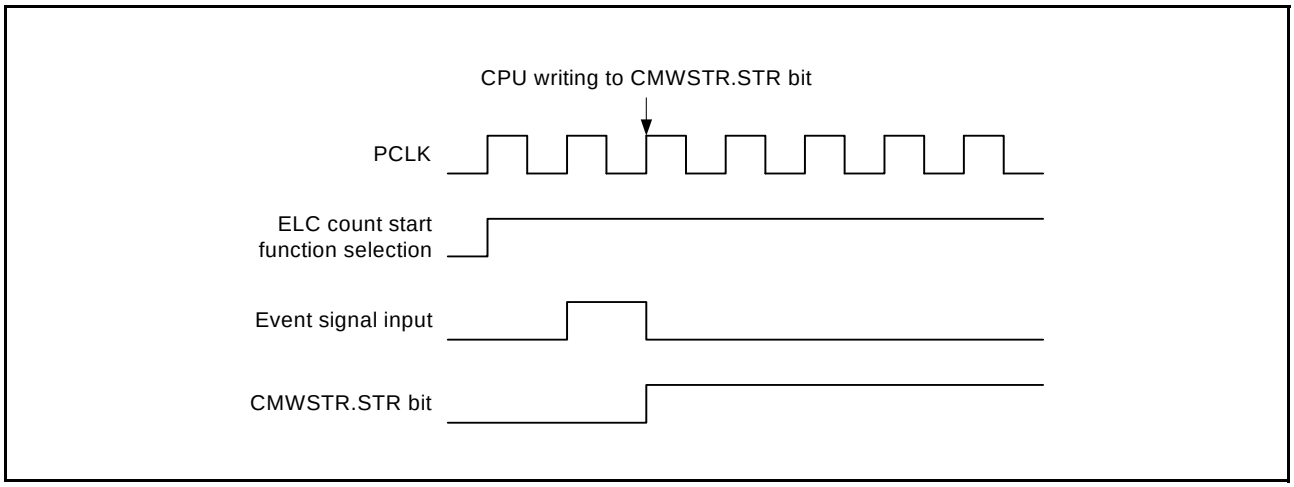


Figure 31.20 Conflict between Event Acceptance and Register Access in Count Start Operation

(2) Event Count

When writing to the CMWCNT counter and acceptance of the event signal are in contention, writing to the CMWCNT counter does not proceed since the count operation in response to the event takes priority.

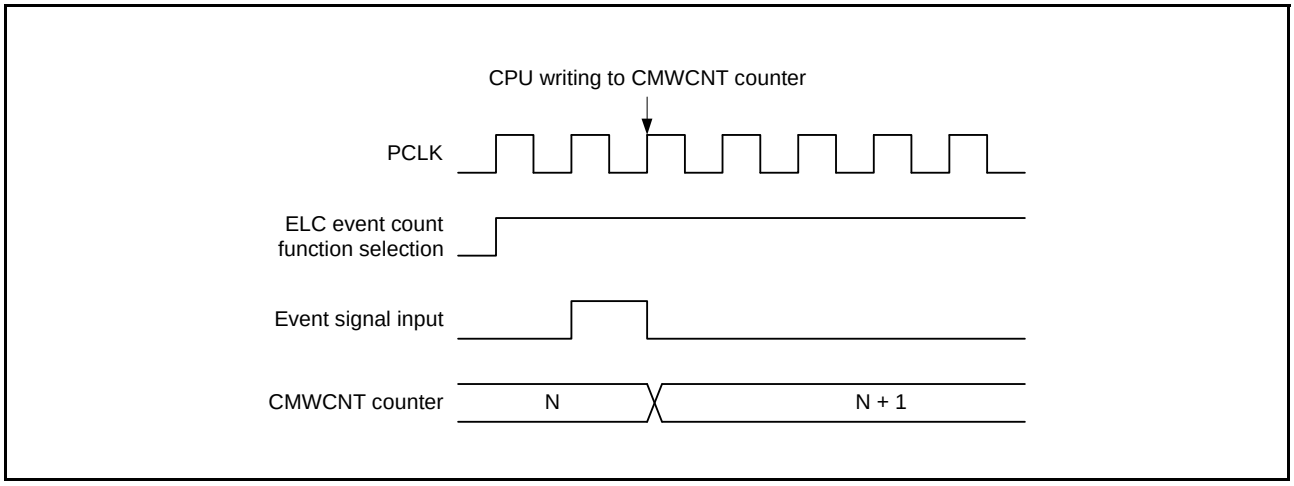


Figure 31.21 Conflict between Event Acceptance and Register Access in Event Counting Operation

(3) Count Restart

When writing to the CMWCNT counter and acceptance of the event signal are in contention, writing to the CMWCNT counter does not proceed since the counter value initialization in response to the event occurrence takes priority.

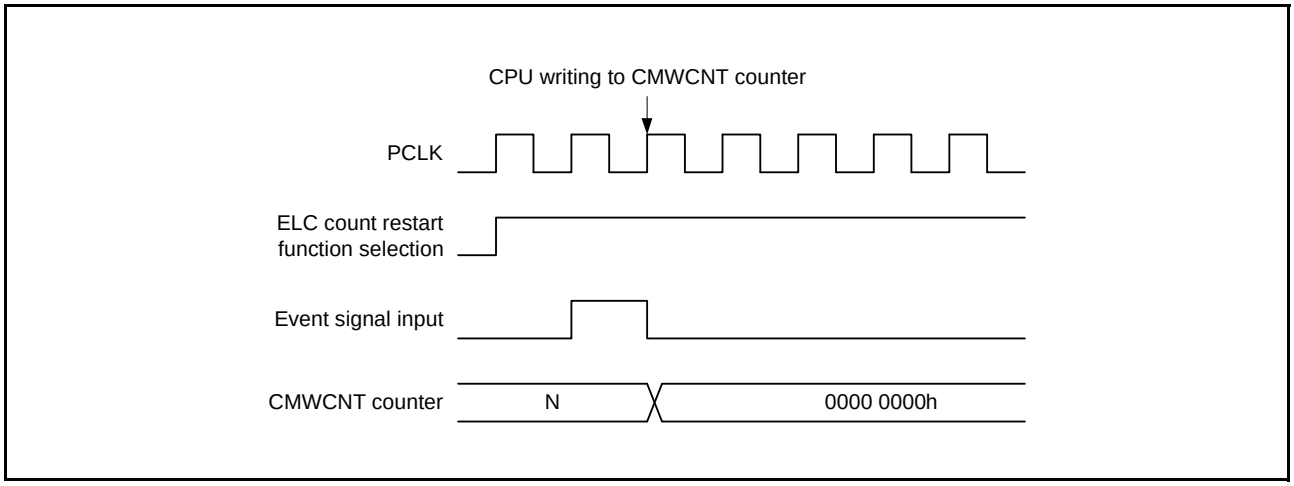


Figure 31.22 Conflict between Event Acceptance and Register Access in Count Restart Operation

31.6 Usage Notes

31.6.1 Setting the Module Stop Function

The CMTW operation can be enabled and disabled using the MSTPCRA register. With the initial value, the CMTW is stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

31.6.2 Conflict between CMWCNT Counter Writing and Compare Match

If the compare match signal is generated during writing to the CMWCNT counter, the compare match request is output but the counter is not cleared since writing to the counter takes priority.
Figure 31.23 shows the timing of conflict between CMWCNT counter writing and compare match.

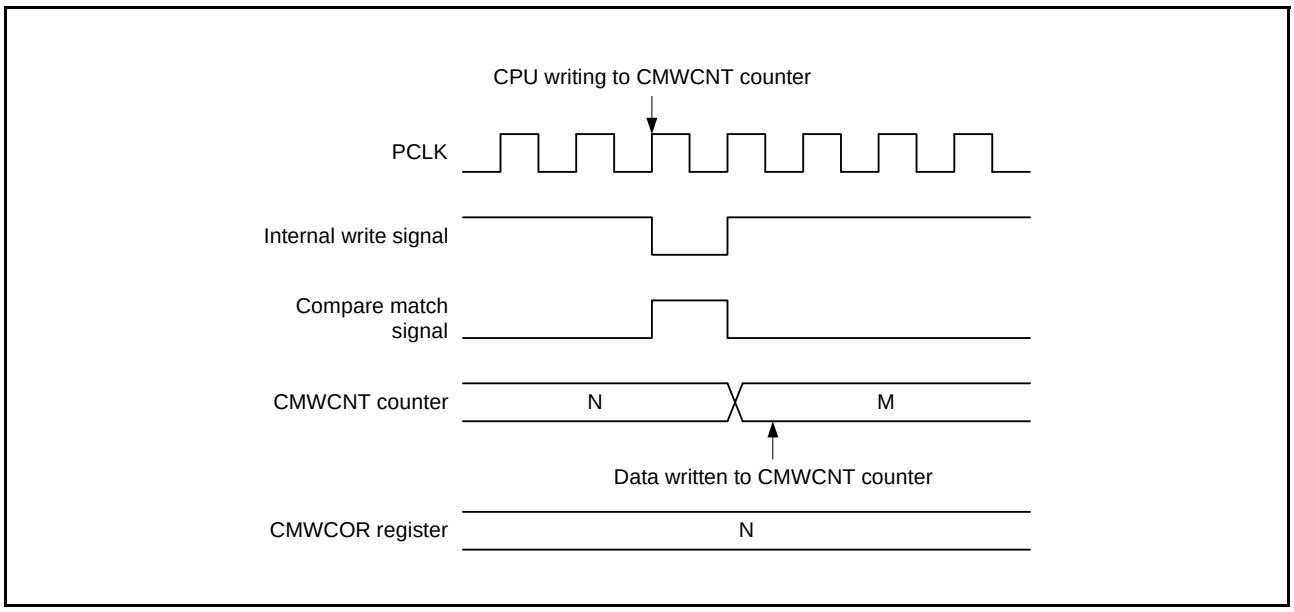


Figure 31.23 Conflict between CMWCNT Counter Writing and Compare Match

31.6.3 Conflict between CMWCNT Counter Writing and Incrementing or Clearing

In case of conflict between incrementation or clearing of the CMWCNT counter and writing to the CMWCNT counter, the counter is not actually incremented or cleared since writing to the CMWCNT counter takes priority.
Figure 31.24 shows the timing in the case of contention between writing to the CMWCNT counter and incrementation or clearing.

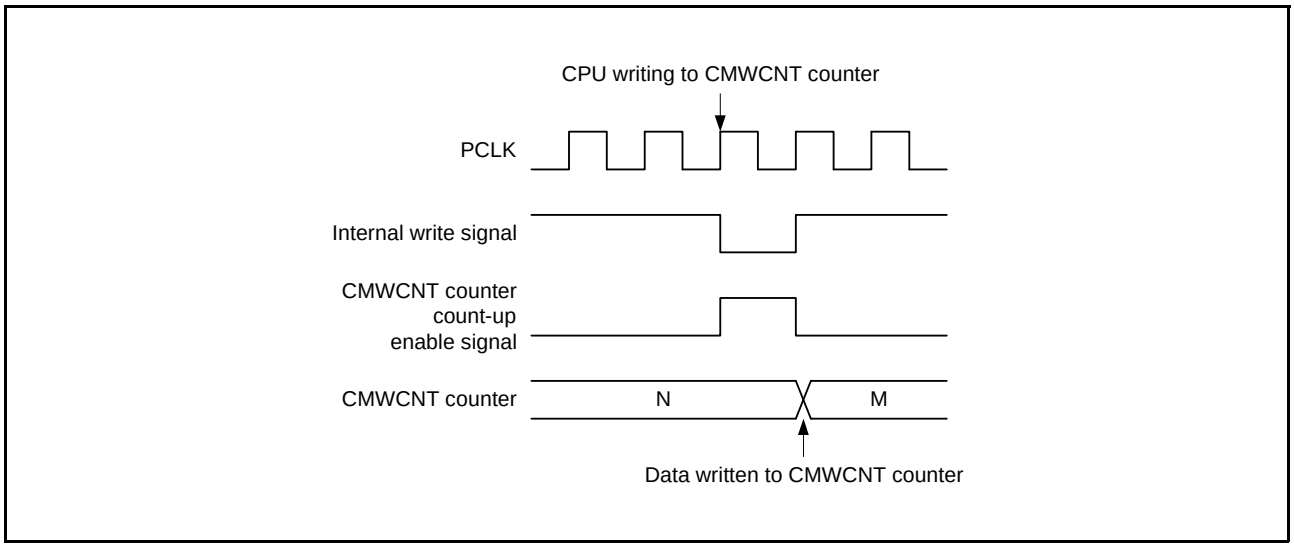


Figure 31.24 Conflict between CMWCNT Counter Writing and Incrementing

31.6.4 Conflict between CMWCOR Register Writing and Compare Match

If the compare match is generated during the CMWCOR register write cycle, the writing to the CMWCOR register takes priority and also the compare match signal is output.
Figure 31.25 shows the timing of conflict between CMWCOR register writing and compare match.

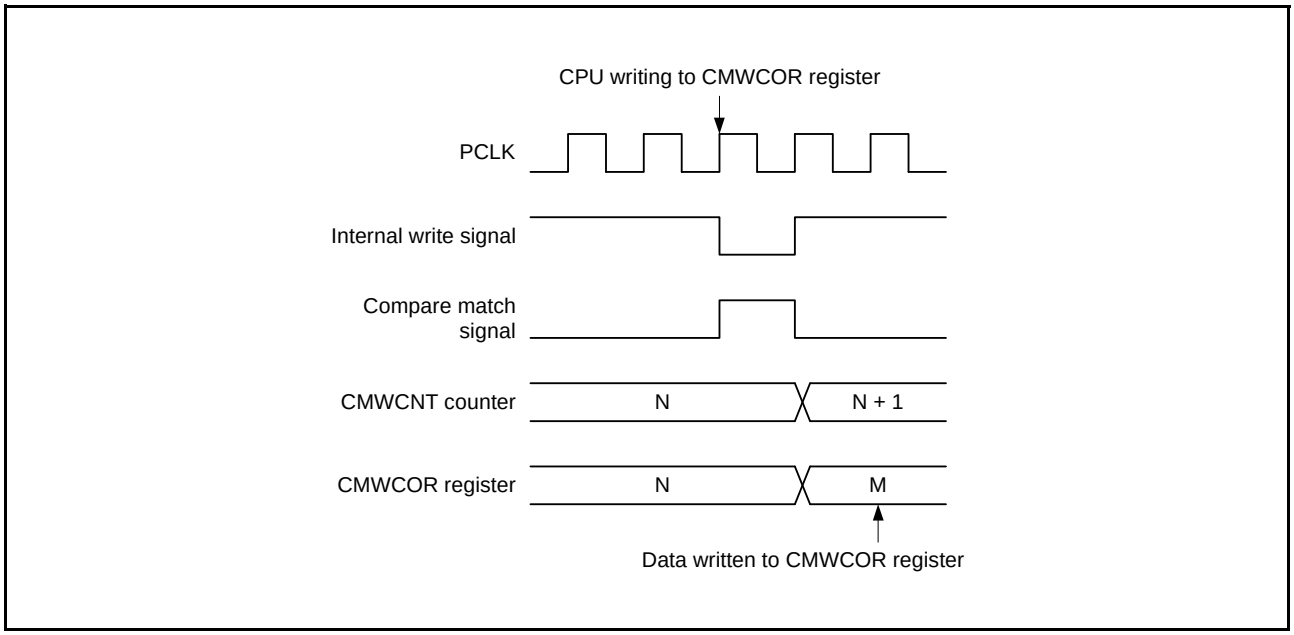


Figure 31.25 Conflict between CMWCOR Register Writing and Compare Match

31.6.5 Conflict between CMWOCR Register Writing and Compare Match

If the compare match is generated during the CMWOCR register write cycle, the writing to the CMWOCR register takes priority and also the compare match signal is output.
Figure 31.26 shows the timing of conflict between CMWOCR register writing and compare match.

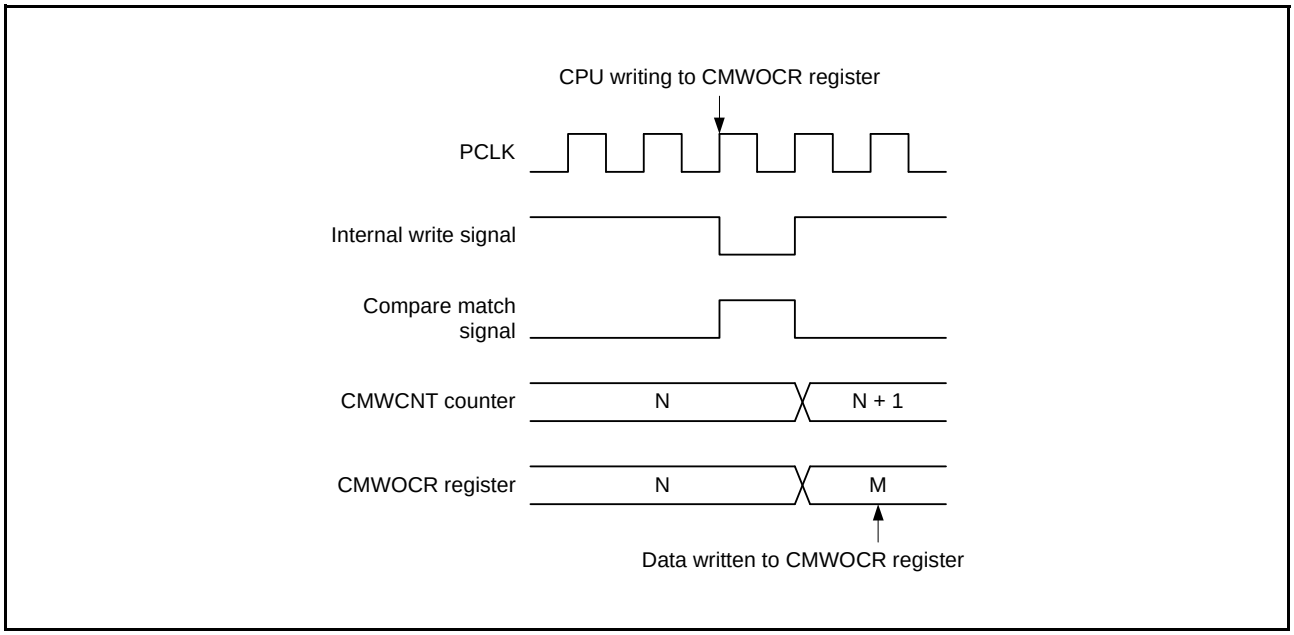


Figure 31.26 Conflict between CMWOCR Register Writing and Compare Match

31.6.6 Conflict between CMWCNT Counter Reading and Incrementing or Clearing

If the CMWCNT counter incrementing or clearing process occurs at the same time that the data of the CMWCNT counter is read, the value having been in the CMWCNT counter before incremented or cleared is read.
Figure 31.27 shows the timing of conflict between CMWCNT counter reading and incrementing.

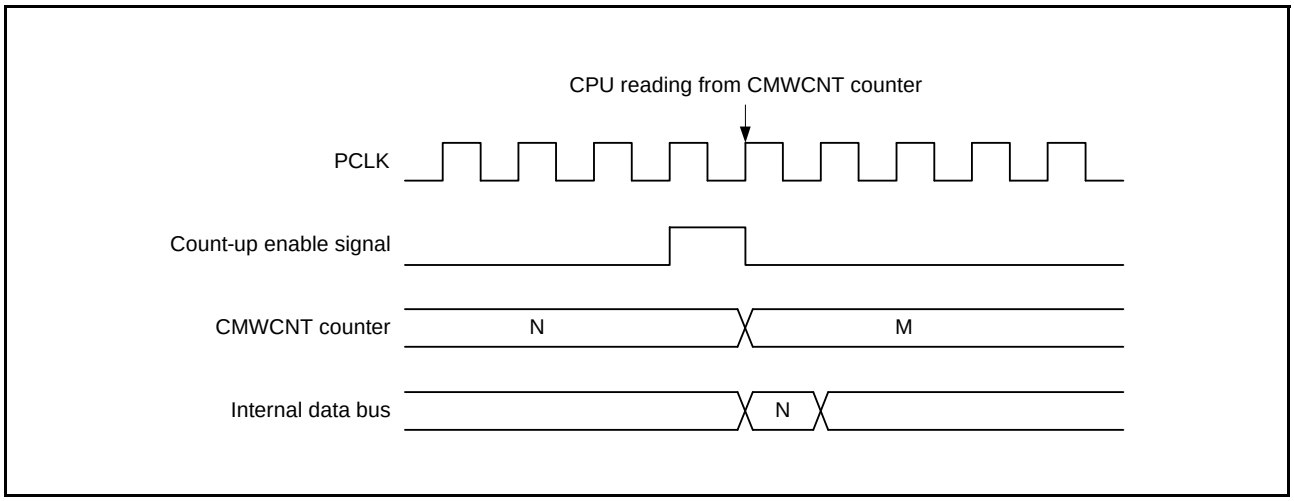


Figure 31.27 Conflict between CMWCNT Counter Reading and Incrementing

31.6.7 Conflict between CMWICR Register Reading and Input Capture

If the input capture detection signal is generated at the same time that the data of the CMWICR register is read, the value having been in the CMWICR register before updated by input capture transfer is read.
Figure 31.28 shows the timing of conflict between CMWICR register reading and input capture.

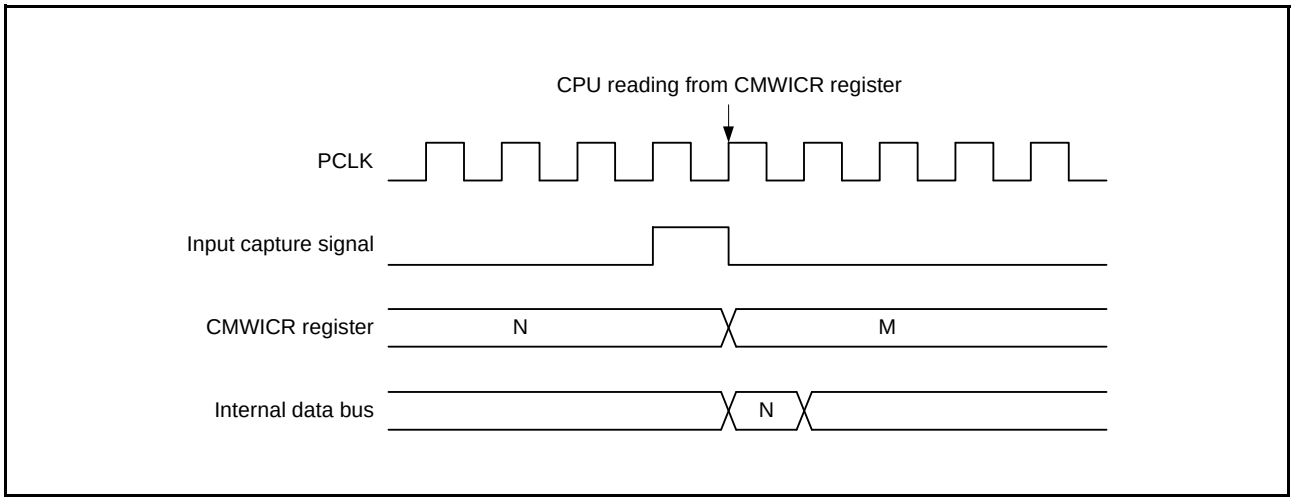


Figure 31.28 Conflict between CMWICR Register Reading and Input Capture

32. Realtime Clock (RTCd)

In this section, “PCLK” is used to refer to PCLKB.

32.1 Overview

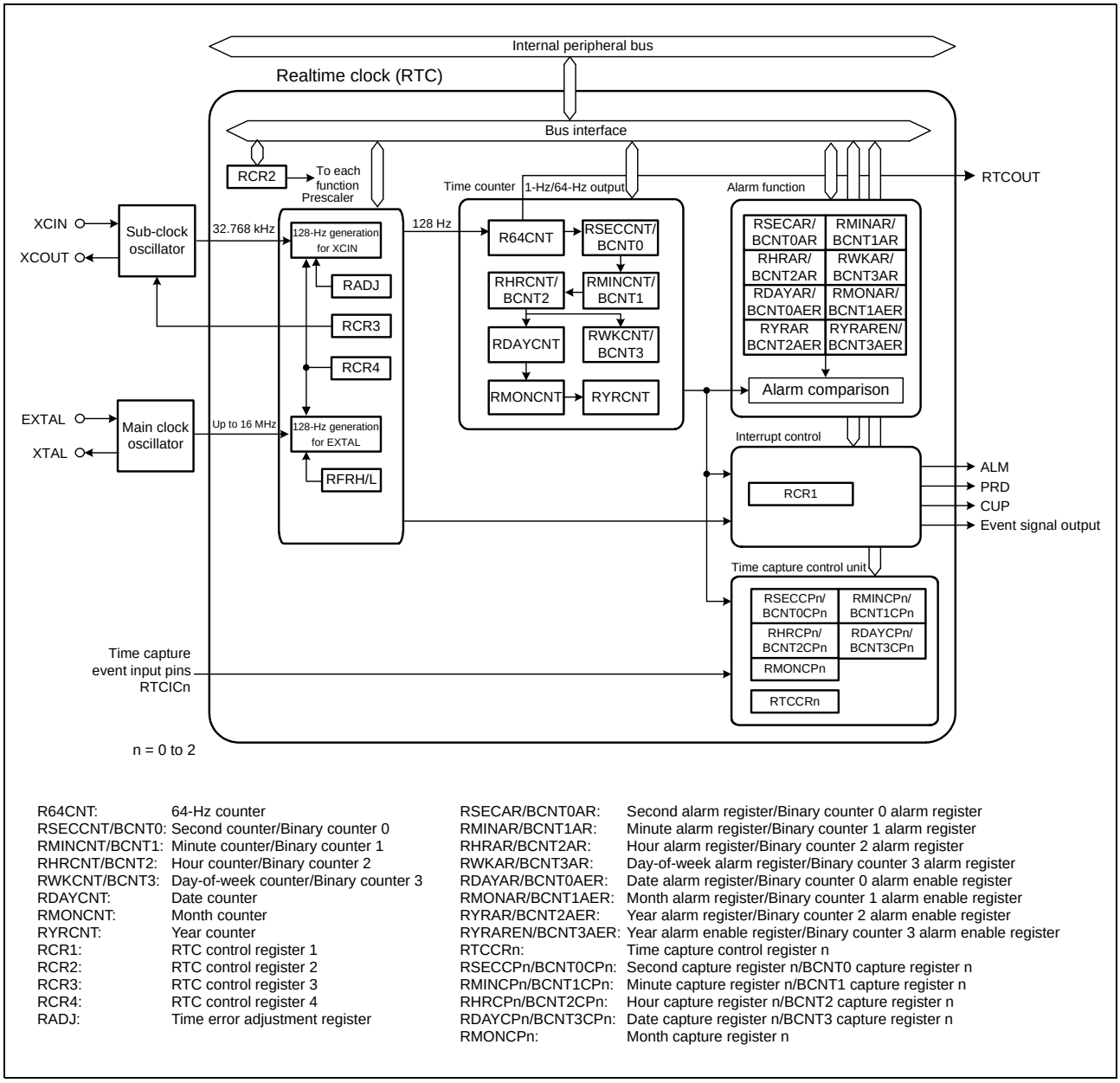
The RTC has two types of counting modes: calendar count mode and binary count mode. They are used by switching the register settings.
For calendar count mode, the RTC has a 100 year calendar from 2000 to 2099 and automatically adjusts dates for leap years.
For binary count mode, the RTC does not count in terms of years, months, dates, day-of-week, hours, or minutes; it counts seconds, and retains the information as a serial value. This mode can be used for calendars other than the Gregorian calendar.
The count source of the time counters is selectable as the sub-clock or main clock.
The RTC uses the 128-Hz clock which is acquired by the count source divided by the prescaler as the reference clock.
Year, month, date, day-of-week, a.m./p.m. (in 12-hour mode), hour, minute, second, or 32-bit binary is counted in 1/128 second units.

Table 32.1 lists the specifications of the RTC, Figure 32.1 shows a block diagram of the RTC, and Table 32.2 shows the pin configuration of the RTC.

Table 32.1 RTC Specifications

Item	Description
Count mode	Calendar count mode/binary count mode
Count source*1	Sub-clock (XCIN) or main clock (EXTAL)
Clock and calendar functions	• Calendar count mode Year, month, date, day-of-week, hour, minute, second are counted, BCD display 12 hours/24 hours mode switching function 30 seconds adjustment function (a number less than 30 is rounded down to 00 seconds, and 30 seconds or more are rounded up to one minute) Automatic adjustment function for leap years • Binary count mode Count seconds in 32 bits, binary display • Common to both modes Start/stop function The sub-second digit is displayed in binary units (1 Hz, 2 Hz, 4 Hz, 8 Hz, 16 Hz, 32 Hz, or 64 Hz). Clock error correction function Clock (1 Hz/64 Hz) output
Interrupts	• Alarm interrupt (ALM) As an alarm interrupt condition, selectable which of the below is compared with: - Calendar count mode: Year, month, date, day-of-week, hour, minute, or second can be selected - Binary count mode: Each bit of the 32-bit binary counter • Periodic interrupt (PRD) 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, or 1/256 second can be selected as an interrupt period. • Carry interrupt (CUP) An interrupt is generated at either of the following timings: - When a carry from the 64-Hz counter to the second counter is generated. - When the 64-Hz counter is changed and the R64CNT register is read at the same time. • Recovery from software standby mode or deep software standby mode can be performed by an alarm interrupt or periodic interrupt
Time capture function	• Times can be captured when the edge of the time capture event input pin is detected. For every event input, month, date, hour, minute, and second are captured or 32-bit binary counter value is captured.
Event link function	Periodic event output

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ the frequency of the count source.



R64CNT: 64-Hz counter
RSECCNT/BCNT0: Second counter/Binary counter 0
RMINCNT/BCNT1: Minute counter/Binary counter 1
RHRCNT/BCNT2: Hour counter/Binary counter 2
RWKCNT/BCNT3: Day-of-week counter/Binary counter 3
RDAYCNT: Date counter
RMONCNT: Month counter
RYRCNT: Year counter
RCR1: RTC control register 1
RCR2: RTC control register 2
RCR3: RTC control register 3
RCR4: RTC control register 4
RADJ: Time error adjustment register

RSECAR/BCNT0AR: Second alarm register/Binary counter 0 alarm register
RMINAR/BCNT1AR: Minute alarm register/Binary counter 1 alarm register
RHRAR/BCNT2AR: Hour alarm register/Binary counter 2 alarm register
RWKAR/BCNT3AR: Day-of-week alarm register/Binary counter 3 alarm register
RDAYAR/BCNT0AER: Date alarm register/Binary counter 0 alarm enable register
RMONAR/BCNT1AER: Month alarm register/Binary counter 1 alarm enable register
RYRAR/BCNT2AER: Year alarm register/Binary counter 2 alarm enable register
RYRAREN/BCNT3AER: Year alarm enable register/Binary counter 3 alarm enable register
RTCCRn: Time capture control register n
RSECCPn/BCNT0CPn: Second capture register n/BCNT0 capture register n
RMINCPn/BCNT1CPn: Minute capture register n/BCNT1 capture register n
RHRCPn/BCNT2CPn: Hour capture register n/BCNT2 capture register n
RDAYCPn/BCNT3CPn: Date capture register n/BCNT3 capture register n
RMONCPn: Month capture register n

Figure 32.1 Block Diagram of RTC

Table 32.2 Pin Configuration of RTC

Pin Name	I/O	Function
XTAL	Output	These pins are used to connect a crystal. The EXTAL pin can also be used to input an external clock. For details, refer to section 9.3.2, External Clock Input.
EXTAL	Input	
XCIN	Input	Connect a 32.768-kHz crystal to these pins.
XCOUT	Output	
RTCOUT	Output	This pin is used to output a 1-Hz/64-Hz waveform, but not in deep software standby mode.
RTCIC0	Input	Time capture event input pins
RTCIC1	Input	
RTCIC2	Input	

32.2 Register Descriptions

When writing to or reading from RTC registers, do so in accordance with section 32.6.5, Notes When Writing to and Reading from Registers.

If the value in an RTC register after a reset is given as x (undefined bits) in the list, it is not initialized by a reset. When RTC enters the reset state or a low power consumption state during counting operations (i.e. while the RCR2.START bit is 1), the year, month, day of the week, date, hours, minutes, seconds, and 64-Hz counters continue to operate. However, when the count source is the main clock, the main clock is not stopped by a reset other than a deep software standby reset. For details, refer to Table 6.2, Targets to be Initialized by Each Reset Source. Note that a reset generated during writing to or updating of a register might destroy the register value. In addition, do not allow the chip to enter software standby mode or deep software standby mode immediately after setting any of these registers. For details, refer to section 32.6.4, Transitions to Low Power Consumption Modes after Setting Registers.

32.2.1 64-Hz Counter (R64CNT)

Address(es): RTC.R64CNT 0008 C400h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	F1HZ	F2HZ	F4HZ	F8HZ	F16HZ	F32HZ	F64HZ
Value after reset:	0	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	F64HZ	64 Hz	Indicate the state between 1 Hz and 64 Hz of the sub-second digit.	R
b1	F32HZ	32 Hz		R
b2	F16HZ	16 Hz		R
b3	F8HZ	8 Hz		R
b4	F4HZ	4 Hz		R
b5	F2HZ	2 Hz		R
b6	F1HZ	1 Hz		R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

The R64CNT counter is used in both calendar count mode and in binary count mode.

The 64-Hz counter (R64CNT) generates a period of one second by counting the 128-Hz reference clock.

The state in the sub-second range can be confirmed by reading this counter.

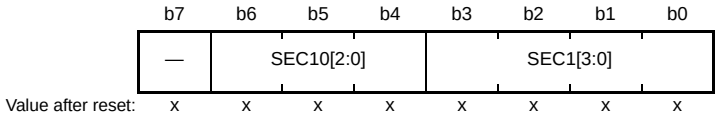
This counter is set to 00h by an RTC software reset or executing 30-second adjustment.

To read this counter, follow the procedure in section 32.3.5, Reading 64-Hz Counter and Time.

32.2.2 Second Counter (RSECCNT)/Binary Counter 0 (BCNT0)

(1) In calendar count mode:

Address(es): RTC.RSECCNT 0008 C402h



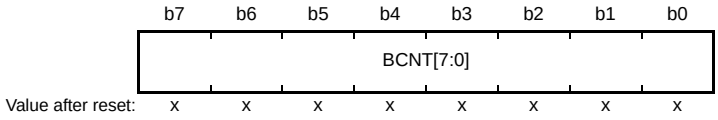
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	1-Second Count	Counts from 0 to 9 every second. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	SEC10[2:0]	10-Second Count	Counts from 0 to 5 for 60-second counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

The RSECCNT counter is used for setting and counting the BCD-coded second value. It counts carries generated once per second in the 64-Hz counter.
The setting range is decimal 00 to 59. The RTC will not operate normally if any other value is set. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
After writing to the RSECCNT register, confirm that its value has actually changed before proceeding with further processing. Refer to section 32.6.5, Notes When Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT0 0008 C402h



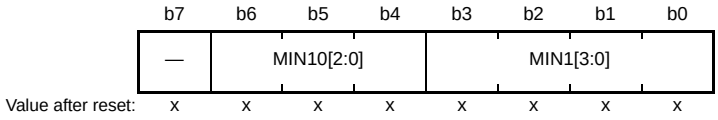
x: Undefined

The BCNT0 counter is a readable/writable 32-bit binary counter b7 to b0.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 32.3.5, Reading 64-Hz Counter and Time.

32.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1)

(1) In calendar count mode:

Address(es): RTC.RMINCNT 0008 C404h



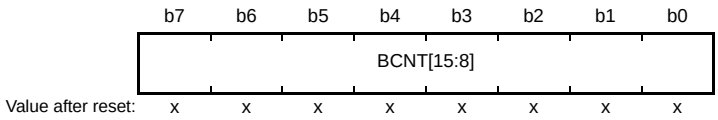
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	1-Minute Count	Counts from 0 to 9 every minute. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	MIN10[2:0]	10-Minute Count	Counts from 0 to 5 for 60-minute counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

The RMINCNT counter is used for setting and counting the BCD-coded minute value. It counts carries generated once per minute in the second counter.
A value from 00 through 59 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
After writing to the RMINCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 32.6.5, Notes When Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT1 0008 C404h



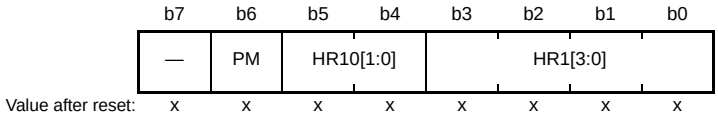
x: Undefined

The BCNT1 counter is a readable/writable 32-bit binary counter b15 to b8.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 32.3.5, Reading 64-Hz Counter and Time.

32.2.4 Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2)

(1) In calendar count mode:

Address(es): RTC.RHRCNT 0008 C406h



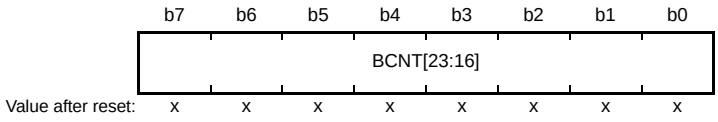
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	1-Hour Count	Counts from 0 to 9 once per hour. When a carry is generated, 1 is added to the tens place.	R/W
b5, b4	HR10[1:0]	10-Hour Count	Counts from 0 to 2 once per carry from the ones place.	R/W
b6	PM	PM	Time Counter Setting for a.m./p.m. 0: a.m. 1: p.m.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

The RHRCNT counter is used for setting and counting the BCD-coded hour value. It counts carries generated once per hour in the minute counter.
The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24).
When the RCR2.HR24 bit is 0: From 00 to 11 (in BCD)
When the RCR2.HR24 bit is 1: From 00 to 23 (in BCD)
If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
The PM bit is only enabled when the RCR2.HR24 bit is 0. Otherwise, the setting in the PM bit has no effect.
After writing to the RHRCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 32.6.5, Notes When Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT2 0008 C406h



x: Undefined

The BCNT2 counter is a readable/writable 32-bit binary counter b23 to b16.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 32.3.5, Reading 64-Hz Counter and Time.

32.2.5 Day-of-Week Counter (RWKCNT)/Binary Counter 3 (BCNT3)

(1) In calendar count mode:

Address(es): RTC.RWKCNT 0008 C408h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	DAYW[2:0]		
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DAYW[2:0]	Day-of-Week Counting	b2 b0 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting Prohibited	R/W
b7 to b3	—	Reserved	Set these bits to 0. They are read as the set value.	R/W

The RWKCNT counter is used for setting and counting in the coded day-of-week value. It counts carries generated once per day in the hour counter.
A value from 0 through 6 can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
Refer to section 32.6.5, Notes When Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT3 0008 C408h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNT[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT3 counter is a readable/writable 32-bit binary counter b31 to b24.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 32.3.5, Reading 64-Hz Counter and Time.

32.2.6 Date Counter (RDAYCNT)

Address(es): RTC.RDAYCNT 0008 C40Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	0	0	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DATE1[3:0]	1-Day Count	Counts from 0 to 9 once per day. When a carry is generated, 1 is added to the tens place.	R/W
b5, b4	DATE10[1:0]	10-Day Count	Counts from 0 to 3 once per carry from the ones place.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RDAYCNT counter is used in calendar count mode.
The RDAYCNT counter is used for setting and counting the BCD-coded date value. It counts carries generated once per day in the hour counter. The count operation depends on the month and whether the year is a leap year.
Leap years are determined according to whether the year counter (RYRCNT) value is divisible by 400, 100, and 4.
A value from 01 through 31 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. (When specifying a value, note that the range of specifiable days depends on the month and whether the year is a leap year.) Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
After writing to the RHRCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 32.6.5, Notes When Writing to and Reading from Registers for notes on accessing registers.

32.2.7 Month Counter (RMONCNT)

Address(es): RTC.RMONCNT 0008 C40Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MON10	MON1[3:0]			
Value after reset:	0	0	0	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	1-Month Count	Counts from 0 to 9 once per month. When a carry is generated, 1 is added to the tens place.	R/W
b4	MON10	10-Month Count	Counts from 0 to 1 once per carry from the ones place.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RMONCNT counter is used in calendar count mode.
The RMONCNT counter is used for setting and counting the BCD-coded month value. It counts carries generated once per month in the date counter.
A value from 01 through 12 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
After writing to the RMONCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 32.6.5, Notes When Writing to and Reading from Registers for notes on accessing registers.

32.2.8 Year Counter (RYRCNT)

Address(es): RTC.RYRCNT 0008 C40Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	YR1[3:0]	1-Year Count	Counts from 0 to 9 once per year. When a carry is generated, 1 is added to the tens place.	R/W
b7 to b4	YR10[3:0]	10-Year Count	Counts from 0 to 9 once per carry from ones place. When a carry is generated in the tens place, 1 is added to the hundreds place.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RYRCNT counter is used in calendar count mode.

The RYRCNT counter is used for setting and counting the BCD-coded year value. It counts carries generated once per year in the month counter.

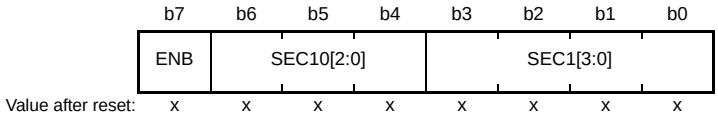
A value from 00 through 99 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

After writing to the RYRCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 32.6.5, Notes When Writing to and Reading from Registers for notes on accessing registers.

32.2.9 Second Alarm Register (RSECAR)/Binary Counter 0 Alarm Register (BCNT0AR)

(1) In calendar count mode:

Address(es): RTC.RSECAR 0008 C410h



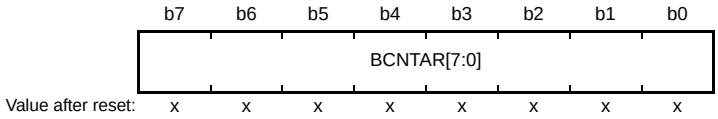
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	1 Second	Value for the ones place of seconds	R/W
b6 to b4	SEC10[2:0]	10 Seconds	Value for the tens place of seconds	R/W
b7	ENB	ENB	0: The register value is not compared with the RSECCNT counter value. 1: The register value is compared with the RSECCNT counter value.	R/W

The RSECAR register is an alarm register corresponding to the BCD-coded second counter RSECCNT. When the ENB bit is set to 1, the RSECAR value is compared with the RSECCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
RSECAR values from 00 through 59 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
After writing to the RSECAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 32.6.5, Notes When Writing to and Reading from Registers for notes on accessing registers.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT0AR 0008 C410h



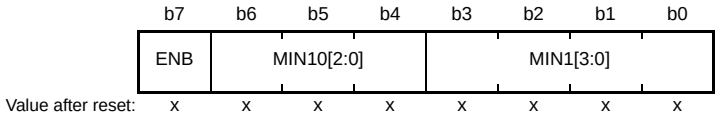
x: Undefined

The BCNT0AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b7 to b0.
This register is set to 00h by an RTC software reset.

32.2.10 Minute Alarm Register (RMINAR)/Binary Counter 1 Alarm Register (BCNT1AR)

(1) In calendar count mode:

Address(es): RTC.RMINAR 0008 C412h



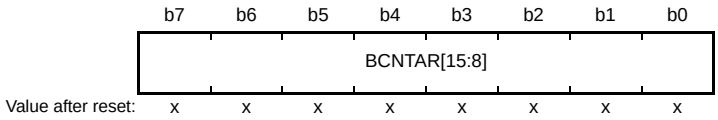
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	1 Minute	Value for the ones place of minutes	R/W
b6 to b4	MIN10[2:0]	10 Minutes	Value for the tens place of minutes	R/W
b7	ENB	ENB	0: The register value is not compared with the RMINCNT counter value. 1: The register value is compared with the RMINCNT counter value.	R/W

The RMINAR register is an alarm register corresponding to the BCD-coded minute counter RMINCNT. When the ENB bit is set to 1, the RMINAR value is compared with the RMINCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
RMINAR values from 00 through 59 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
After writing to the RMINAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 32.6.5, Notes When Writing to and Reading from Registers for notes on accessing registers.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT1AR 0008 C412h



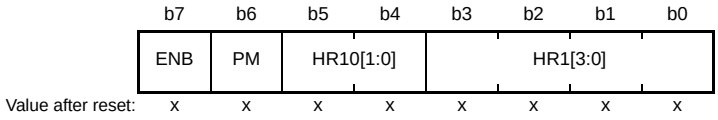
x: Undefined

The BCNT1AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b15 to b8.
This register is set to 00h by an RTC software reset.

32.2.11 Hour Alarm Register (RHRAR)/Binary Counter 2 Alarm Register (BCNT2AR)

(1) In calendar count mode:

Address(es): RTC.RHRAR 0008 C414h



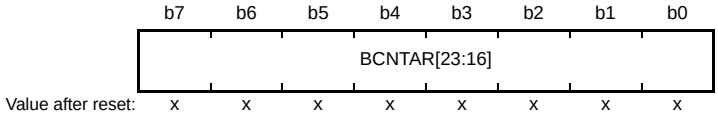
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	1 Hour	Value for the ones place of hours	R/W
b5, b4	HR10[1:0]	10 Hours	Value for the tens place of hours	R/W
b6	PM	PM	Time Alarm Setting for a.m./p.m. 0: a.m. 1: p.m.	R/W
b7	ENB	ENB	0: The register value is not compared with the RHRCNT counter value. 1: The register value is compared with the RHRCNT counter value.	R/W

The RHRAR register is an alarm register corresponding to the BCD-coded hour counter RHRCNT. When the ENB bit is set to 1, the RHRAR value is compared with the RHRCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24).
When the RCR2.HR24 bit is 0: From 00 to 11 (in BCD)
When the RCR2.HR24 bit is 1: From 00 to 23 (in BCD)
If a value outside of this range is specified, the RTC does not operate correctly.
When the RCR2.HR24 bit is 0, be sure to set the PM bit.
When the RCR2.HR24 bit is 1, the setting in the PM bit has no effect.
After writing to the RHRAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 32.6.5, Notes When Writing to and Reading from Registers for notes on accessing registers.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT2AR 0008 C414h



x: Undefined

The BCNT2AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b23 to b16.
This register is set to 00h by an RTC software reset.

32.2.12 Day-of-Week Alarm Register (RWKAR)/Binary Counter 3 Alarm Register (BCNT3AR)

(1) In calendar count mode:

Address(es): RTC.RWKAR 0008 C416h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	—	—	DAYW[2:0]		
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DAYW[2:0]	Day-of-Week Setting	b2 b0 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting Prohibited	R/W
b6 to b3	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RWKCNT counter value. 1: The register value is compared with the RWKCNT counter value.	R/W

The RWKAR register is an alarm register corresponding to the coded day-of-week counter RWKCNT. When the ENB bit is set to 1, the RWKAR value is compared with the RWKCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.

RWKAR values from 0 through 6 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

After writing to the RWKAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 32.6.5, Notes When Writing to and Reading from Registers for notes on accessing registers.

This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT3AR 0008 C416h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT3AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b31 to b24.

This register is set to 00h by an RTC software reset.

32.2.13 Date Alarm Register (RDAYAR)/Binary Counter 0 Alarm Enable Register (BCNT0AER)

(1) In calendar count mode:

Address(es): RTC.RDAYAR 0008 C418h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DATE1[3:0]	1 Day	Value for the ones place of days	R/W
b5, b4	DATE10[1:0]	10 Days	Value for the tens place of days	R/W
b6	—	Reserved	Set this bit to 0. It is read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RDAYCNT counter value. 1: The register value is compared with the RDAYCNT counter value.	R/W

The RDAYAR register is an alarm register corresponding to the BCD-coded date counter RDAYCNT. When the ENB bit is set to 1, the RDAYAR value is compared with the RDAYCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
RDAYAR values from 01 through 31 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
After writing to the RDAYAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 32.6.5, Notes When Writing to and Reading from Registers for notes on accessing registers.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT0AER 0008 C418h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT0AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b7 to b0. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.
This register is set to 00h by an RTC software reset.

32.2.14 Month Alarm Register (RMONAR)/Binary Counter 1 Alarm Enable Register (BCNT1AER)

(1) In calendar count mode:

Address(es): RTC.RMONAR 0008 C41Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	MON10	MON1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	1 Month	Value for the ones place of months	R/W
b4	MON10	10 Months	Value for the tens place of months	R/W
b6, b5	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RMONCNT counter value. 1: The register value is compared with the RMONCNT counter value.	R/W

The RMONAR register is an alarm register corresponding to the BCD-coded month counter RMONCNT. When the ENB bit is set to 1, the RMONAR value is compared with the RMONCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
RMONAR values from 01 through 12 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
After writing to the RMONAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 32.6.5, Notes When Writing to and Reading from Registers for notes on accessing registers.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT1AER 0008 C41Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT1AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b15 to b8. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.
This register is set to 00h by an RTC software reset.

32.2.15 Year Alarm Register (RYRAR)/Binary Counter 2 Alarm Enable Register (BCNT2AER)

(1) In calendar count mode:

Address(es): RTC.RYRAR 0008 C41Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	YR1[3:0]	1 Year	Value for the ones place of years	R/W
b7 to b4	YR10[3:0]	10 Years	Value for the tens place of years	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RYRAR register is an alarm register corresponding to the BCD-coded year counter RYRCNT.
RYRAR values from 00 through 99 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
After writing to the RYRAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 32.6.5, Notes When Writing to and Reading from Registers for notes on accessing registers.
This register is set to 0000h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT2AER 0008 C41Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ENB[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

The BCNT2AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b23 to b16. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.
This register is set to 0000h by an RTC software reset.

32.2.16 Year Alarm Enable Register (RYRAREN)/Binary Counter 3 Alarm Enable Register (BCNT3AER)

(1) In calendar count mode:

Address(es): RTC.RYRAREN 0008 C41Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RYRCNT counter value. 1: The register value is compared with the RYRCNT counter value.	R/W

When the ENB bit in the RYRAREN register is set to 1, the RYRAR value is compared with the RYRCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT3AER 0008 C41Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT3AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b31 to b24. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.
This register is set to 00h by an RTC software reset.

32.2.17 RTC Control Register 1 (RCR1)

Address(es): RTC.RCR1 0008 C422h

	b7	b6	b5	b4	b3	b2	b1	b0
	PES[3:0]				RTCOS	PIE	CIE	AIE
Value after reset:	x	x	x	x	0	x	0	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	AIE	Alarm Interrupt Enable	0: An alarm interrupt request is disabled. 1: An alarm interrupt request is enabled.	R/W
b1	CIE	Carry Interrupt Enable	0: A carry interrupt request is disabled. 1: A carry interrupt request is enabled.	R/W
b2	PIE	Periodic Interrupt Enable	0: A periodic interrupt request is disabled. 1: A periodic interrupt request is enabled.	R/W
b3	RTCOS	RTCOUT Output Select	0: RTCOUT outputs 1 Hz. 1: RTCOUT outputs 64 Hz.	R/W
b7 to b4	PES[3:0]	Periodic Interrupt Select	b7 b4 0 1 1 0: A periodic interrupt is generated every 1/256 second.*1 0 1 1 1: A periodic interrupt is generated every 1/128 second. 1 0 0 0: A periodic interrupt is generated every 1/64 second. 1 0 0 1: A periodic interrupt is generated every 1/32 second. 1 0 1 0: A periodic interrupt is generated every 1/16 second. 1 0 1 1: A periodic interrupt is generated every 1/8 second. 1 1 0 0: A periodic interrupt is generated every 1/4 second. 1 1 0 1: A periodic interrupt is generated every 1/2 second. 1 1 1 0: A periodic interrupt is generated every 1 second. 1 1 1 1: A periodic interrupt is generated every 2 seconds. Other than above: No periodic interrupts are generated.	R/W

Note 1. When the main clock is selected (RCR4.RCKSEL = 1) while PES[3:0] = 0110b, a periodic interrupt is generated every 1/128 second.

The RCR1 register is used in both calendar count mode and in binary count mode.
Bits AIE, PIE, and PES[3:0] are updated synchronously with the count source. When the RCR1 register is modified, check that all the bits have been updated before proceeding to the next processing.

AIE Bit (Alarm Interrupt Enable)

This bit enables or disables alarm interrupt requests.
If the times indicated by the counters and alarm settings match in deep software standby mode, the MCU returns from the mode regardless of the value of the AIE bit.

CIE Bit (Carry Interrupt Enable)

This bit enables and disables interrupt requests when a carry to the RSECCNT/BCNT0 register occurs, or when a carry to the 64-Hz counter (R64CNT) occurs while reading the 64-Hz counter.

PIE Bit (Periodic Interrupt Enable)

This bit enables or disabled a periodic interrupt.
If the periods indicated by the counters and PES[3:0] settings match in deep software standby mode, the MCU returns from the mode regardless of the value of the PIE bit.

RTCOS Bit (RTCOUT Output Select)

This bit selects the RTCOUT output period. The RTCOS bit must be rewritten while count operation is stopped (the RCR2.START bit is 0) and RTCOUT output is disabled (the RCR2.RTCOE bit is 0). When the RTCOUT is output to an external pin, the RCR2.RTCOE bit must be enabled. For details on controlling I/O ports, refer to section 23.4.1, Procedure for Specifying Input/Output Pin Function.

PES[3:0] Bits (Periodic Interrupt Select)

These bits specify the period for the periodic interrupt. A periodic interrupt is generated with the period specified by these bits.

32.2.18 RTC Control Register 2 (RCR2)

Address(es): RTC.RCR2 0008 C424h

	b7	b6	b5	b4	b3	b2	b1	b0
	CNTMD	HR24	AADJP	AADJE	RTCOE	ADJ30	RESET	START
Value after reset:	x	x	x	x	0	0	0	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	START	Start	0: Prescaler and counter are stopped. 1: Prescaler and counter operate normally.	R/W
b1	RESET	RTC Software Reset	• In writing 0: Writing is invalid. 1: The prescaler and the target registers for RTC software reset*1 are initialized • In reading 0: In normal time operation, or an RTC software reset has completed. 1: During an RTC software reset	R/W
b2	ADJ30	30-Second Adjustment*2	• In writing 0: Writing is invalid. 1: 30-second adjustment is executed. • In reading 0: In normal time operation, or 30-second adjustment has completed. 1: During 30-second adjustment	R/W
b3	RTCOE	RTCOUT Output Enable	0: RTCOUT output disabled. 1: RTCOUT output enabled.	R/W
b4	AADJE	Automatic Adjustment Enable *3, *4	0: Automatic adjustment is disabled. 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select*3, *4	0: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every minute (every 32 seconds in binary counter mode). 1: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every 10 seconds (every 8 seconds in binary counter mode).	R/W
b6	HR24	Hours Mode*2, *4	0: The RTC operates in 12-hour mode. 1: The RTC operates in 24-hour mode.	R/W
b7	CNTMD	Count Mode Select	0: The calendar count mode. 1: The binary count mode.	R/W

Note 1. R64CNT, RSECAR/BCNT0AR, RMINAR/BCNT1AR, RHRAR/BCNT2AR, RWKAR/BCNT3AR, RDAYAR/BCNT0AER, RMONAR/BCNT1AER, RYRAR/BCNT2AER, RYRAREN/BCNT3AER, RADJ, RTCCRy, RSECCPy/BCNT0CPy, RMINCPy/BCNT1CPy, RHRCPy/BCNT2CPy, RDAYCPy/BCNT3CPy, RMONCPy, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP
Note 2. This bit is reserved in binary counter mode. The write value should be 0.
Note 3. When the main clock is selected, the setting of this bit is disabled.
Note 4. After writing to this bit, confirm that its value has actually changed before proceeding with further processing. Refer to section 32.6.5, Notes When Writing to and Reading from Registers for notes on accessing registers.

The RCR2 register is related to hours mode, automatic adjustment function, enabling the RTCOUT output, 30-second adjustment, RTC software reset, and controlling count operation.

START Bit (Start)

This bit stops or restarts the prescaler or counter (clock) operation.

The START bit is updated in synchronization with the count source. When the START bit is modified, check that the bit is updated before proceeding to the next processing.

RESET Bit (RTC Software Reset)

This bit initializes the prescaler and registers to be reset by RTC software reset.
When 1 is written to the RESET bit, the initialization starts in synchronization with the count source. When the initialization is completed, the RESET bit is automatically set to 0.
When 1 is written to the RESET bit, check that the bit is set to 0, and then make next settings.

ADJ30 Bit (30-Second Adjustment)

This bit is for 30-second adjustment.
When 1 is written to the ADJ30 bit, the RSECCNT value of 30 seconds or less is rounded down to 00 second and the value of 30 seconds or more is rounded up to 1 minute.
The 30-second adjustment is performed in synchronization with the count source. When 1 is written to this bit, the ADJ30 bit is automatically set to 0 after the 30-second adjustment is completed. In case when 1 is written to the ADJ30 bit, check that the bit is set to 0, and then make next settings.
When the 30-second adjustment is performed, the prescaler and R64CNT are also reset.
The ADJ30 bit is set to 0 by an RTC software reset.
This bit is reserved in binary counter mode. The write value should be 0.

RTCOE Bit (RTCOUT Output Enable)

This bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin.
Use the START bit to stop counting by the counters before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time.
When RTCOUT signal is to be output from an external pin, set the RTCOE bit to 1 and set up the port control for the pin.

AADJE Bit (Automatic Adjustment Enable)

This bit controls (enables or disables) automatic adjustment.
Set the plus–minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit.
The AADJE bit is set to 0 by an RTC software reset.

AADJP Bit (Automatic Adjustment Period Select)

This bit selects the automatic-adjustment period.
Set the plus–minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJP bit.
The AADJP bit is set to 0 by an RTC software reset.

HR24 Bit (Hours Mode)

This bit specifies whether the RTC will operate in 12- or 24-hour mode.
Use the START bit to stop counting by the counters before changing the value of the HR24 bit. Do not stop counting (write 0 to the START bit) and change the value of the HR24 bit at the same time.
This bit is reserved in binary counter mode. The write value should be 0.

CNTMD Bit (Count Mode Select)

This bit specifies whether the RTC count mode is operated in calendar count mode or in binary count mode.
When setting the count mode, execute an RTC software reset and start again from the initial settings.
This bit is updated synchronously with the count source, and its value is fixed before the RTC software reset is completed.
For details on initial settings, refer to section 32.3.1, Outline of Initial Settings of Registers after Power On.

32.2.19 RTC Control Register 3 (RCR3)

Address(es): RTC.RCR3 0008 C426h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	RTCDV[2:0]			RTCEN
Value after reset:	0	0	0	0	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	RTCEN	Sub-Clock Oscillator Control	0: Sub-clock oscillator is stopped. 1: Sub-clock oscillator is operating.	R/W
b3 to b1	RTCDV[2:0]	Sub-Clock Oscillator Drive Capacity Control	b3 b1 0 0 0: Setting prohibited 0 0 1: Drive capacity for low CL 0 1 0: Setting prohibited 0 1 1: Setting prohibited 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Drive capacity for standard CL 1 1 1: Setting prohibited	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RCR3 register is used for controlling the sub-clock oscillator in the clock generation circuit. For details on controlling the sub-clock oscillator, refer to section 9, Clock Generation Circuit.
This register is a function common to calendar count mode and binary count mode.
When this register is modified, check that all the bits have been updated before proceeding to the next processing.

RTCEN Bit (Sub-Clock Oscillator Control)

The RTCEN bit and a clock generation circuit register (the SOSCCR.SOSTP bit) control whether to operate or stop the sub-clock oscillator. If one of the bits is set so as to enable the operation, the sub-clock oscillator runs.
When using the sub-clock as the count source to the RTC, set the sub-clock oscillator using the RTCEN bit.
When the main clock is selected (RCR4.RCKSEL bit = 1), making the sub-clock oscillator run or stop is controlled only by the SOSCCR.SOSTP bit regardless of the value set in the RTCEN bit.

RTCDV[2:0] Bits (Sub-Clock Oscillator Drive Capacity Control)

These bits control the drive capacity of the sub-clock oscillator.
It must be noted that the oscillation accuracy of the sub-clock oscillator is affected if the RTCDV[2:0] bits are set to 001b (drive capacity for low CL) when an on-chip debugging emulator is attached. The oscillation accuracy is not affected when the RCR3.RTCDV[2:0] bits are set to 110b (drive capacity for standard CL).
Before modifying the RTCDV[2:0] bits, be sure to stop the sub-clock oscillator. When the main clock is selected (RCR4.RCKSEL bit = 1), making the sub-clock oscillator run or stop is controlled only by the SOSCCR.SOSTP bit regardless of the value set in the RTCEN bit.

32.2.19.1 Notes on using a low CL crystal unit

When the RCR3.RTCDV[2:0] bits are 001b (drive capacity for low CL), the oscillator is susceptible to noise. Especially when the signal level of any pin near the XCIN or XCOUT pin is changed, the oscillation accuracy of the sub-clock oscillator may be affected. The accuracy is affected differently depending on the board traces and how the signal level of any pin near the XCIN or XCOUT pin is changed. When designing a board using a low CL crystal unit, refer to the application note “Design Guide for Low CL Sub-clock Circuits” (R01AN1187EJ) to reduce the influence from noise. The following are examples that may significantly affect oscillation accuracy:

(1) When connecting an on-chip debugging emulator to the FINED pin

Since the FINED pin (FINE interface pin) is near the XCIN and XCOUT pins, the oscillation accuracy of the sub-clock oscillator is affected when using the FINED pin in debugging. When using the FINED pin in debugging, keep using the low CL crystal unit and set the RCR3.RTCDV[2:0] bits to 110b (drive capacity for standard CL). However, this measure may affect the reliability of the crystal unit. Therefore, use this measure only when using an on-chip debugging emulator. Set the RCR3.RTCDV[2:0] bits to 001b (drive capacity for low CL) in mass production programs. The oscillation accuracy is not affected when connecting an on-chip debugging emulator to a JTAG pin (TCK, TRST, TMS, TDI, or TDO pin).

(2) When supplying an external clock to the main clock oscillator

When inputting an external clock to the EXTAL pin, the oscillation accuracy of the sub-clock oscillator may be affected. When inputting an inverted external clock to the XTAL pin, the oscillation accuracy will be affected more significantly.

32.2.20 RTC Control Register 4 (RCR4)

Address(es): RTC.RCR4 0008 C428h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RCKSEL
Value after reset:	0	0	0	0	0	0	0	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	RCKSEL	Count Source Select	0: Sub-clock oscillator is selected. 1: Main clock oscillator is selected.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RCR4 register is used for selecting the count source. This function is used in both calendar count mode and binary count mode.
When the RCKSEL bit is set to 0, the time is counted with the sub-clock. When the bit is set to 1, the time is counted with the main clock.

RCKSEL Bit (Count Source Select)

This bit selects the count source from the sub-clock and the main clock.
The count source should be selected only once before the initial settings of the RTC registers at power on.

32.2.21 Frequency Register H/L (RFRH/RFRL)

Address(es): RTC.RFRH 0008 C42Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFC[16]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

x: Undefined

Address(es): RTC.RFRL 0008 C42Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFC[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

- RFRH register

Bit	Symbol	Bit Name	Description	R/W
b0	RFC[16]	Frequency Divide Ratio Setting	Set the divide ratio of the main clock to generate the reference clock of 128 Hz. If the setting value in the RFC[16:0] bits is n, the prescaler divides the main clock by n + 1.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- RFRL register

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RFC[15:0]	Frequency Divide Ratio Setting	Set the divide ratio of the main clock to generate the reference clock of 128 Hz. If the setting value in the RFC[16:0] bits is n, the prescaler divides the main clock by n + 1.	R/W

RFRH/RFRL is a register for controlling the prescaler when the main clock is selected.
The RTC time counter operates on a 128-Hz clock signal as the reference clock. Therefore, when the main clock is selected, the main clock is divided by the prescaler to generate a 128-Hz clock signal. Set the frequency divide ratio in the RFC[16:0] bits to generate a 128-Hz clock from the main clock frequency. As for the calculation method, refer to the formula below.

A value from 0 0007h through 1 FFFFh can be specified in the RFC[16:0] bits; if a value outside of this range is specified, the RTC does not operate correctly. Rewrite this register while the RCR2.START bit is 0 (counter is stopped). In addition, the range of the main clock frequency that can be used to generate a 128-Hz clock is 1.024 kHz to 16.778 MHz.
The frequency of the peripheral module clock and the main clock should be in the relationship that the peripheral module clock ≥ the main clock.

- Calculation method of frequency comparison value

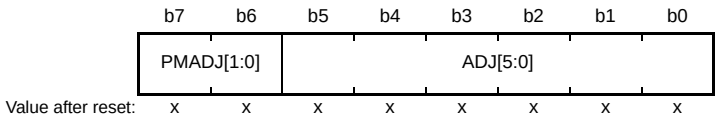
RFC[16:0] = (Main clock frequency) ÷ 128 – 1

Table 32.3 RFRH/L Register Settings by the Main Clock Frequency

Main Clock Frequency	RFRH/L Register Setting Value
4 MHz	0000 7A11h
8 MHz	0000 F423h
10 MHz	0001 312Ch
12 MHz	0001 6E35h
16 MHz	0001 E847h

32.2.22 Time Error Adjustment Register (RADJ)

Address(es): RTC.RADJ 0008 C42Eh



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	ADJ[5:0]	Adjustment Value	These bits specify the adjustment value from the prescaler.	R/W
b7, b6	PMADJ[1:0]	Plus–Minus	b7 b6 0 0: Adjustment is not performed. 0 1: Adjustment is performed by the addition to the prescaler. 1 0: Adjustment is performed by the subtraction from the prescaler. 1 1: Setting prohibited	R/W

Adjustment is performed by the addition to or subtraction from the prescaler.
In case when the automatic adjustment enable (RCR2.AADJE) bit is 0, adjustment is performed when writing to the RADJ.
In case when the RCR2.AADJE bit is 1, adjustment is performed in the interval specified by the automatic adjustment period select (RCR2.AADJP) bit.
The current adjustment by software (disabling automatic adjustment) may be invalid if the following adjustment value is specified within 320 cycles of the count source after the register setting. To perform adjustment consecutively, wait for 320 cycles or more of the count source after the register setting and then specify the next adjustment value.
RADJ is updated in synchronization with the count source. When RADJ is modified, check that all the bits have been updated before continuing with further processing.
This register is set to 00h by an RTC software reset.
The setting of this register is enabled only when the sub-clock is selected.
When the main clock is selected, adjustment is not performed.

ADJ[5:0] Bits (Adjustment Value)

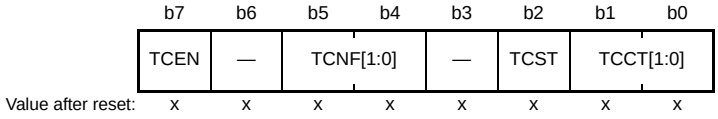
These bits specify the adjustment value (the number of sub-clock cycles) from the prescaler.

PMADJ[1:0] Bits (Plus–Minus)

These bits select whether the clock is set ahead or back depending on the error-adjustment value set in the ADJ[5:0] bits.

32.2.23 Time Capture Control Register y (RTCCRy) (y = 0 to 2)

Address(es): RTC.RTCCR0 0008 C440h, RTC.RTCCR1 0008 C442h, RTC.RTCCR2 0008 C444h



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TCCT[1:0]	Time Capture Control	b1 b0 0 0: No event is detected. 0 1: Rising edge is detected. 1 0: Falling edge is detected. 1 1: Both edges are detected.	R/W
b2	TCST	Time Capture Status	0: No event is detected. 1: An event is detected.*1	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	TCNF[1:0]	Time Capture Noise Filter Control	b5 b4 0 0: The noise filter is off. 0 1: Setting prohibited 1 0: The noise filter is on (count source). 1 1: The noise filter is on (count source by divided by 32).	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	TCEN	Time Capture Event Input Pin Enable	0: The RTCICn pin is disabled as the time capture event input. 1: The RTCICn pin is enabled as the time capture event input. (n = 0 to 2)	R/W

Note 1. Indicates that an event has been detected. Writing 1 to this bit has no effect. Writing 0 sets this bit to 0.

The RTCCRy register is used both in calendar count mode and in binary count mode.
RTCCR0, RTCCR1, and RTCCR2 control the RTCIC0, RTCIC1, and RTCIC2 pins, respectively.
RTCCRy is updated in synchronization with the count source. When RTCCRy is modified, check that all the bits except for the TCST bit have been updated before continuing with further processing.
This register is set to 00h by an RTC software reset.

TCCT[1:0] Bits (Time Capture Control)

These bits control the edge detection of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2). The detection edge is selectable. The TCCT[1:0] bits should be set while the TCEN bit is 1.

TCST Bit (Time Capture Status)

This bit indicates that an event of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2) has been detected. When the TCST bit is 0, no event is detected.
When the TCST bit is 1, this bit indicates that an event of the corresponding pin has been detected and the capture register is valid. When multiple events have been detected, the capture time for the first event is retained.
If an event is detected while the count operation is stopped (the RCR2.START bit is 0), the captured value is not guaranteed. In this case, set the TCST bit to 0 for deleting the captured value.
Writing 0 sets the TCST bit to 0. In addition, writing any other value except 0 has no effect.
Set the TCST bit while the TCCT[1:0] bits are 00b (no event is detected).
The TCST bit is set to 0 in synchronization with the count source. When the TCST bit is set to 0, check that the bit has been updated before continuing with further processing.

TCNF[1:0] Bits (Time Capture Noise Filter Control)

These bits control the noise filter of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2).
When the noise filter is on, the count source divided by 1 or divided by 32 is selectable. In this case, when the input level on the time capture event input pin matches three consecutive times at the set sampling period, the input level is determined.
Set the TCNF[1:0] bits while the TCCT[1:0] bits are 00b (no event is detected). When the noise filter is used, set the TCNF[1:0] bits, wait for three cycles of the specified sampling period, and then set the TCCT[1:0] bits. Set the TCNF[1:0] bits when the TCEN bit is 1.

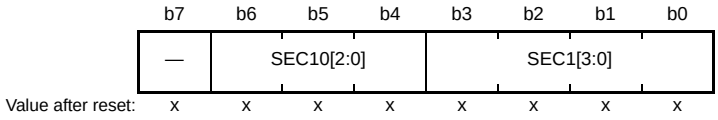
TCEN Bit (Time Capture Event Input Pin Enable)

This bit enables or disables the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2).
When the sub-clock is selected (RCR4.RCKSEL bit = 0) and the sub-clock oscillator is stopped (RCR3.RTCEN bit = 0), the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2) are disabled regardless of the value of the TCEN bit.
When the functions of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2) are multiplexed, set the port control and enable this bit. In this case, the port control should be set first. If the TCEN bit is set to 0, set also the TCCT[1:0] bits to 00b.

32.2.24 Second Capture Register y (RSECCPy) (y = 0 to 2)/BCNT0 Capture Register y (BCNT0CPy) (y = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RSECCP0 0008 C452h, RTC.RSECCP1 0008 C462h, RTC.RSECCP2 0008 C472h



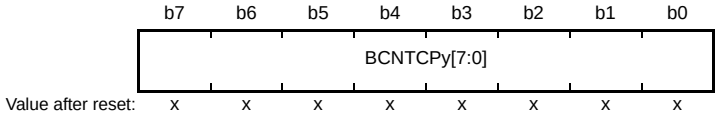
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	1-Second Capture	Capture value for the ones place of seconds	R
b6 to b4	SEC10[2:0]	10-Second Capture	Capture value for the tens place of seconds	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset.	R

RSECCPy is a read-only register that captures the RSECCNT value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RSECCP0, RSECCP1, and RSECCP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT0CP0 0008 C452h, RTC.BCNT0CP1 0008 C462h, RTC.BCNT0CP2 0008 C472h



x: Undefined

BCNT0CPy is a read-only register that captures the BCNT0 value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT0CP0, BCNT0CP1, and BCNT0CP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

32.2.25 Minute Capture Register y (RMINCPy) (y = 0 to 2)/BCNT1 Capture Register y (BCNT1CPy) (y = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RMINCP0 0008 C454h, RTC.RMINCP1 0008 C464h, RTC.RMINCP2 0008 C474h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MIN10[2:0]			MIN1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	1-Minute Capture	Capture value for the ones place of minutes	R
b6 to b4	MIN10[2:0]	10-Minute Capture	Capture value for the tens place of minutes	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset.	R

RMINCPy is a read-only register that captures the RMINCNT value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMINCP0, RMINCP1, and RMINCP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT1CP0 0008 C454h, RTC.BCNT1CP1 0008 C464h, RTC.BCNT1CP2 0008 C474h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTCPy[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT1CPy is a read-only register that captures the BCNT1 value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT1CP0, BCNT1CP1, and BCNT1CP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

32.2.26 Hour Capture Register y (RHRCPy) (y = 0 to 2)/BCNT2 Capture Register y (BCNT2CPy) (y = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RHRCP0 0008 C456h, RTC.RHRCP1 0008 C466h, RTC.RHRCP2 0008 C476h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	1-Hour Capture	Capture value for the ones place of hours	R
b5, b4	HR10[1:0]	10-Hour Capture	Capture value for the tens place of hours	R
b6	PM	PM	0: a.m. 1: p.m.	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset.	R

RHRCPy is a read-only register that captures the RHRCNT value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RHRCP0, RHRCP1, and RHRCP2 registers, respectively.
The PM bit is only enabled when the RCR2.HR24 bit is 0 (in 12-hour mode).
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT2CP0 0008 C456h, RTC.BCNT2CP1 0008 C466h, RTC.BCNT2CP2 0008 C476h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTCPy[23:16]							
Value after reset:	x	x	x	x	x	x	x	x

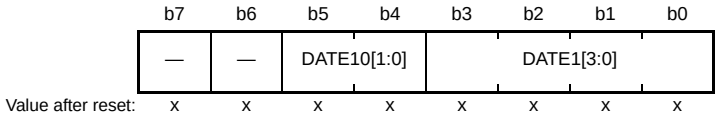
x: Undefined

BCNT2CPy is a read-only register that captures the BCNT2 value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT2CP0, BCNT2CP1, and BCNT2CP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

32.2.27 Date Capture Register y (RDAYCPy) (y = 0 to 2)/BCNT3 Capture Register y (BCNT3CPy) (y = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RDAYCP0 0008 C45Ah, RTC.RDAYCP1 0008 C46Ah, RTC.RDAYCP2 0008 C47Ah



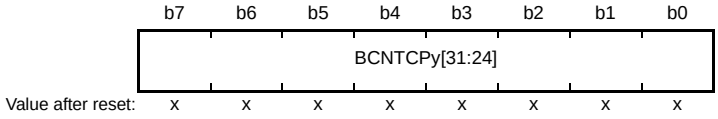
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	1-Day Capture	Capture value for the ones place of days	R
b5, b4	HR10[1:0]	10-Day Capture	Capture value for the tens place of days	R
b7, b6	—	Reserved	These bits are read as 0 after an RTC software reset.	R

RDAYCPy is a read-only register that captures the RDAYCNT value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RDAYCP0, RDAYCP1, and RDAYCP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT3CP0 0008 C45Ah, RTC.BCNT3CP1 0008 C46Ah, RTC.BCNT3CP2 0008 C47Ah



x: Undefined

BCNT3CPy is a read-only register that captures the BCNT3 value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT3CP0, BCNT3CP1, and BCNT3CP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

32.2.28 Month Capture Register y (RMONCPy) (y = 0 to 2)

Address(es): RTC.RMONCP0 0008 C45Ch, RTC.RMONCP1 0008 C46Ch, RTC.RMONCP2 0008 C47Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MON10	MON1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	1-Month Capture	Capture value for the ones place of months	R
b4	HR10[1:0]	10-Month Capture	Capture value for the tens place of months	R
b7 to b5	—	Reserved	These bits are read as 0	R

RMONCPy is a read-only register that captures the RMONCNT value when a time capture event is detected. The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMONCP0, RMONCP1, and RMONCP2 registers, respectively.

This register is set to 00h by an RTC software reset.

Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

32.3 Operation

32.3.1 Outline of Initial Settings of Registers after Power On

After the power is turned on, the initial settings for the clock setting, count mode setting, time error adjustment, time setting, alarm, interrupt, and time capture control register should be performed.

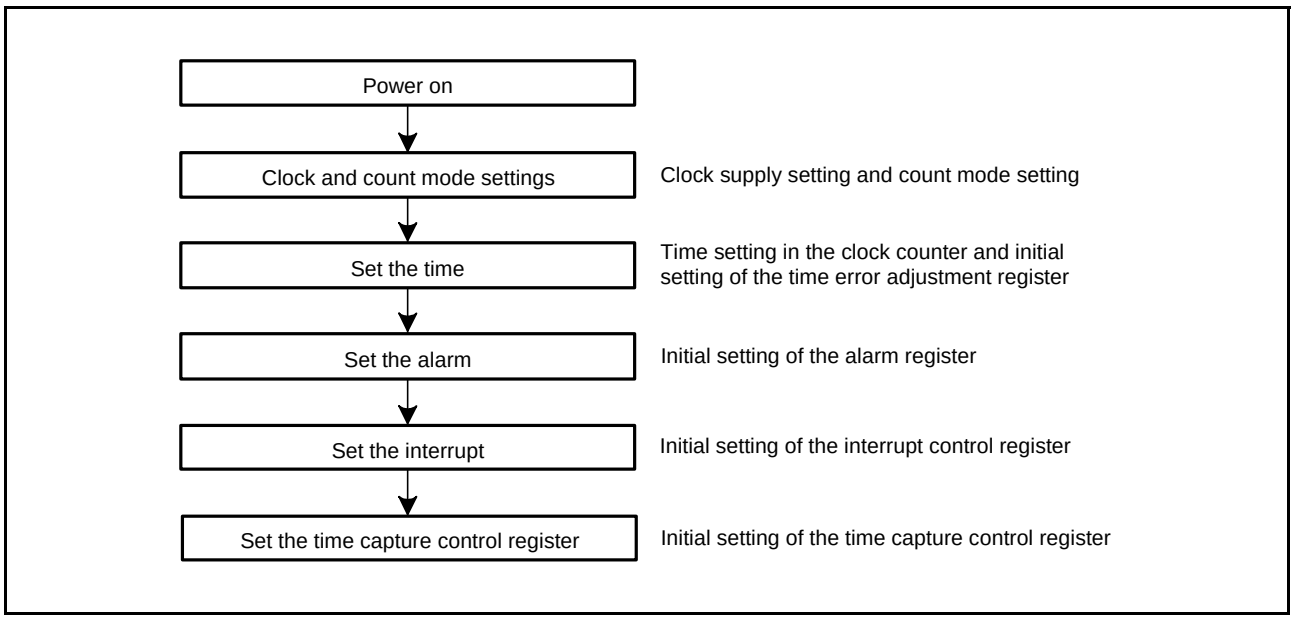
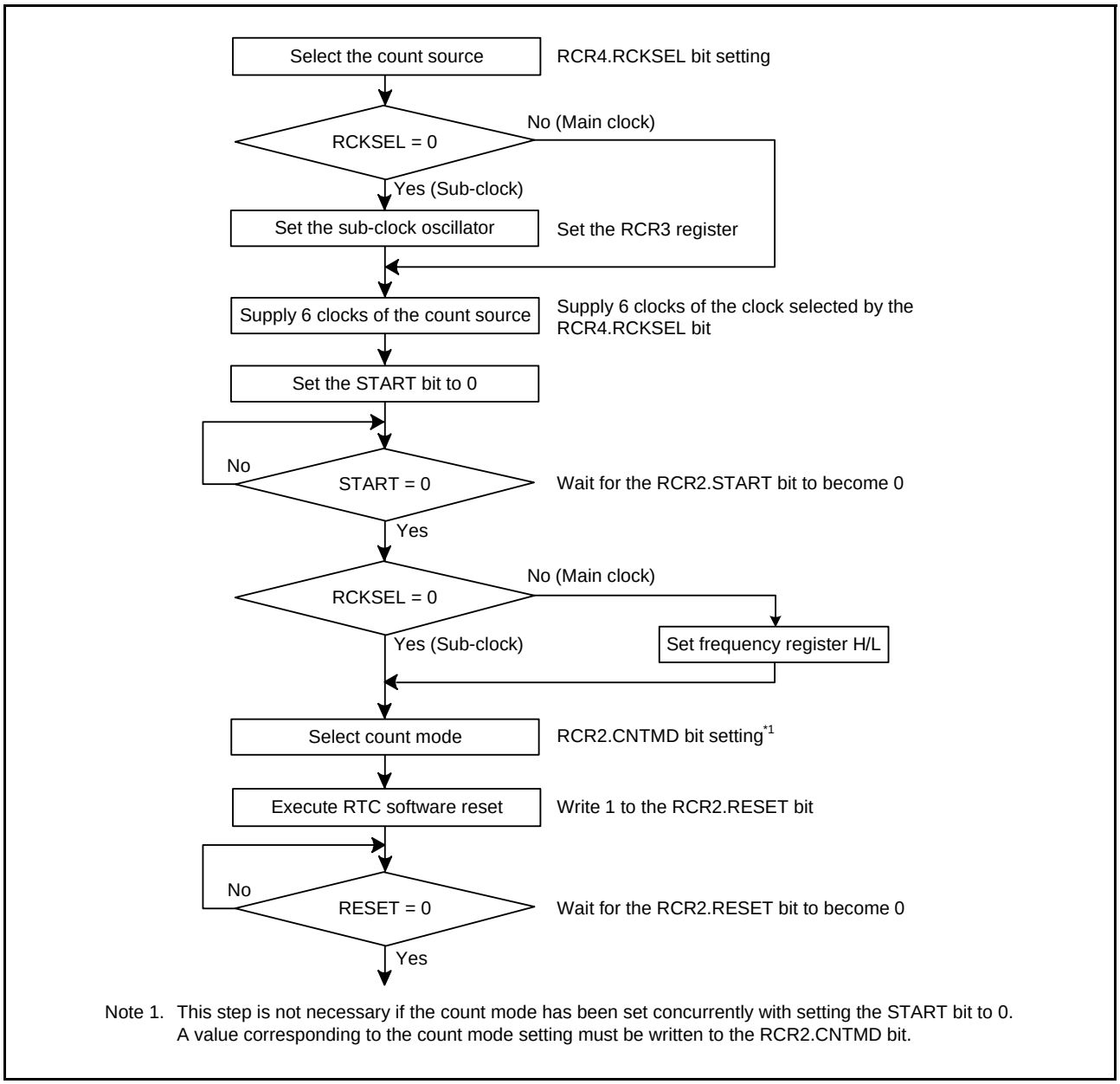


Figure 32.2 Outline of Initial Settings after Power On

32.3.2 Clock and Count Mode Setting Procedure

Figure 32.3 shows how to set the clock and the count mode.

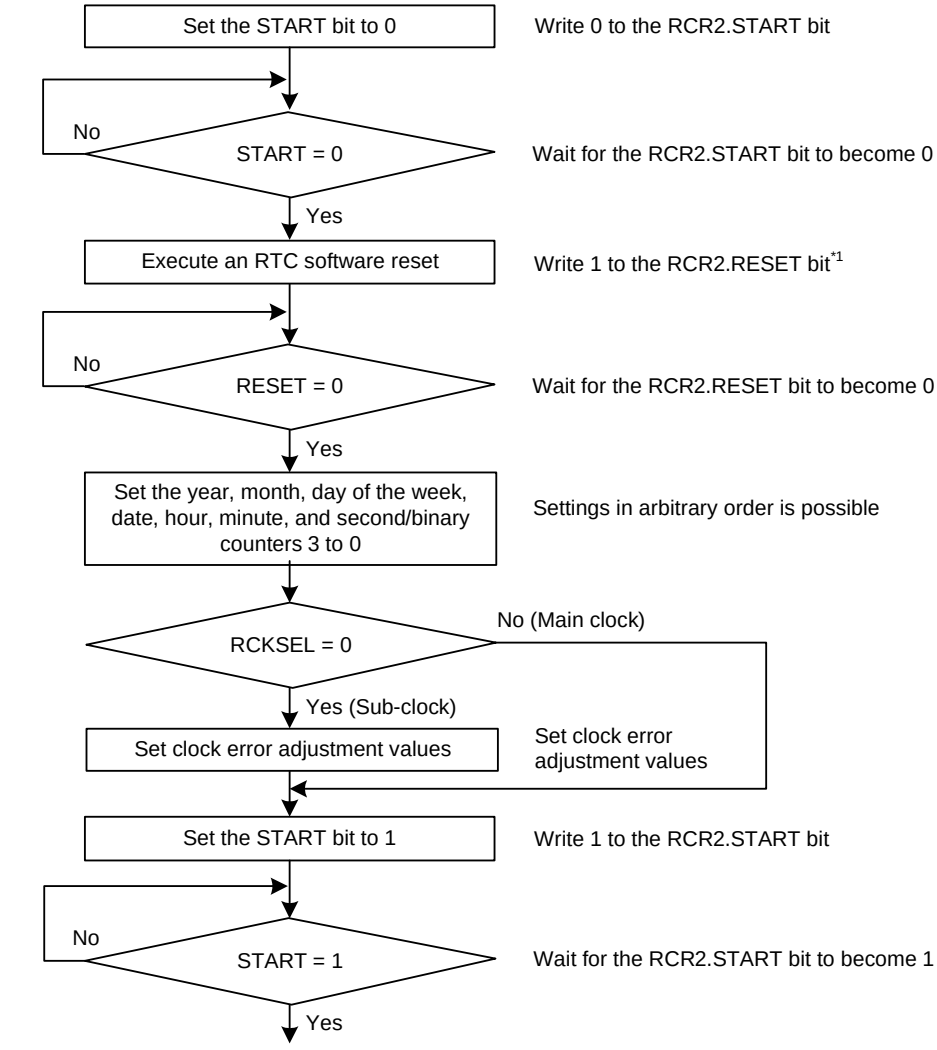


Note 1. This step is not necessary if the count mode has been set concurrently with setting the START bit to 0. A value corresponding to the count mode setting must be written to the RCR2.CNTMD bit.

Figure 32.3 Clock and Count Mode Setting Procedure

32.3.3 Setting the Time

Figure 32.4 shows how to set the time.



Note 1. This step is not necessary for the time-setting procedure because an RTC software reset is executed in the clock setting procedure of the initial settings for the power supply.

Figure 32.4 Setting the Time

32.3.4 30-Second Adjustment

Figure 32.5 shows how to execute 30-second adjustment.

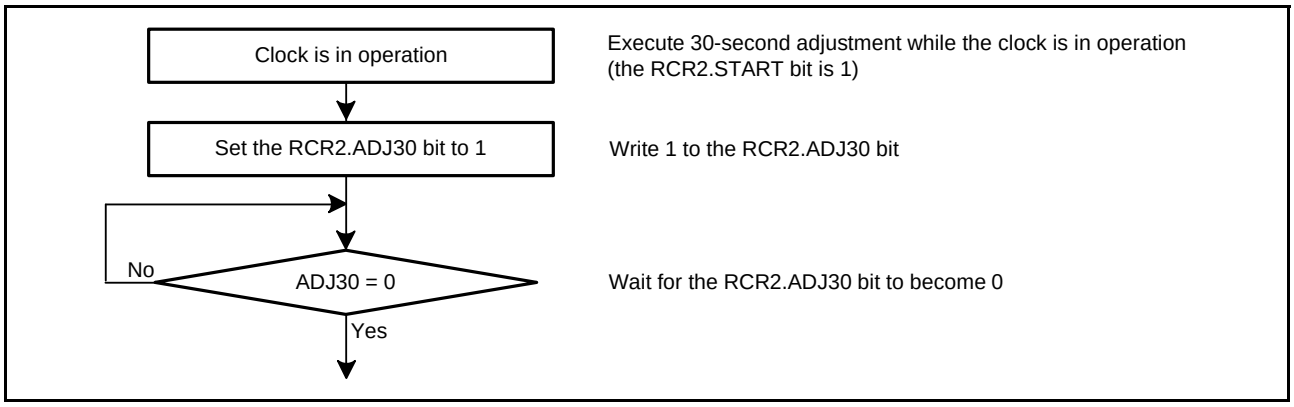


Figure 32.5 30-Second Adjustment

32.3.5 Reading 64-Hz Counter and Time

Figure 32.6 shows how to read the 64-Hz counter and time.

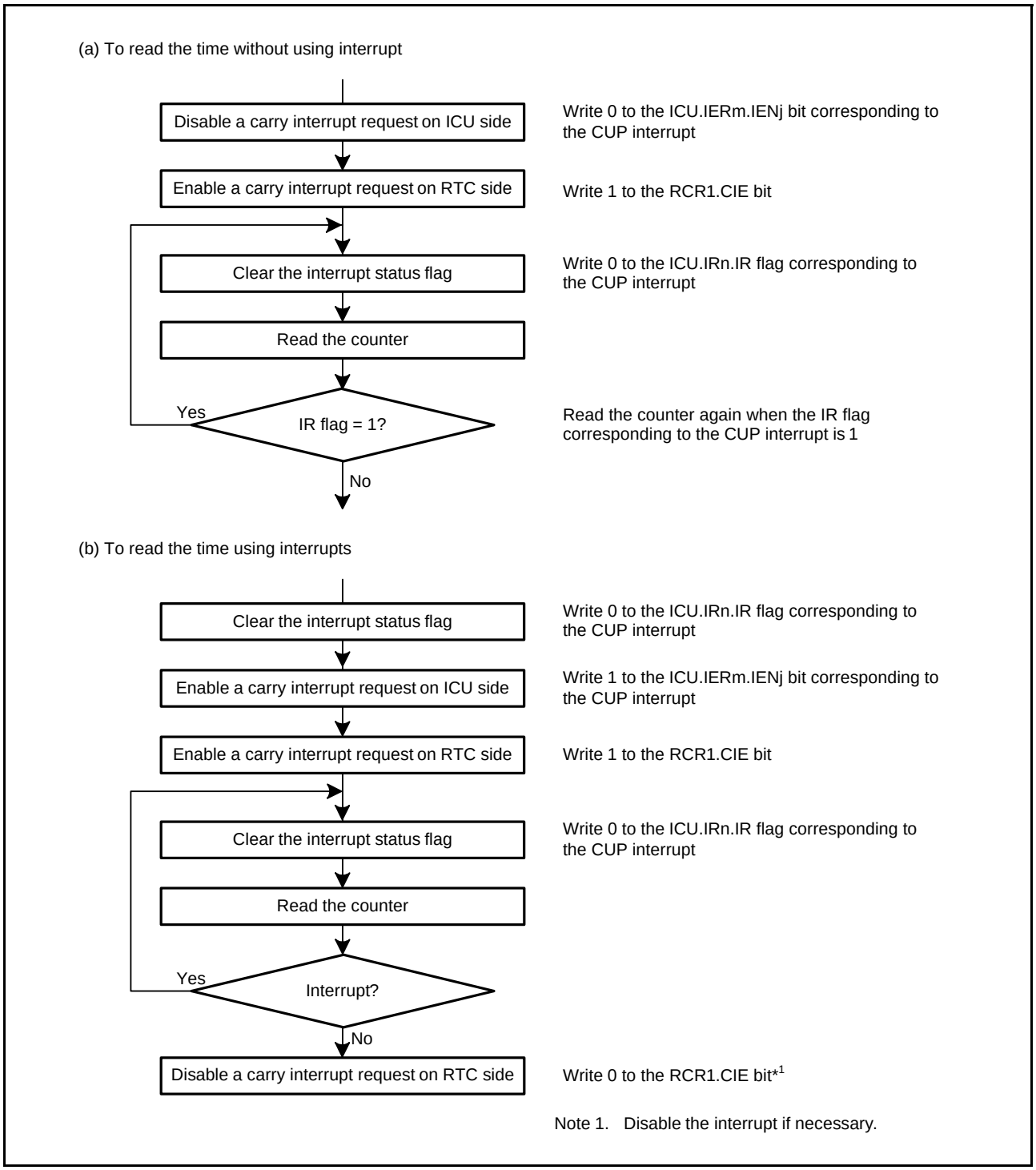


Figure 32.6 Reading Time

If a carry occurs while the 64-Hz counter and time are being read, the correct time will not be obtained, so they must be read again. The procedure for reading the time without using interrupts is shown in (a) in Figure 32.6, and the procedure using carry interrupts in (b). To keep the program simple, method (a) should be used in most cases.

32.3.6 Alarm Function

Figure 32.7 shows how to use the alarm function.

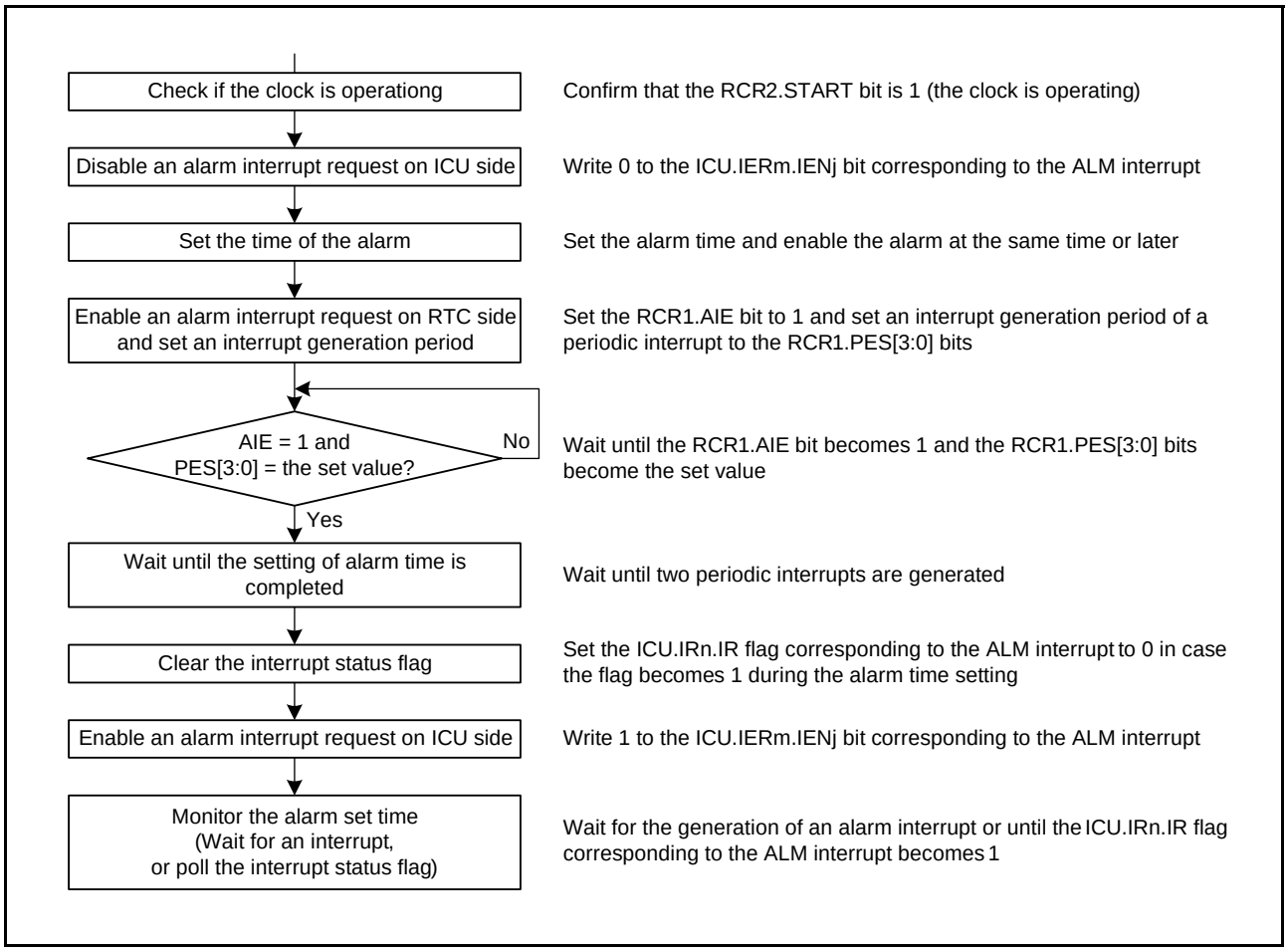


Figure 32.7 Using Alarm Function

In calendar count mode, an alarm can be generated by any one of year, month, date, day-of-week, hour, minute or second, or any combination of those. Write 1 to the ENB bit in the alarm registers involved in the alarm setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not involved in the alarm setting.

In binary count mode, an alarm can be generated in any bit combination of 32 bits. Write 1 to the ENB bit of the alarm enable register corresponding to the target bit of the alarm, and set the alarm time to the alarm register. For bits that are not target of the alarm, write 0 to the ENB bit of the alarm enable register.

When the counter and the alarm time match, the IR flag corresponding to the ALM interrupt is set to 1. Alarm detection can be confirmed by reading this bit, but an interrupt should be used in most cases. If 1 has been set in the interrupt request enable bit corresponding to the ALM interrupt, an alarm interrupt is generated in the event of alarm, enabling the alarm to be detected.

Writing 0 sets the IR flag corresponding to the ALM interrupt to 0.

When the counter and the alarm time match in a low power consumption state, the MCU returns from the low power consumption state. In deep software standby mode, the MCU returns from the deep software standby mode even when the alarm interrupt request is disabled.

32.3.7 Procedure for Disabling Alarm Interrupt

Figure 32.8 shows the procedure for disabling the enabled alarm interrupt request.

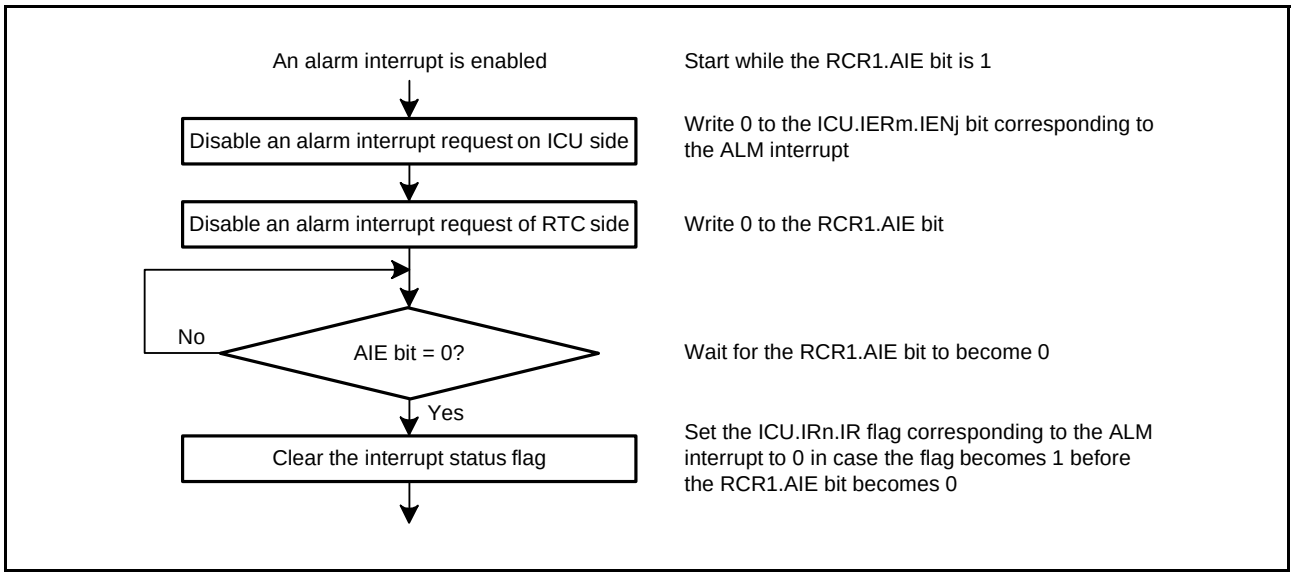


Figure 32.8 Procedure for Disabling Alarm Interrupt Request

32.3.8 Time Error Adjustment Function

The time error adjustment function is used to correct errors (running fast or slow) in the time due to the precision of oscillation by the sub-clock. Since 32,768 cycles of the sub-clock constitute 1 second of operation when the sub-clock is selected, the clock runs fast if the sub-clock frequency is high and slow if the sub-clock frequency is low. This function can be used to correct errors due to the clock running fast or slow.
Two types of time error adjustment functions are provided: automatic adjustment and adjustment by software.
Use the RCR2.AADJE bit to select automatic adjustment or adjustment by software.

32.3.8.1 Automatic Adjustment

Enable automatic adjustment by setting the RCR2.AADJE bit to 1.
Automatic adjustment is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register every time the adjustment period selected by the RCR2.AADJE bit elapses.
Examples are shown below.

[Example 1] Sub-clock running at 32.769 kHz
Adjustment procedure:
When the sub-clock is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by 60 clock cycles per minute, so adjustment can take the form of setting the clock back by 60 cycles every minute.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 0 (adjustment every minute)
- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler.)
- RADJ.ADJ[5:0] = 60 (3Ch)

[Example 2] Sub-clock running at 32.766 kHz
Adjustment procedure:
When the sub-clock is running at 32.766 kHz, 1 second elapses every 32,766 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs slow by two clock cycles every second. The time on the clock is slow by 20 clock cycles every 10 seconds, so adjustment can take the form of setting the clock forward by 20 cycles every 10 seconds.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 1 (adjustment every 10 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler.)
- RADJ.ADJ[5:0] = 20 (14h)

[Example 3] Sub-clock running at 32.764 kHz
Adjustment procedure:
At 32.764 kHz, 1 second elapses on 32,764 clock cycles. Since the RTC operates for 32,768 clock cycles as 1 second, the clock is delayed for four clock cycles per second. In 8 seconds, the delay is 32 clock cycles, so correction can be made by proceeding the clock for 32 clock cycles every 8 seconds.

Register settings when the RCR2.CNTMD bit is 1

- RCR2.AADJP = 1 (adjustment every 8 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler.)
- RADJ.ADJ[5:0] = 32 (20h)

32.3.8.2 Adjustment by Software

Enable adjustment by software by setting the RCR2.AADJE bit to 0.
Adjustment by software is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register at the time of execution of an instruction for writing to the RADJ register.
An example is shown below.

[Example 1] Sub-clock running at 32.769 kHz
Adjustment procedure:
When the sub-clock is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by one clock cycle per second, so adjustment can take the form of setting the clock back by one cycle every second.

Register settings:

- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler.)
- RADJ.ADJ[5:0] = 1 (01h)
  This is written to the RADJ register once per 1-second interrupt.

32.3.8.3 Procedure for Changing the Mode of Adjustment

When changing the mode of adjustment, change the value of the AADJE bit in RCR2 after setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

Changing from adjustment by software to automatic adjustment:

(1) Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
(2) Set the RCR2.AADJE bit to 1 (automatic adjustment is enabled).
(3) Use the RCR2.AADJP bit to select the period of adjustment.
(4) In RADJ, set the PMADJ[1:0] bits for addition or subtraction and the ADJ[5:0] bits to the value for use in time error adjustment.

Changing from adjustment by software to automatic adjustment:

(1) Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
(2) Set the RCR2.AADJE bit to 0 (adjustment by software is enabled).
(3) Proceed with adjustment by setting the RADJ.PMADJ[1:0] bits for addition or subtraction and the RADJ.ADJ[5:0] bits to the value for use in time error adjustment at the desired time. After that, the time is adjusted every time a value is written to the RADJ register.

32.3.8.4 Procedure for Stopping Adjustment

Stop adjustment by setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

32.3.8.5 Capturing the Time

The RTC is capable of storing the month, date, hour, minute and second/binary counters 3 to 0 by detecting an edge of a signal on a time capture event input pin.

A noise filter can also be used on a time capture event input pin. If the noise filter is enabled, the TCST bit is set to 1 when the input level on the pin matches three times.

The noise filter can be switched on or off for each of the time capture event input pins. Operation when the noise filter is off is shown in Figure 32.9 and operation when the noise filter is on is shown in Figure 32.10.

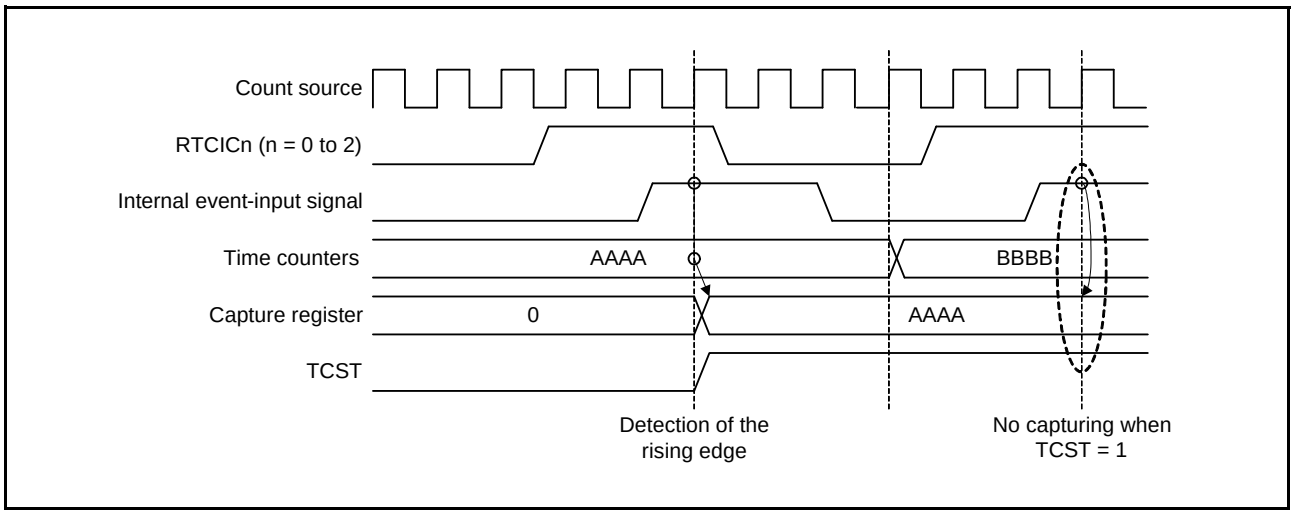


Figure 32.9 Timing of a Time Capture Operation (with the Filter Off)

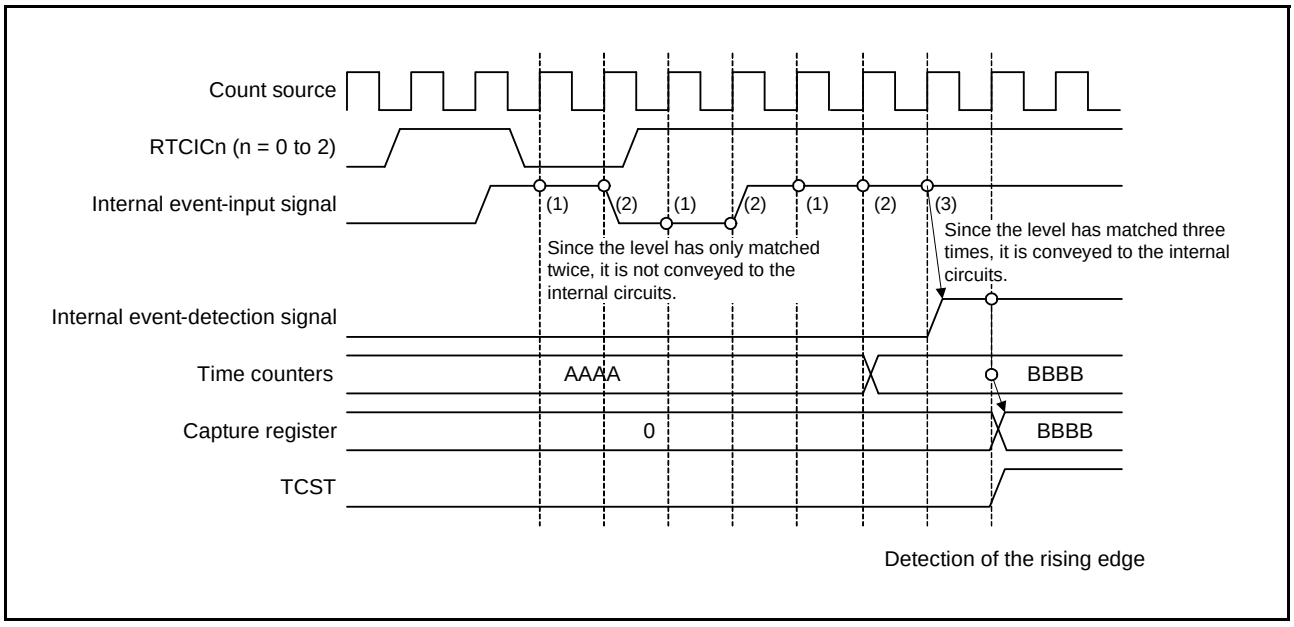


Figure 32.10 Timing of a Time Capture Operation (with the Filter On)

32.4 Interrupt Sources

There are three interrupt sources in the realtime clock. Table 32.4 lists interrupt sources for the RTC.

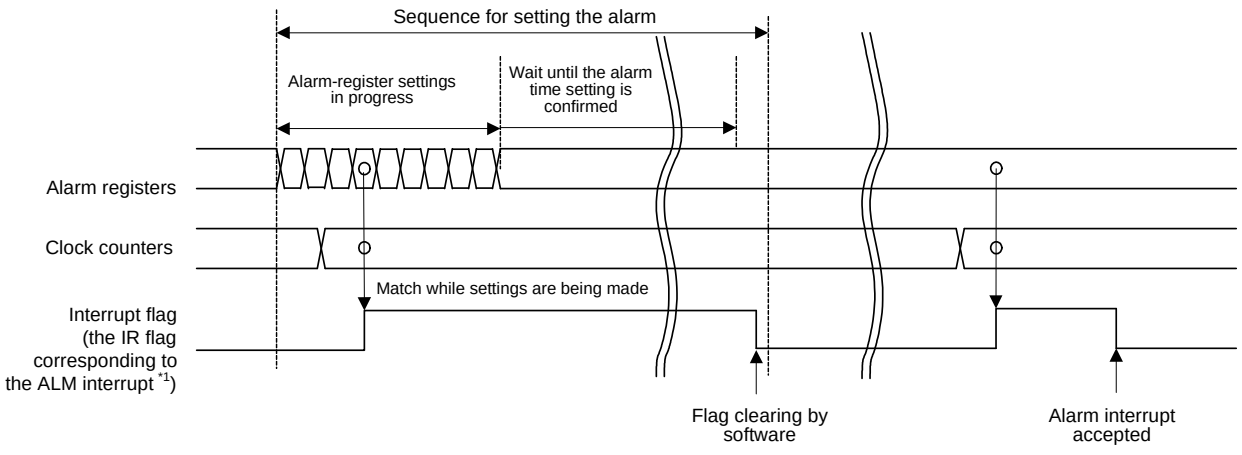
Table 32.4 RTC Interrupt Sources

Name	Interrupt Sources
ALM	Alarm interrupt
PRD	Periodic interrupt
CUP	Carry interrupt

(1) Alarm interrupt (ALM)

This interrupt is generated according to the result of comparison between the alarm registers and realtime clock counters (for details, refer to section 32.3.6, Alarm Function).

Since there is a possibility that the interrupt flag may be set to 1 when the settings of the alarm registers match the clock counters, wait for the alarm time settings to be confirmed and set the IR flag corresponding to the ALM interrupt to 0 again after modifying values of the alarm registers. Once the interrupt flag for the alarm interrupt has been set to 1 and the state has returned to non-matching of the alarm registers and clock counters, the flag will not be set again until there is a further match or the values of the alarm registers are modified again.



Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number .

Figure 32.11 Timing Chart for the Alarm Interrupt (ALM)

(2) Periodic interrupt (PRD)

This interrupt is generated at intervals of 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second. The interrupt interval can be selected through the RCR1.PES[3:0] bits.

(3) Carry interrupt (CUP)

This interrupt is generated when a carry to the second counter/binary counter 0 occurred or a carry to the R64CNT counter occurred during read access to the 64-Hz counter.

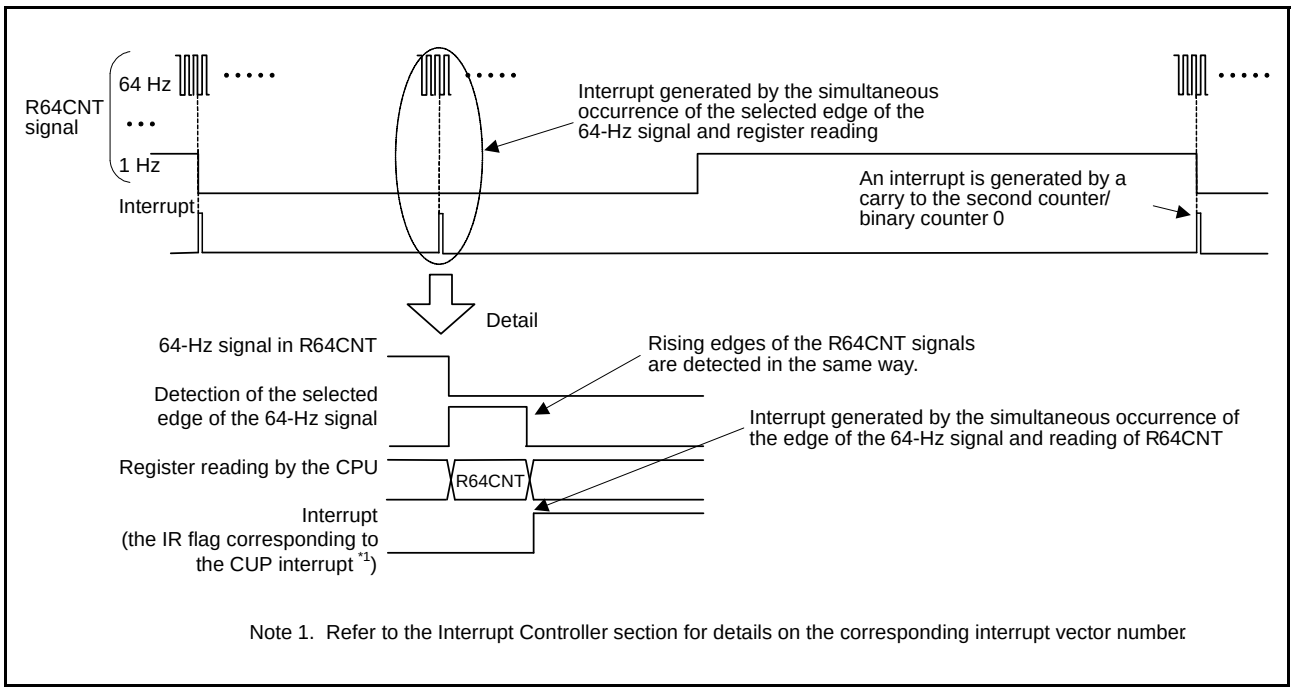


Figure 32.12 Carry Interrupt (CUP) Timing Chart

32.5 Event Link Output

The RTC outputs the following event signals for the event link controller (ELC), and these can be used to initiate operations by other modules selected in advance.

(1) Periodic event output

The periodic event signal is output at the interval selected from among 1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, and 2 seconds by the setting of the RCR1.PES[3:0] bits.

The event generation period immediately after the event generation is selected is not guaranteed.

Note: If event linking from the RTC is to be used, only make the ELC settings after making the RTC settings (initialization, time settings, etc.). Making the RTC settings after the ELC settings can lead to the output of unexpected event signals.

32.5.1 Interrupt Handling and Event Linking

The RTC has a bit to enable or disable periodic interrupts. An interrupt request signal is output for the CPU when an interrupt source is generated while the corresponding enable bit is enabled.

In contrast, an event link output signal is sent to other modules as an event signal via the ELC when an interrupt source is generated, regardless of the setting of the corresponding interrupt enable bit.

Note: Although alarm and periodic interrupts can still be output during software standby or deep software standby, the periodic event signals for the ELC are not output.

32.6 Usage Notes

32.6.1 Register Writing during Counting

The following registers should not be written to during counting (while the RCR2.START bit = 1).

RSECCNT/BCNT0, RMINCNT/BCNT1, RHRCNT/BCNT2, RDAYCNT, RWKCNT/BCNT3, RMONCNT, RYRCNT, RCR1.RTCOS, RCR2.RTCOE, RCR2.HR24, RFRH, RFRL

The counter must be stopped before writing to any of the above registers.

32.6.2 Use of Periodic Interrupts

The procedure for using periodic interrupts is shown in Figure 32.13.
The generation and period of the periodic interrupt can be changed by the setting of the RCR1.PES[3:0] bits. However, since the prescaler, R64CNT, and RSECCNT/BCNT0 are used to generate interrupts, the interrupt period is not guaranteed immediately after setting of the RCR1.PES[3:0] bits.
Furthermore, stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the interrupt period. When the time error adjustment function is used, the interrupt generation period after adjustment is added or subtracted according to the adjustment value.

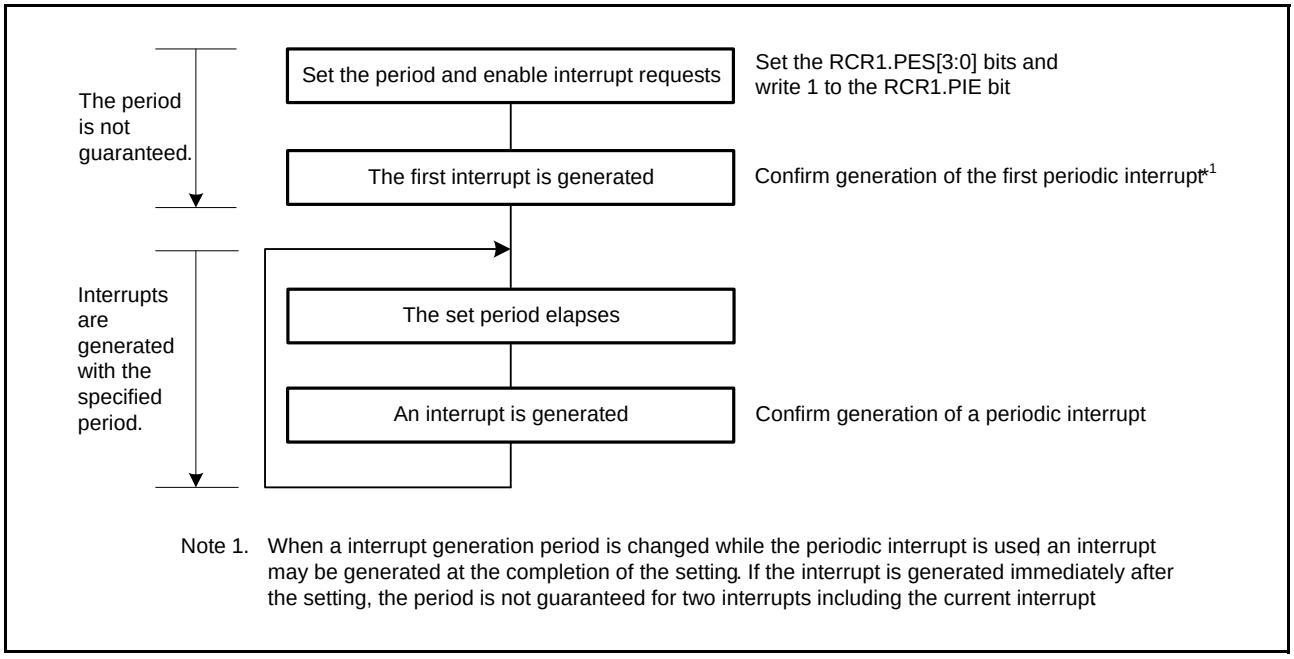


Figure 32.13 Using Periodic Interrupt Function

32.6.3 RTCOUT (1-Hz/64-Hz) Clock Output

Stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the period of RTCOUT (1-Hz/64-Hz) output. When the time error adjustment function is used, the period of RTCOUT (1-Hz/64-Hz) output after adjustment is added or subtracted according to the adjustment value.

32.6.4 Transitions to Low Power Consumption Modes after Setting Registers

A transition to a low power consumption state (software standby mode, deep software standby mode, or battery backup) during writing to or updating of an RTC register might destroy the register's value. After setting a register, confirm that the setting is in place before initiating a transition to a low power consumption state.

32.6.5 Notes When Writing to and Reading from Registers

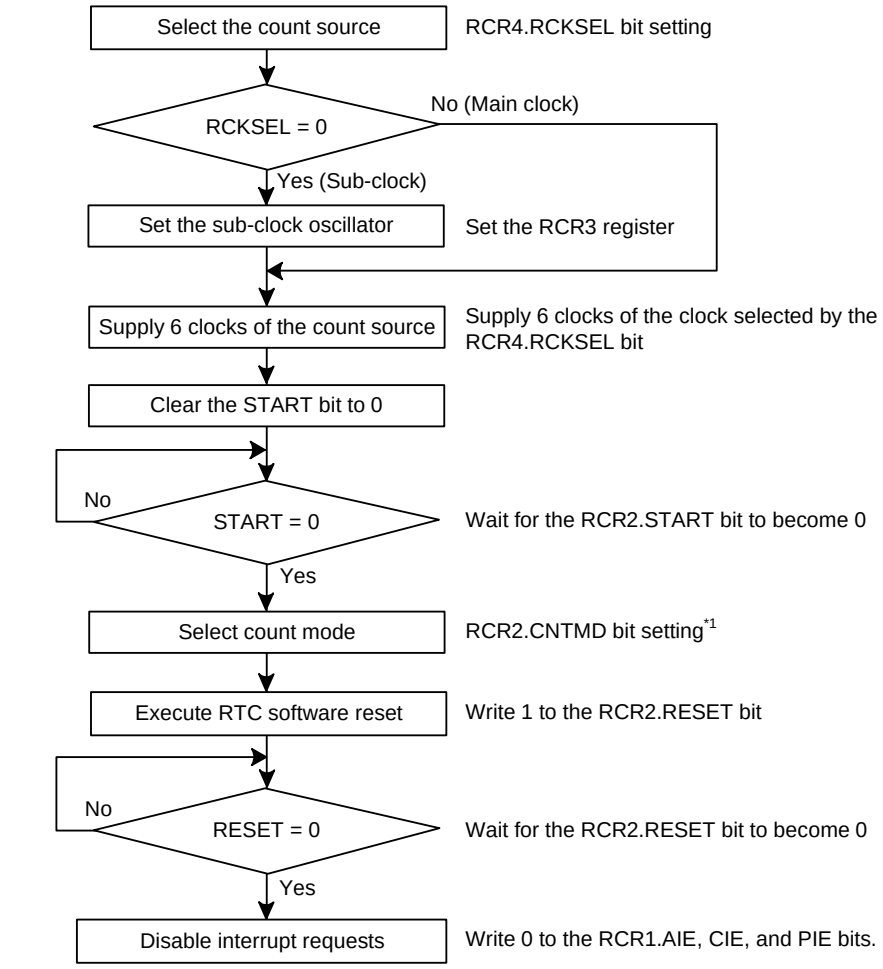
- When reading a counter register such as the second counter after having written to the counter register, follow the procedure in section 32.3.5, Reading 64-Hz Counter and Time.
- The value written to the count registers, alarm registers, year alarm enable register, bits RCR2.AADJE, AADJP, and HR24, RCR3, RCR4, RFRH, or RFRL register is reflected when four read operations are performed after writing.
- The values written to the RCR1.CIE, RTCOS, and RCR2.RTCOE bits can be read immediately after writing.
- To read the value from the timer counter after return from a reset, deep software standby mode, software standby mode, or the battery backup state, wait for1/128 second while the clock is operating (RCR2.START bit = 1).
- After a reset is generated, write to the RTC register when six cycles of the count source have elapsed.

32.6.6 Changing the Count Mode

When changing the count mode (calendar/binary), set the RCR2.START bit to 0, stop counting operation, then start again from the initial setting. For details on initial setting, refer to section 32.3.1, Outline of Initial Settings of Registers after Power On.

32.6.7 Initialization Procedure When the Realtime Clock is Not to be Used

Registers in the RTC are not initialized by a reset. Accordingly, depending on the initial state, the generation of an unintentional interrupt request or operation of the counter may lead to increased power consumption.
For products that do not require a realtime clock, initialize the registers by following the initialization procedure shown in Figure 32.14.
Alternatively, when the sub-clock is not used as the system clock or realtime clock, the counter can be stopped by writing 0 (sub-clock oscillator is selected) to the RCR4.RCKSEL bit and stopping the sub-clock.
When making the setting to stop the sub-clock, write 0 to the RCR3.RTCEN bit and write 1 to the SOSCCR.SOSTP bit.
For details on the setting of the SOSCCR.SOSTP bit, refer to section 9, Clock Generation Circuit.



Note 1. This step is not necessary if the count mode has been set concurrently with setting the START bit to 0.

Figure 32.14 Initialization Procedure

33. Watchdog Timer (WDTA)

The watchdog timer (WDT) is a 14-bit down-counter. It can be used to reset this MCU when the counter underflows because its value cannot be refreshed due to the system being out of control.
In addition, a non-maskable interrupt can be generated by an underflow.
The refresh-permitted period can be set to refresh the counter and used as the condition to detect when the system runs out of control.
In this section, “PCLK” is used to refer to PCLKB.

33.1 Overview

Table 33.1 lists the specifications of the WDT and Figure 33.1 shows a block diagram of the WDT.

Table 33.1 WDT Specifications

Item	Specifications
Count source	Peripheral module clock (PCLK)
Clock division ratio	Divide by 4, 64, 128, 512, 2048, or 8192
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	• Auto-start mode: Counting automatically starts after a reset or after an underflow or refresh error occurs • Register start mode: Counting is started by refresh operation (writing to the WDTRR register)
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • A counter underflows or a refresh error is generated
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Watchdog timer Reset sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt/interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading the counter value	The down-counter value can be read by the WDTSR register.

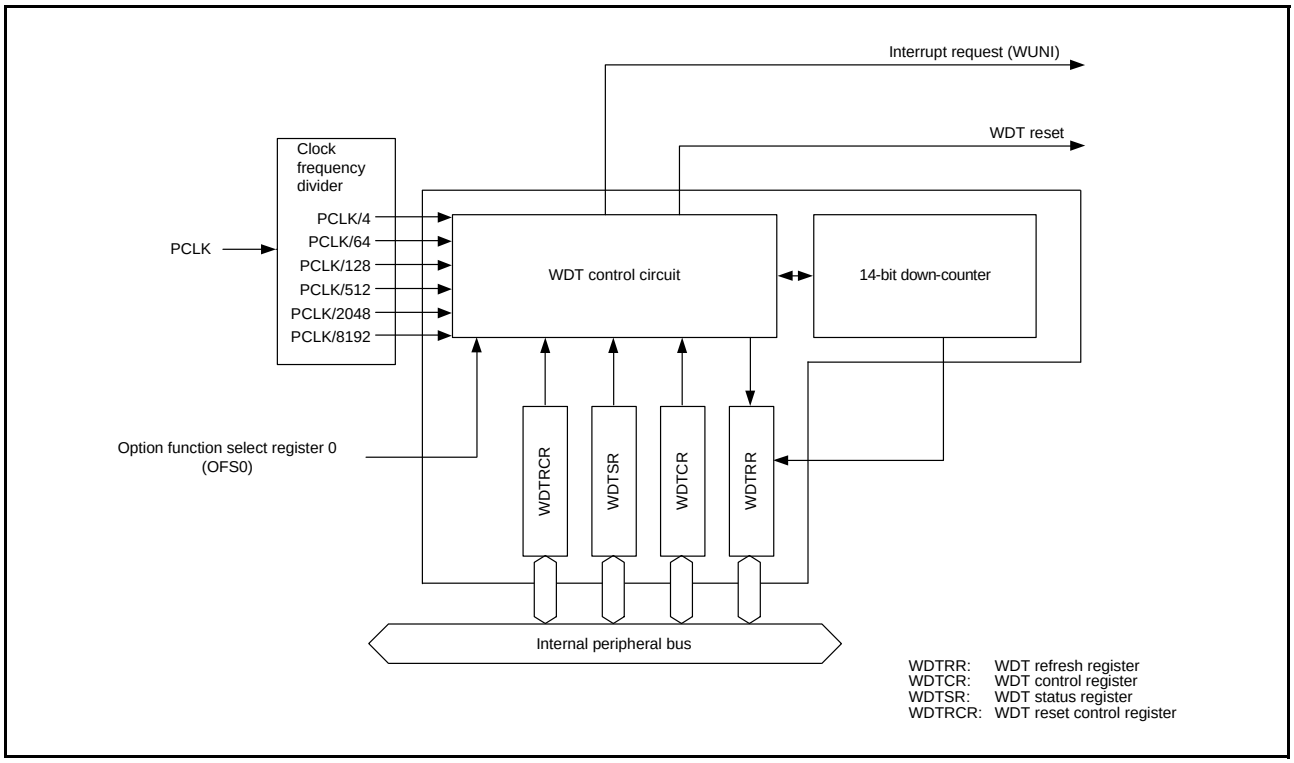
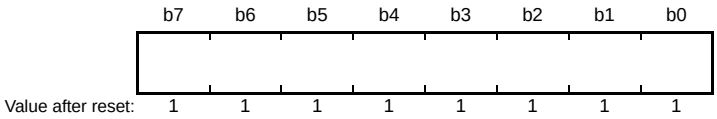


Figure 33.1 WDT Block Diagram

33.2 Register Descriptions

33.2.1 WDT Refresh Register (WDTRR)

Address(es): 0008 8020h



Bit	Description	R/W
b7 to b0	The down-counter is refreshed by writing 00h and then writing FFh to this register	R/W

WDTRR refreshes the down-counter of the WDT.
The down-counter of the WDT is refreshed by writing 00h and then writing FFh to WDTRR (refresh operation) within the refresh-permitted period.
After the down-counter has been refreshed, it starts counting down from the value selected by setting the WDT timeout period select bits (OFS0.WDTTOPS[1:0]) in option function select register 0 in auto-start mode. In register start mode, counting down starts from the value selected by setting the timeout period selection bits (WDTCR.TOPS[1:0]) in the WDT control register.
When 00h is written, the read value is 00h, when a value other than 00h is written, the read value is FFh.
For details of the refresh operation, refer to section 33.3.3, Refresh Operation.

33.2.2 WDT Control Register (WDTCR)

Address(es): 0008 8022h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RPSS[1:0]		—	—	RPES[1:0]		CKS[3:0]				—	—	TOPS[1:0]	
Value after reset:	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOPS[1:0]	Timeout Period Selection	b1 b0 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R/W
b3, b2	—	Reserved	These bits are read as 0 and cannot be modified.	R
b7 to b4	CKS[3:0]	Clock Division Ratio Selection	b7 b4 0 0 0 1: Divide-by-4 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 1 0: Divide-by-512 0 1 1 1: Divide-by-2048 1 0 0 0: Divide-by-8192 Setting other than above are prohibited.	R/W
b9, b8	RPES[1:0]	Window End Position Selection	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified)	R/W
b11, b10	—	Reserved	These bits are read as 0 and cannot be modified.	R
b13, b12	RPSS[1:0]	Window Start Position Selection	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified)	R/W
b15, b14	—	Reserved	These bits are read as 0 and cannot be modified.	R

There are some restrictions on writing to the WDTCR register. For details, refer to section 33.3.2, Control over Writing to the WDTCR and WDTRCR Registers.
In auto-start mode, the settings in the WDTCR register are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the WDTCR register can also be made in OFS0 register. For details, refer to section 33.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

TOPS[1:0] Bits (Timeout Period Selection)

These bits select the timeout period (period until the down-counter underflows) from among 1024, 4096, 8192, and 16384 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.
After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of PCLK cycles) until the counter underflows.
Relations between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of PCLK cycles are listed in Table 33.2.

Table 33.2 Timeout Period Settings

CKS[3:0] Bits				TOPS[1:0] Bits		Clock Division Ratio	Timeout Period (Number of Cycles)	Cycles of PCLK Clock
b7	b6	b5	b4	b1	b0			
0	0	0	1	0	0	Divide-by-4	1024	4096
				0	1		4096	16384
				1	0		8192	32768
				1	1		16384	65536
0	1	0	0	0	0	Divide-by-64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	Divide-by-128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	1	0	0	0	Divide-by-512	1024	524288
				0	1		4096	2097152
				1	0		8192	4194304
				1	1		16384	8388608
0	1	1	1	0	0	Divide-by-2048	1024	2097152
				0	1		4096	8388608
				1	0		8192	16777216
				1	1		16384	33554432
1	0	0	0	0	0	Divide-by-8192	1024	8388608
				0	1		4096	33554432
				1	0		8192	67108864
				1	1		16384	134217728

CKS[3:0] Bits (Clock Division Ratio Selection)

These bits specify the division ration of the clock used for the down-counter. The division ration can be selected from among the peripheral module clock (PCLK) divided by 4, 64, 128, 512, 2048, and 8192. Combined with the TOPS[1:0] bit setting, a count period between 4,096 and 134,217,728 cycles of the PCLK clock can be selected for the WDT.

RPES[1:0] Bits (Window End Position Selection)

These bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. The selected window end position should be a value smaller than the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.

RPSS[1:0] Bits (Window Start Position Selection)

These bits specify the window start position that indicates the refresh-permitted period. 25%, 50%, 75%, or 100% of the timeout period can be selected for the window end position. The window start position should be set to a value greater the window end position. If the window start position is set to a value smaller than or equal to the window end position, the window end position is set to 0%.

Table 33.3 lists the counter values for the window start and end positions and Figure 33.2 shows the refresh-permitted period set by the RPSS[1:0], RPES[1:0], and TOPS[1:0] bits.

Table 33.3 Relationship between Timeout Period and Window Start and End Counter Values

TOPS[1:0] Bits		Timeout Period		Window Start and End Counter Value			
		Cycles	Counter Value	100%	75%	50%	25%
0	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
0	1	4096	0FFFh	0FFFh	0BFFh	07FFh	03FFh
1	0	8192	1FFFh	1FFFh	17FFh	0FFFh	07FFh
1	1	16384	3FFFh	3FFFh	2FFFh	1FFFh	0FFFh

RPSS[1:0] Bits		RPES[1:0] Bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75

Note: If window end setting ≥ window start setting, the window end setting is set to 0%.

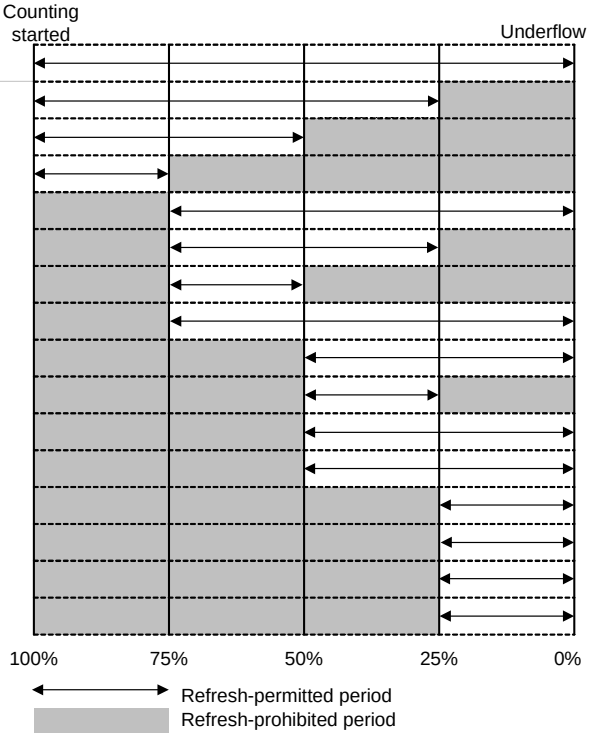


Figure 33.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

33.2.3 WDT Status Register (WDTSR)

Address(es): 0008 8024h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REFEF	UNDFF	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CNTVAL[13:0]	Down-Counter Value	Value counted by the down-counter	R
b14	UNDFF	Underflow Flag	0: No underflow occurred 1: Underflow occurred	R(/W) *1
b15	REFEF	Refresh Error Flag	0: No refresh error occurred 1: Refresh error occurred	R(/W) *1

Note 1. Only 0 can be written to clear the flag.

CNTVAL[13:0] Bits (Down-Counter Value)

Read these bits to confirm the value of the down-counter, but note that the read value may differ from the actual count by a value of one count.

UNDFF Flag (Underflow Flag)

Read this flag to confirm whether or not an underflow has occurred in the down-counter.
The value 1 indicates that the down-counter has underflowed. The value 0 indicates that the down-counter has not underflowed.
Write 0 to the UNDFF flag to set the value to 0. Writing 1 has no effect.

REFEF Flag (Refresh Error Flag)

Read this flag to confirm whether or not a refresh error (performing a refresh operation during a refresh-prohibited period) has occurred.
The value 1 indicates that a refresh error has occurred. The value 0 indicates that no refresh error has occurred.
Write 0 to the REFEF flag to set the value to 0. Writing 1 has no effect.

33.2.4 WDT Reset Control Register (WDTRCR)

Address(es): 0008 8026h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTIRQS	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0 and cannot be modified.	R
b7	RSTIRQS	Reset Interrupt Request Selection	0: Non-maskable interrupt request or interrupt request output is enabled 1: Reset output is enabled	R/W

There are some restrictions on writing to the WDTRCR register. For details, refer to section 33.3.2, Control over Writing to the WDTCR and WDTRCR Registers.

In auto-start mode, the WDTRCR register settings are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the WDTCR register can also be made in the OFS0 register. For details, refer to section 33.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

33.2.5 Option Function Select Register 0 (OFS0)

For details on the OFS0 register, refer to section 33.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

33.3 Operation

33.3.1 Count Operation in Each Start Mode

The WDT has two start modes: auto-start mode, in which counting automatically starts after release from the reset state, and register start mode, in which counting is started by refreshing (writing to the register).
In auto-start mode, counting automatically starts after release from the reset state in accordance with the settings in option function select register 0 (OFS0) in the ROM.
In register start mode, counting is started by refreshing (writing to the register) after the respective registers are set after release from the reset state.
Select auto-start mode or register start mode by setting the WDT start mode select bit (OFS0.WDTSTRT) in the OFS0 register.
When the auto-start mode is selected, the settings in the WDT control register (WDTCR) and WDT reset control register (WDTRCR) are disabled, and the settings in the OFS0 register are enabled.
On the other hand, when the register start mode is selected, the setting of the OFS0 register is disabled, and the settings of the WDT control register (WDTCR) and WDT reset control register (WDTRCR) are enabled.

33.3.1.1 Register Start Mode

When the WDT start mode select bit (OFS0.WDTSTRT) is 1, register start mode is selected, and the WDT control register (WDTCR) and WDT reset control register (WDTRCR) are enabled.
After the reset state is released, set the clock division ratio, window start and end positions, and timeout period in the WDTCR register, and the reset output or interrupt request output in the WDTRCR register. Then, refresh the down-counter to start counting down from the value set by the timeout period selection bits (WDTCR.TOPS[1:0]).
Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The WDT does not output the reset signal as long as this continues. However, if the down-counter underflows because the down-counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the WDT outputs a reset signal or a non-maskable interrupt request/interrupt request (WUNI). Reset output or interrupt request output can be selected by setting the WDT reset interrupt request selection bit (WDTRCR.RSTIRQS).

Figure 33.3 shows an example of operation under the following conditions.

- Register start mode (OFS0.WDTSTRT = 1)
- Reset output is enabled (WDTRCR.RSTIRQS = 1)
- The window start position is 75% (WDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (WDTCR.RPES[1:0] = 10b)

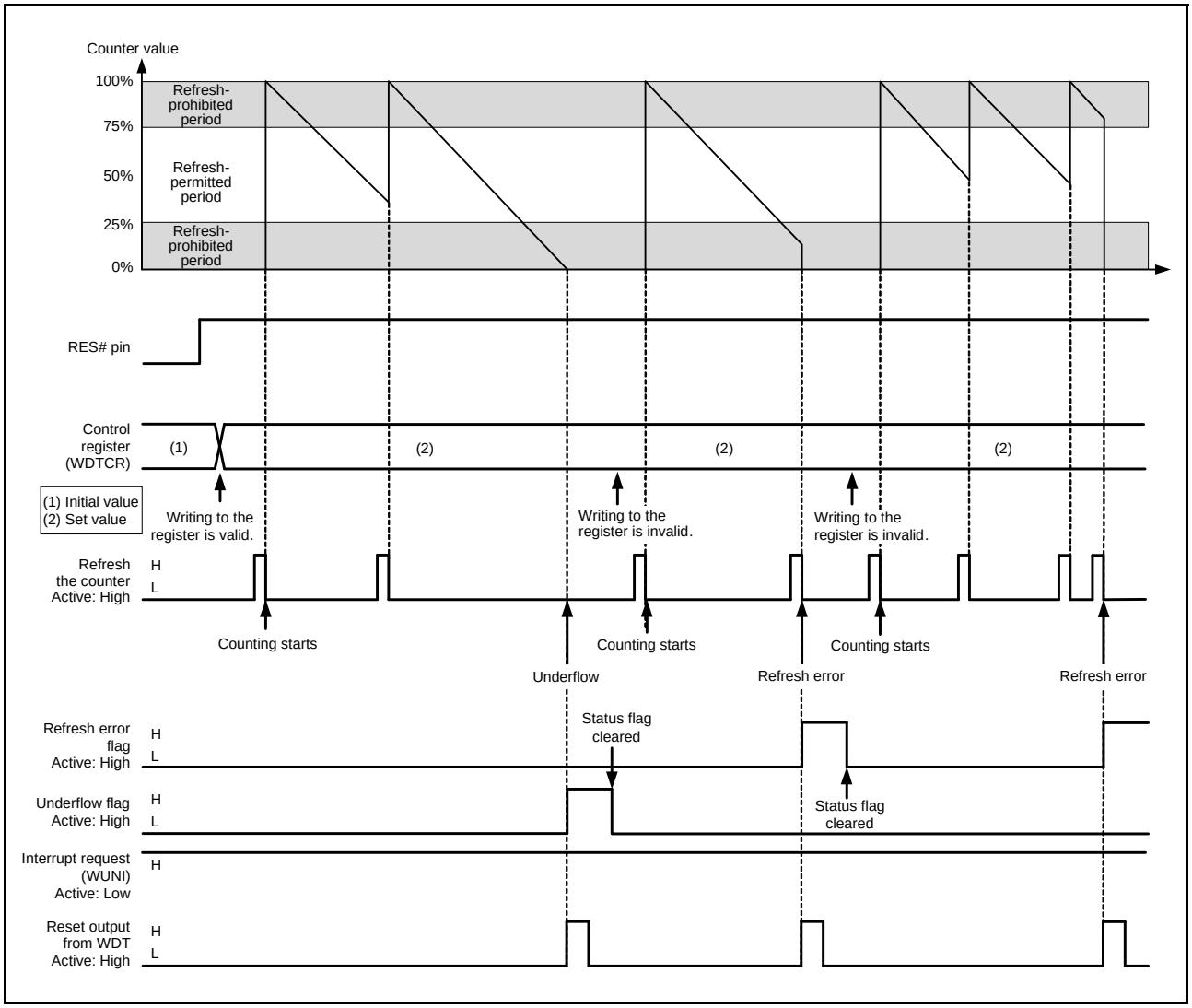


Figure 33.3 Operation Example in Register Start Mode

33.3.1.2 Auto-Start Mode

When the WDT start mode select bit (OFS0.WDTSTRT) in option function select register 0 (OFS0) is 0, auto-start mode is selected, the WDT control register (WDTCR) and WDT reset control register (WDTRCR) are disabled, and the settings in the OFS0 register are enabled.
Within the reset state, the setting values (clock division ratio, window start and end positions, timeout period, and reset output or interrupt request) of option function select register 0 (OFS0) are set in the WDT registers.
When the reset state is released, the down-counter automatically starts counting down from the value set by the WDT timeout period select bits (OFS0.WDTTOPS[1:0]).
After that, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The WDT does not output the reset signal as long as this continues.
However, if the down-counter underflows because refreshing of the down-counter is not possible due to the program having entered crashed execution or if a refresh error occurs due to refreshing outside the refresh-permitted period, the WDT outputs the reset signal or non-maskable interrupt request/interrupt request (WUNI).
After the reset signal or non-maskable interrupt request/interrupt request is output of for one cycle of counting, the value of the timeout period is set in the down-counter counting is restarted.
Reset output or interrupt request output can be selected by setting the WDT reset interrupt request select bit (OFS0.WDTRSTIRQS).

Figure 33.4 shows an example of operation (non-maskable interrupt) under the following conditions.

- Auto start mode (OFS0.WDTSTRT = 0)
- Non-maskable interrupt request output is enabled (OFS0.WDTRSTIRQS = 0)
- The window start position is 75% (WDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (WDTCR.RPES[1:0] = 10b)

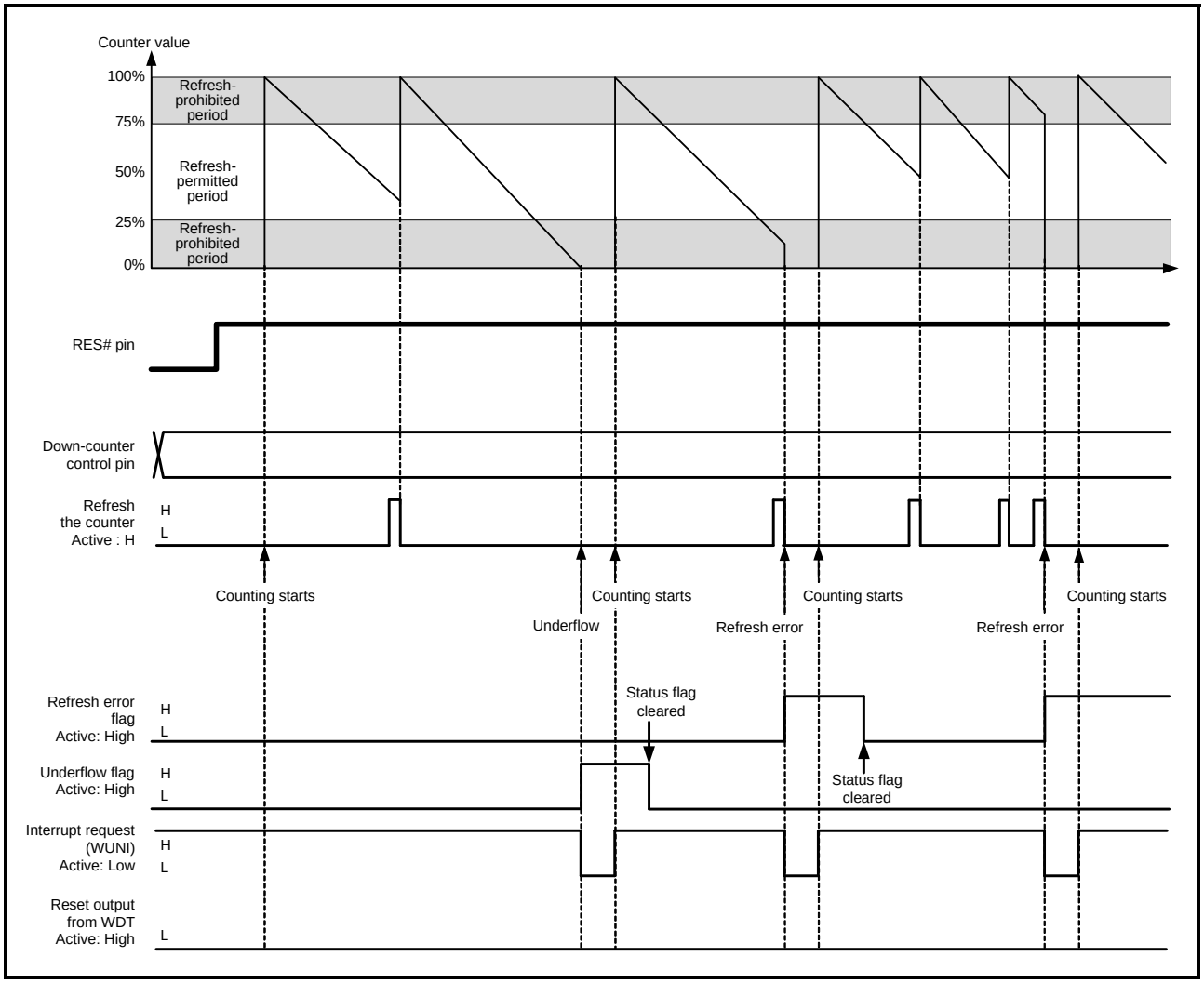


Figure 33.4 Operation Example in Auto-Start Mode

33.3.2 Control over Writing to the WDTCR and WDTRCR Registers

Writing to the WDT control register (WDTCR) or WDT reset control register (WDTRCR) is only possible once between the release from the reset state and the first refresh operation.
After a refresh operation (counting starts) or by writing to WDTCR or WDTRCR, the protection signal in the WDT becomes 1 to protect WDTCR and WDTRCR against subsequent attempts at writing.
This protection is released by the reset source of the WDT. With other reset sources, the protection is not released.
Figure 33.5 shows control waveforms produced in response to writing to the WDTCR.

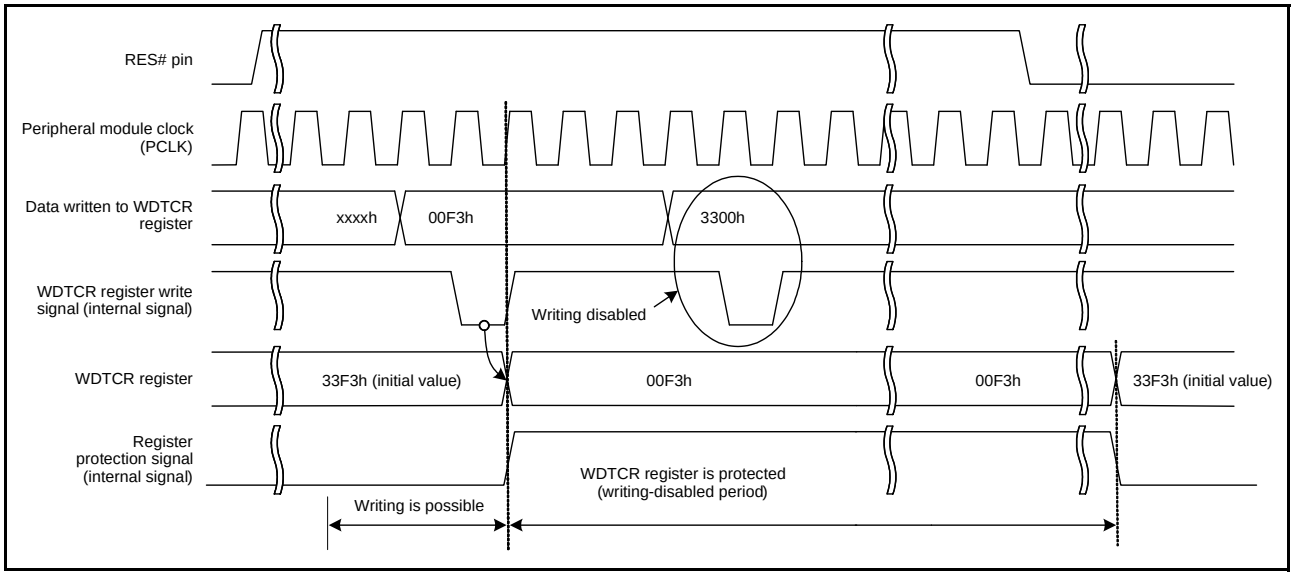


Figure 33.5 Control Waveforms Produced in Response to Writing to the WDTCR Register

33.3.3 Refresh Operation

The down-counter is refreshed by writing the values 00h and then FFh to the WDT refresh register (WDTRR). If a value other than FFh is written after 00h, the down-counter is not refreshed. After such invalid writing, correct refreshing is performed by again writing to 00h and then FFh to the WDTRR register.
Even if a register other than WDTRR is accessed or WDTRR is read between writing 00h and writing FFh to WDTRR, correct refreshing will be done.
Writing to refresh the counter must be performed within the refresh-permitted period. Whether writing is done within the refresh-permitted period is determined when writing FFh. For this reason, correct refreshing will be done even if 00h is written outside the refresh-permitted period.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h → FFh
- 00h (n–1-th time) → 00h (nth time) → FFh
- 00h → access to another register or read from WDTRR → FFh

[Sample sequences of writing that are not valid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h→ AAh (00h and a value other than FFh) → FFh

After FFh is written to the WDT refresh register (WDTRR), refreshing the down-counter requires up to four cycles of the signal for counting. Therefore, writing FFh to the WDTRR should be completed four-count cycles before the down-counter underflows.

Figure 33.6 shows the WDT refresh-operation waveforms when the clock division ratio = PCLK/64.

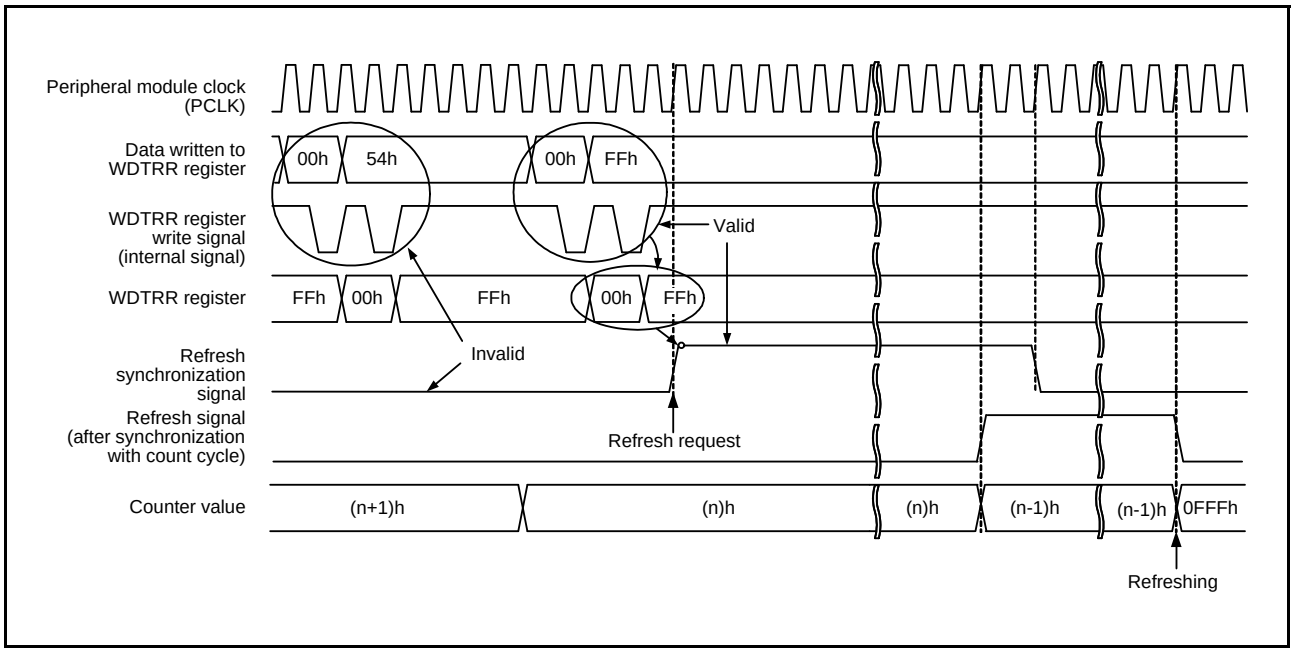


Figure 33.6 WDT Refresh Operation Waveforms (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

33.3.4 Reset Output

When the reset interrupt selection bit (WDTRCR.RSTIRQS) is set to 1 in register start mode or when the WDT reset interrupt request select bit (OFS0.WDTRSTIRQS) in option function select register 0 (OFS0) is set to 1 in auto-start mode, a reset signal is output for one-count cycle when an underflow in the down-counter or a refresh error occurs.
In register start mode, the down-counter is initialized (all bits set to 0) and stopped in that state after output of the reset signal. After the reset state is released and the program is restarted, the counter is set up again and counting down is started by refreshing.
In auto-start mode, counting down automatically starts after the reset state is released.

33.3.5 Interrupt Source

When the reset interrupt selection bit (WDTRCR.RSTIRQS) is set to 0 in register start mode or when the WDT reset interrupt request select bit (OFS0.WDTRSTIRQS) in option function select register 0 (OFS0) is set to 0 in auto-start mode, an interrupt (WUNI) signal is generated when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt and an interrupt. For details, refer to section 15, Interrupt Controller (ICUA).

Table 33.4 WDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
WUNI	Down-counter underflow Refresh error	Not possible	Not possible

33.3.6 Reading the Down-Counter Value

The WDT stores the counter value in the down-counter value (WDTSR.CNTVAL[13:0]) bits of the WDT status register. Thus, the counter value can be checked through the WDTSR.CNTVAL[13:0] bits.

Figure 33.7 shows the processing for reading the WDT down-counter value when the clock division ratio = PCLK/64.

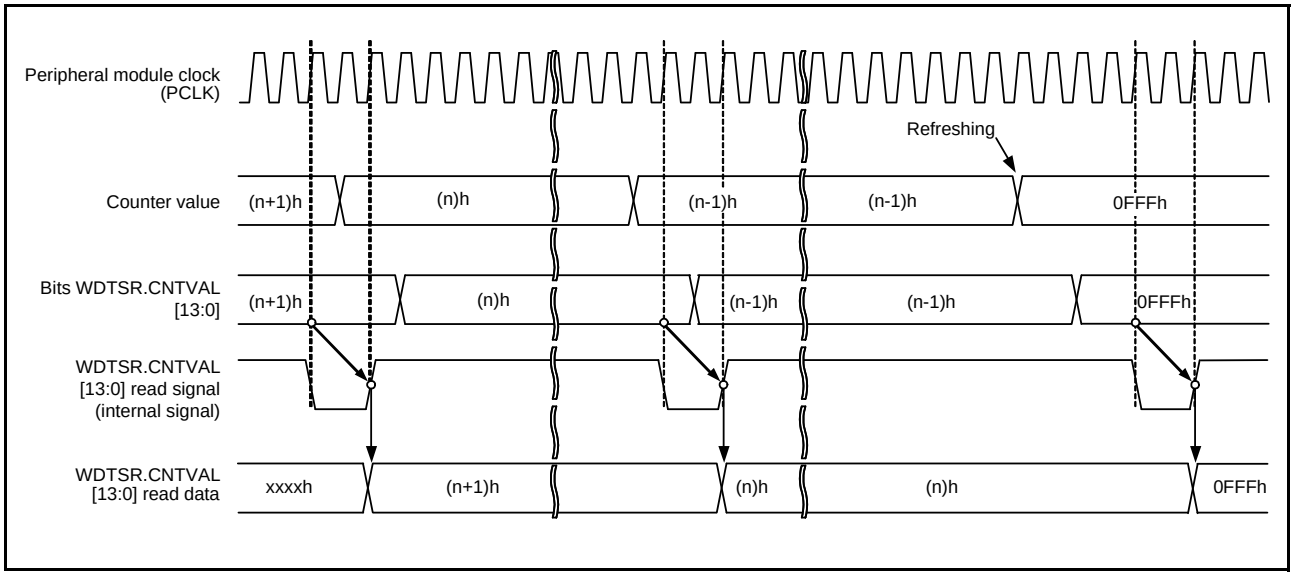


Figure 33.7 Processing for Reading WDT Down-Counter Value (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

33.3.7 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers

Table 33.5 lists the correspondence between option function select register 0 (OFS0) used in auto-start mode and the registers used in register start mode.

Do not change the OFS0 register setting during WDT operation.

For details on option function select register 0 (OFS0), refer to section 7.2.3, Option Function Select Register 0 (OFS0).

Table 33.5 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers

Target of Control	Function	OFS0 Register (Enabled in Auto-Start Mode) OFS0.WDTSTRT = 0	WDT Registers (Enabled in Register Start Mode) OFS0.WDTSTRT = 1
Down-counter	Timeout period selection	OFS0.WDTTOPS[1:0]	WDTCR.TOPS[1:0]
	Clock division ratio selection	OFS0.WDTCKS[3:0]	WDTCR.CKS[3:0]
	Window start position selection	OFS0.WDTRPSS[1:0]	WDTCR.RPSS[1:0]
	Window end position selection	OFS0.WDTRPES[1:0]	WDTCR.RPES[1:0]
Reset output or interrupt request output	Reset output or interrupt request output selection	OFS0.WDTRSTIRQS	WDTRCR.RSTIRQS

34. Independent Watchdog Timer (IWDTa)

In this section, “PCLK” is used to refer to PCLKB.

34.1 Overview

The independent watchdog timer (IWDT) can be used to detect programs being out of control.
The user can detect when a program runs out of control if an underflow occurs, by creating a program that refreshes the IWDT counter before it underflows.
The functions of the IWDT are different from those of the WDT in the following respects.

- The divided IWDT-dedicated low-speed clock (IWDTCLK) is used as the count source (not affected by the PCLK).
- When making a transition to sleep mode, software standby mode, deep software standby mode, or all-module clock stop mode, the IWDTCSTPR.SLCSTP bit or the OFS0.IWDTSLCSTP bit can be used to select whether to stop the counter or not.

Table 34.1 lists the specifications of the IWDT and Figure 34.1 shows a block diagram of the IWDT.

Table 34.1 IWDT Specifications

Item	Description
Count source*1	IWDT-dedicated clock (IWDTCLK)
Clock divide ratio	Divide by 1, 16, 32, 64, 128, or 256
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	• Counting automatically starts after a reset (auto-start mode) • Counting is started (register start mode) by refreshing the counter (writing 00h and then FFh to the IWDTRR register).
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • A counter underflows or a refresh error occurs Counting restarts (In auto-start mode, counting automatically restarts after a reset or after a non-maskable interrupt request/interrupt request is output. In register start mode, counting restarts after refreshing.)
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Reset output sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt/ interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading the counter value	The down-counter value can be read by the IWDTSR register.
Event link function (output)	• Down-counter underflow event output • Refresh error event output
Output signal (internal signal)	• Reset output • Interrupt request output • Sleep mode count stop control output
Auto-start mode (controlled by option function select register 0 (OFS0))	• Selecting the clock frequency divide ratio after a reset (OFS0.IWDTCKS[3:0] bits) • Selecting the timeout period of the independent watchdog timer (OFS0.IWDTTOPS[1:0] bits) • Selecting the window start position in the independent watchdog timer (OFS0.IWDTRPSS[1:0] bits) • Selecting the window end position in the independent watchdog timer (OFS0.IWDTRPES[1:0] bits) • Selecting the reset output or interrupt request output (OFS0.IWDTRSTIRQS bit) • Selecting the down-count stop function at transition to sleep mode, software standby mode, deep software standby mode, or all-module clock stop mode (OFS0.IWDTSLCSTP bit)
Register start mode (controlled by the IWDT registers)	• Selecting the clock frequency divide ratio after refreshing (IWDTCR.CKS[3:0] bits) • Selecting the timeout period of the independent watchdog timer (IWDTCR.TOPS[1:0] bits) • Selecting the window start position in the independent watchdog timer (IWDTCR.RPSS[1:0] bits) • Selecting the window end position in the independent watchdog timer (IWDTCR.RPES[1:0] bits) • Selecting the reset output or interrupt request output (IWDTRCR.RSTIRQS bit) • Selecting the down-count stop function at transition to sleep mode, software standby mode, deep software standby mode, or all-module clock stop mode (IWDTCSTPR.SLCSTP bit)

Note 1. Satisfy the frequency of the peripheral module clock (PCLK) ≥ 4 × (the frequency of the count source after divide).

To use the IWDT, the IWDT-dedicated clock (IWDTCLK) should be supplied so that the IWDT operates even if the peripheral module clock (PCLK) stops. The bus interface and registers operate with PCLK, and the 14-bit counter and control circuits operate with IWDTCLK.
Figure 34.1 is a block diagram of the IWDT.

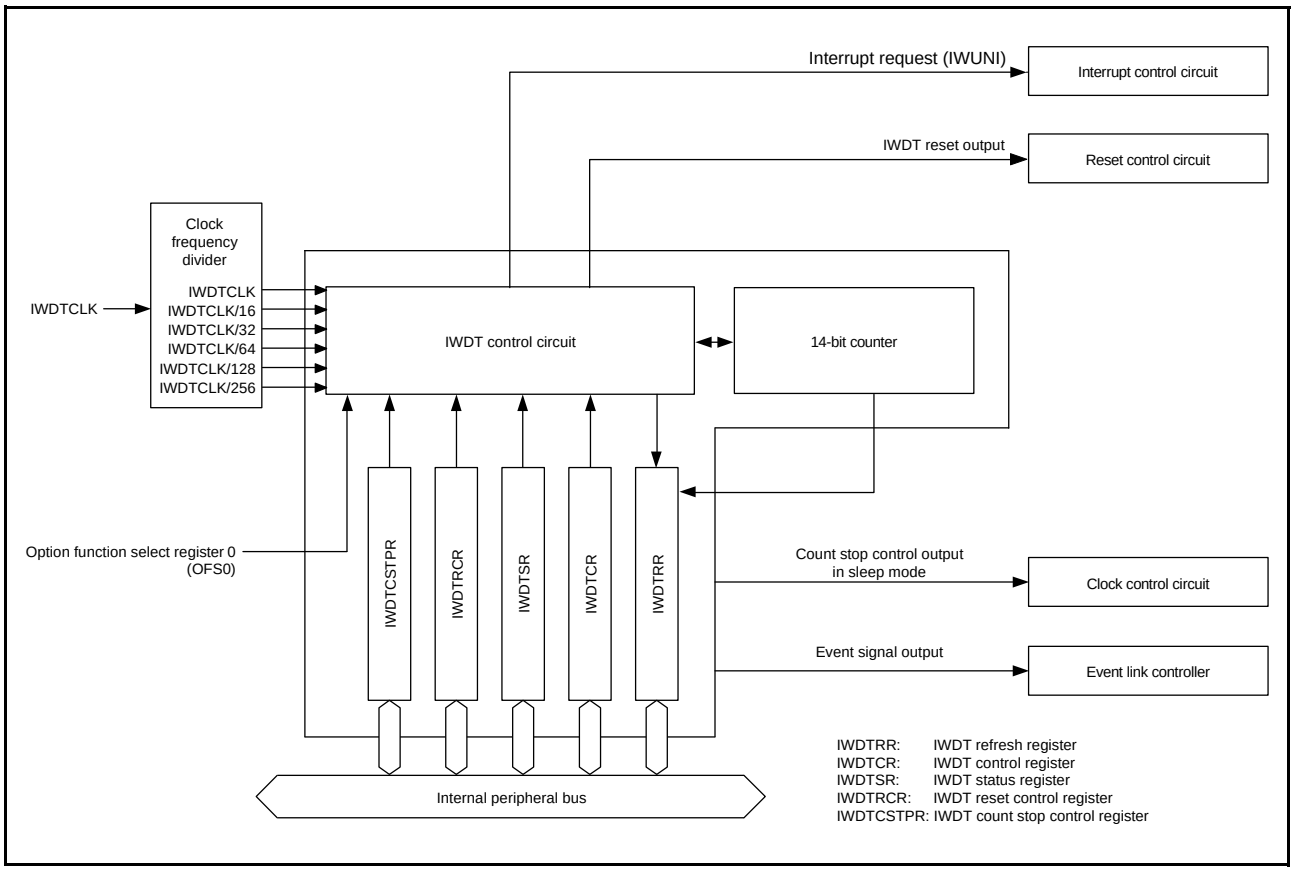
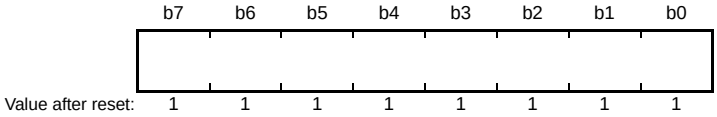


Figure 34.1 IWDT Block Diagram

34.2 Register Descriptions

34.2.1 IWDT Refresh Register (IWDTRR)

Address(es): IWDT.IWDTRR 0008 8030h



Bit	Description	R/W
b7 to b0	The counter is refreshed by writing 00h and then writing FFh to this register.	R/W

The IWDTRR register refreshes the counter of the IWDT.
The counter of the IWDT is refreshed by writing 00h and then writing FFh to the IWDTRR register (refresh operation) within the refresh-permitted period.
After the counter has been refreshed, it starts counting down from the value selected by the IWDT timeout period select bits (OFS0.IWDTTOPS[1:0]) in option function select register 0 (OFS0) in auto-start mode. In register start mode, counting down starts from the value selected by setting the timeout period select bits (TOPS[1:0]) in the IWDT control register (IWDTCR) in the first refresh operation after release from the reset state.
When 00h is written, the read value is 00h. When a value other than 00h is written, the read value is FFh.
For details of the refresh operation, refer to section 34.3.3, Refresh Operation.

34.2.2 IWDT Control Register (IWDTCR)

Address(es): IWDT.IWDTCR 0008 8032h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RPSS[1:0]		—	—	RPES[1:0]		CKS[3:0]				—	—	TOPS[1:0]	
Value after reset:	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOPS[1:0]	Timeout Period Select	b1 b0 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R/W
b3, b2	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7 to b4	CKS[3:0]	Clock Divide Ratio Select	b7 b4 0 0 0 0: No division 0 0 1 0: Divide-by-16 0 0 1 1: Divide-by-32 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 0 1: Divide-by-256 Other settings are prohibited.	R/W
b9, b8	RPES[1:0]	Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified.)	R/W
b11, b10	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b13, b12	RPSS[1:0]	Window Start Position Select	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified.)	R/W
b15, b14	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

There are some restrictions on writing to the IWDTCR register. For details, refer to section 34.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the settings in the IWDTCR register are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the IWDTCR register can also be made in option function select register 0 (OFS0). For details, refer to section 34.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

TOPS[1:0] Bits (Timeout Period Select)

These bits select the timeout period (period until the counter underflows) from among 1024, 4096, 8196, or 16384 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.
After the counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of IWDTCLK cycles) until the counter underflows.
Relations between the CKS[3:0] and TOPS[1:0] bit setting, the timeout period, and the number of IWDTCLK cycles are listed in Table 34.2.

Table 34.2 Settings and Timeout Periods

CKS[3:0] Bits				TOPS[1:0] Bits				
b7	b6	b5	b4	b1	b0	Clock Divide Ratio	Timeout Period (Number of Cycles)	Cycles of IWDTCLK
0	0	0	0	0	0	No division	1024	1024
				0	1		4096	4096
				1	0		8192	8192
				1	1		16384	16384
0	0	1	0	0	0	Divide-by-16	1024	16384
				0	1		4096	65536
				1	0		8192	131072
				1	1		16384	262144
0	0	1	1	0	0	Divide-by-32	1024	32768
				0	1		4096	131072
				1	0		8192	262144
				1	1		16384	524288
0	1	0	0	0	0	Divide-by-64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	Divide-by-128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	0	1	0	0	Divide-by-256	1024	262144
				0	1		4096	1048576
				1	0		8192	2097152
				1	1		16384	4194304

CKS[3:0] Bits (Clock Divide Ratio Select)

These bits select the IWDTCLK clock divide ratio from among divide-by 1, 16, 32, 64, 128, and 256. Combination with the TOPS[1:0] bit setting, a count period between 1024 and 4194304 cycles of the IWDTCLK clock can be selected for the IWDT.

RPES[1:0] Bits (Window End Position Select)

These bits select 75%, 50%, 25% or 0% of the count period for the window end position of the counter. The window end position should be a value smaller than the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.
The counter values for the window start and end positions selected by setting the RPSS[1:0] and RPES[1:0] bits change depending on the TOPS[1:0] bit setting.
Table 34.3 lists the counter values for the window start and end positions corresponding to TOPS[1:0] bit values.

Table 34.3 Relationship between Timeout Period and Window Start and End Counter Values

TOPS[1:0] Bits		Timeout Period		Window Start and End Counter Value			
b1	b0	Cycles	Counter Value	100%	75%	50%	25%
0	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
0	1	4096	0FFFh	0FFFh	0BFFh	07FFh	03FFh
1	0	8192	1FFFh	1FFFh	17FFh	0FFFh	07FFh
1	1	16384	3FFFh	3FFFh	2FFFh	1FFFh	0FFFh

RPSS[1:0] Bits (Window Start Position Select)

These bits select a counter window start position from 100%, 75%, 50%, or 25% of the count period (100% when the count starts and 0% when the counter underflows). The interval between the window start position and window end position is the refresh-permitted period and the other periods are refresh-prohibited periods.
Figure 34.2 shows the relationship between of the RPSS[1:0] and RPES[1:0] bit setting and the refresh-permitted and refresh-prohibited periods.

RPSS[1:0] Bits		RPES[1:0] Bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75

Note: If window end setting is greater than or equal to window start setting, the window end setting is set to 0%.

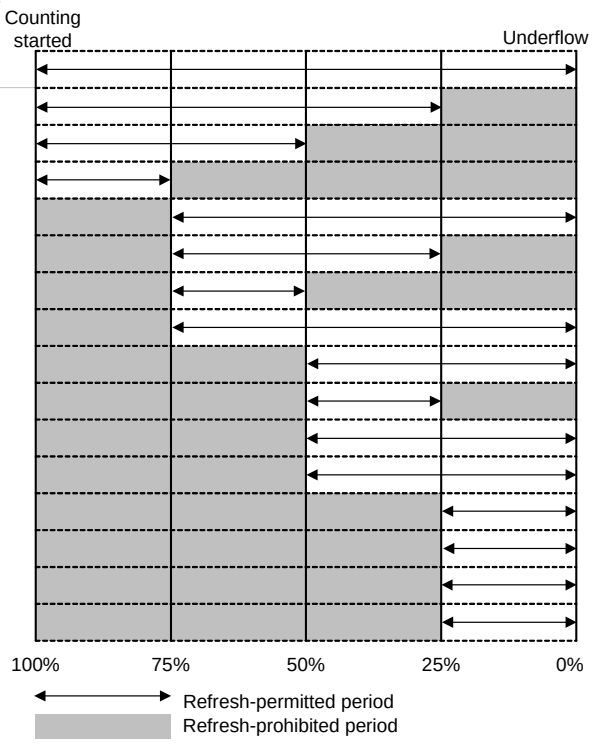


Figure 34.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

34.2.3 IWDT Status Register (IWDTSR)

Address(es): IWDT.IWDTSR 0008 8034h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REFEF	UNDFF	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CNTVAL[13:0]	Counter Value	Value counted by the counter	R
b14	UNDFF	Underflow Flag	0: No underflow occurred 1: Underflow occurred	R/(W) *1
b15	REFEF	Refresh Error Flag	0: No refresh error occurred 1: Refresh error occurred	R/(W) *1

Note 1. Only 0 can be written to clear the flag.

The IWDTSR register is initialized by the reset source of the IWDT. The IWDTSR register is not initialized by other reset sources.

CNTVAL[13:0] Bits (Counter Value)

These bits are used to confirm the counter value of the counter, but note that the read value may differ from the actual count by a value of one count.

UNDFF Flag (Underflow Flag)

This bit is used to confirm whether or not an underflow has occurred in the counter.
The value 1 indicates that the counter has underflowed. The value 0 indicates that the counter has not underflowed.
Write 0 to the UNDFF flag to set the value to 0. Writing 1 has no effect.

REFEF Flag (Refresh Error Flag)

This bit is used to confirm whether or not a refresh error (performing a refresh operation during a refresh-prohibited period).
The value 1 indicates that a refresh error has occurred. The value 0 indicates that no refresh error has occurred.
Write 0 to the REFEF flag to set the value to 0. Writing 1 has no effect.

34.2.4 IWDT Reset Control Register (IWDTRCR)

Address(es): IWDT.IWDTRCR 0008 8036h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTIRQS	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	RSTIRQS	Reset Interrupt Request Select	0: Non-maskable interrupt request or interrupt request output is enabled. 1: Reset output is enabled.	R/W

There are some restrictions on writing to the IWDTRCR register. For details, refer to section 34.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the IWDTRCR register setting are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting mode to the IWDTRCR register can also be made in option function select register 0. For details, refer to section 34.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

34.2.5 IWDT Count Stop Control Register (IWDTCSTPR)

Address(es): IWDT.IWDTCSTPR 0008 8038h

	b7	b6	b5	b4	b3	b2	b1	b0
	SLCSTP	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	SLCSTP	Sleep Mode Count Stop Control	0: Count stop is disabled. 1: Count is stopped at a transition to sleep mode, software standby mode, deep software standby mode, or all-module clock stop mode.	R/W

The IWDTCSTPR register controls whether to stop the IWDT counter in a low power consumption state. There are some restrictions on writing to the IWDTCSTPR register. For details, refer to section 34.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the IWDTCSTPR register setting are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting mode to the IWDTCSTPR register can also be made in option function select register 0 (OFS0). For details, refer to section 34.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

SLCSTP Bit (Sleep Mode Count Stop Control)

This bit selects whether to stop counting at a transition to sleep mode, software standby mode, deep software standby mode, or all-module clock stop mode.

34.2.6 Option Function Select Register 0 (OFS0)

For option function select register 0 (OFS0), refer to section 34.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

34.3 Operation

34.3.1 Count Operation in Each Start Mode

Select the IWDT start mode by setting the IWDT start mode select bit (OFS0.IWDTSTRT) in option function select register 0.
When the OFS0.IWDTSTRT bit is 1 (register start mode), the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR) are enabled, and counting is started by refreshing (writing) the IWDT refresh register (IWDTRR). When the OFS0.IWDTSTRT bit is 0 (auto-start mode), the setting of option function select register 0 (OFS0) is enabled, and counting automatically starts after reset.

34.3.1.1 Register Start Mode

When the IWDT start mode select bit (OFS0.IWDTSTRT) in option function select register 0 is 1, register start mode is selected, and the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR) are enabled.
After the reset state is released, set the clock divide ratio, window start and end positions, and timeout period in the IWDTCR register, the reset output or interrupt request output in the IWDTRCR register, and the counter stop control at transitions to low power consumption states in the IWDTCSTPR register. Then refresh the counter to start counting down from the value selected by setting the timeout period select bits (IWDTCR.TOPS[1:0]).
Thereafter, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the counter underflows because the counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the IWDT outputs a reset signal or a non-maskable interrupt request/interrupt request (IWUNI). Set the IWDT reset interrupt request select bit (IWDTRCR.RSTIRQS) to select either reset output or interrupt request output.

Figure 34.3 shows an example of operation under the following conditions.

- The IWDT start mode select bit (OFS0.IWDTSTRT) is 1 (register start mode)
- The IWDT reset interrupt request select bit (IWDTRCR.RSTIRQS) is 1 (reset output is enabled)
- The IWDT window start position select bits (IWDTCR.RPSS[1:0]) are 10b (75%)
- The IWDT window end position select bits (IWDTCR.RPES[1:0]) are 10b (25%)

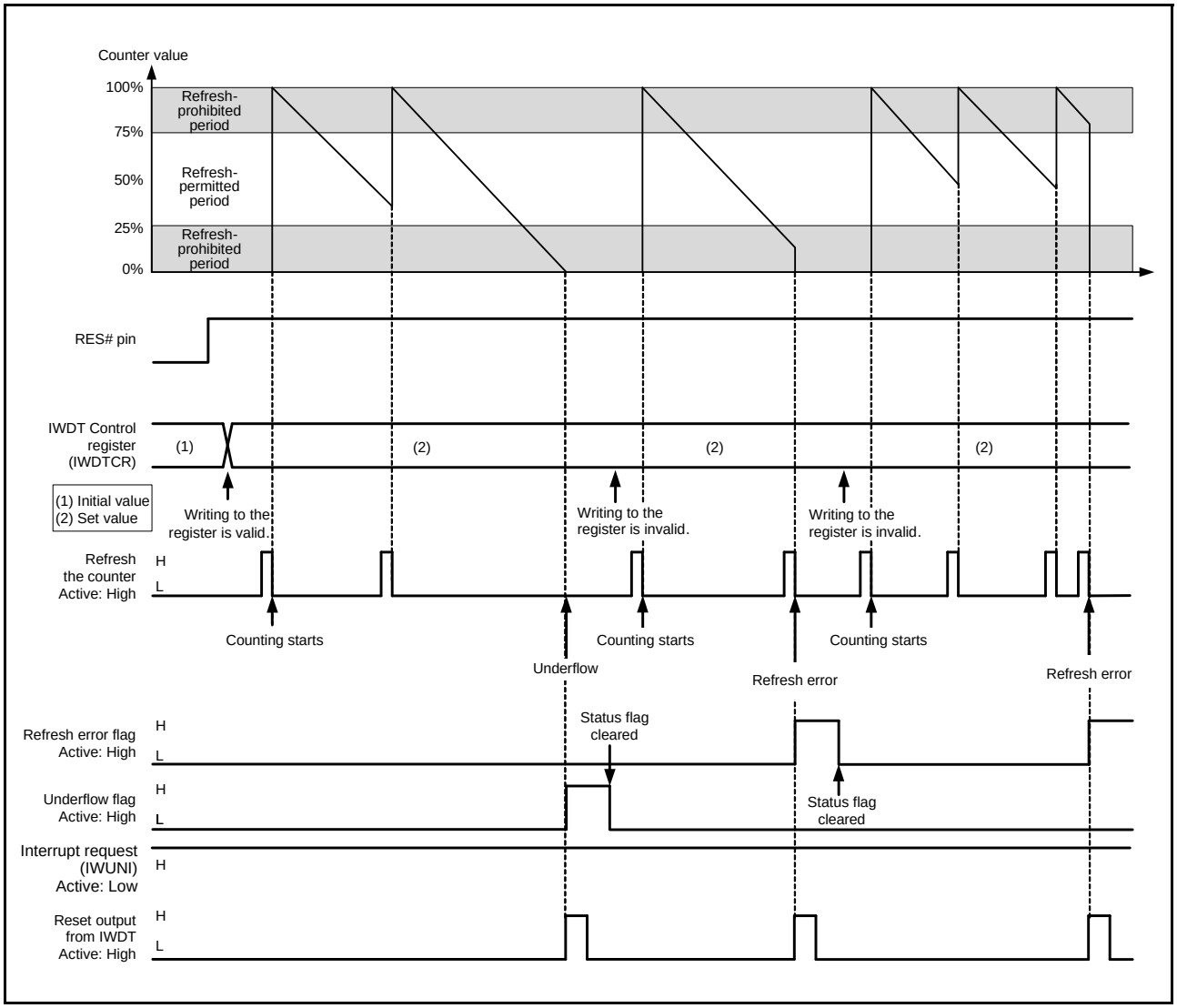


Figure 34.3 Operation Example in Register Start Mode

34.3.1.2 Auto-Start Mode

When the IWDT start mode select bit (OFS0.IWDTSTRT) in option function select register 0 is 0, auto-start mode is selected, and the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR) are disabled.
Within the reset state, the clock divide ratio, window start and end positions, timeout period, reset output or interrupt request output, and counter stop control at transitions to low power consumption states are set using the values specified in option function select register 0 (OFS0). When the reset state is released, the counter automatically starts counting down from the value selected by the IWDT timeout period select bits (OFS0.IWDTTOPS[1:0]).
After that, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the counter underflows because refreshing of the counter is not possible due to the program having entered crashed execution or if a refresh error occurs due to refreshing outside the refresh-permitted period, the IWDT outputs the reset signal or non-maskable interrupt request/interrupt request (IWUNI). After the reset signal or non-maskable interrupt request/interrupt request (IWUNI) is generated, the counter reloads the timeout period after counting for one cycle, and restarts counting. Set the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) to select either reset output or interrupt request output.

Figure 34.4 shows an example of operation under the following conditions.

- The IWDT start mode select bit (OFS0.IWDTSTRT) is 0 (auto-start mode)
- The IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) is 0 (non-maskable interrupt request output is enabled)
- The IWDT window start position select bits (OFS0.IWDTRPSS[1:0]) are 10b (75%)
- The IWDT window end position select bits (OFS0.IWDTRPES[1:0]) are 10b (25%)

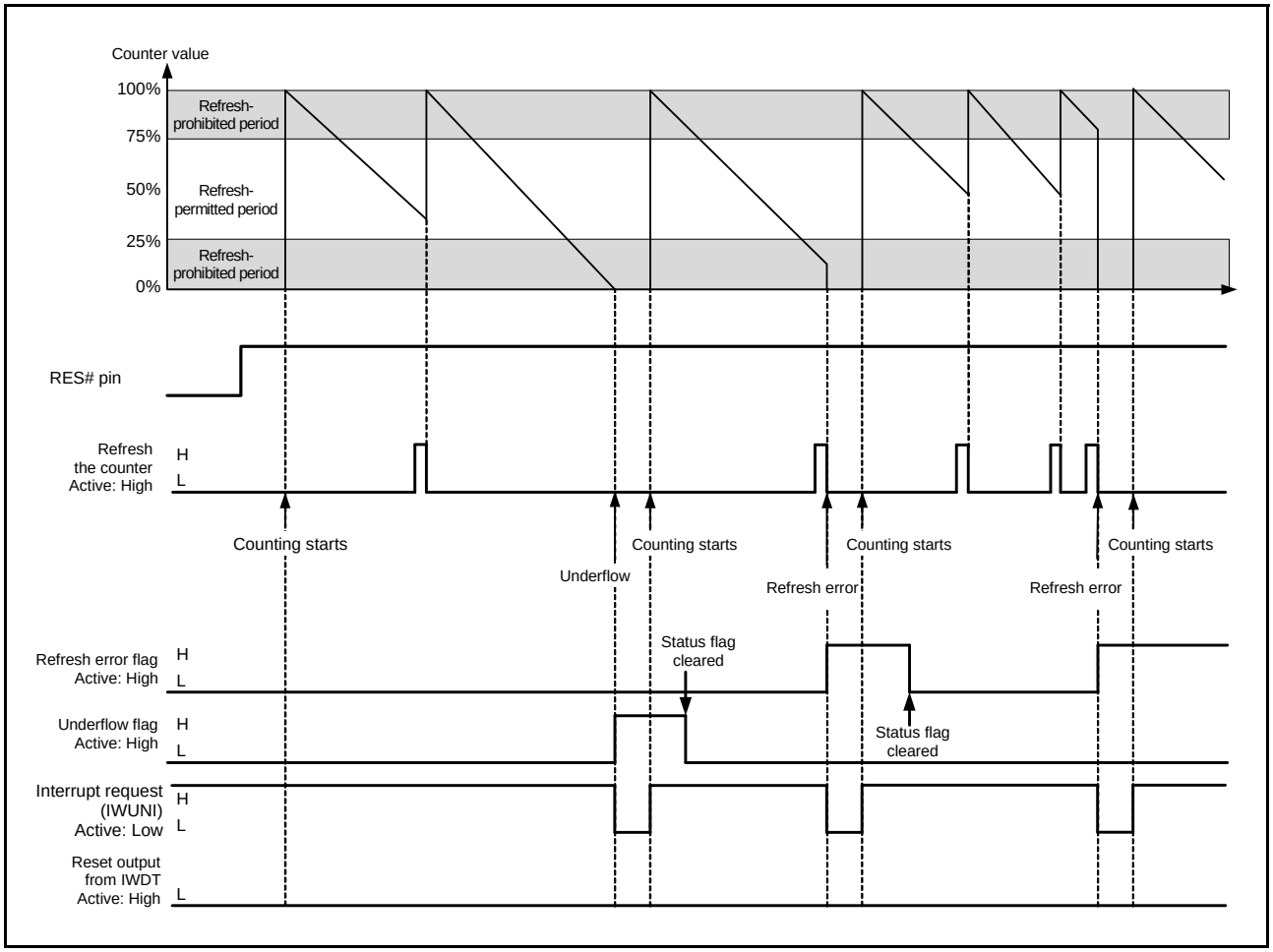


Figure 34.4 Operation Example in Auto-Start Mode

34.3.2 Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers

Writing to the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), or IWDT count stop control register (IWDTCSTPR) is only possible once between the release from the reset state and the first refresh operation. After a refresh operation (counting starts) or the IWDTCR, IWDTRCR, or IWDTCSTPR register is written to, the protection signal in the IWDT becomes 1 to protect registers IWDTCR, IWDTRCR, and IWDTCSTPR against subsequent attempts at writing.

This protection is released by the reset source of the IWDT. With other reset sources, the protection is not released.

Figure 34.5 shows control waveforms produced in response to writing to the IWDTCR register.

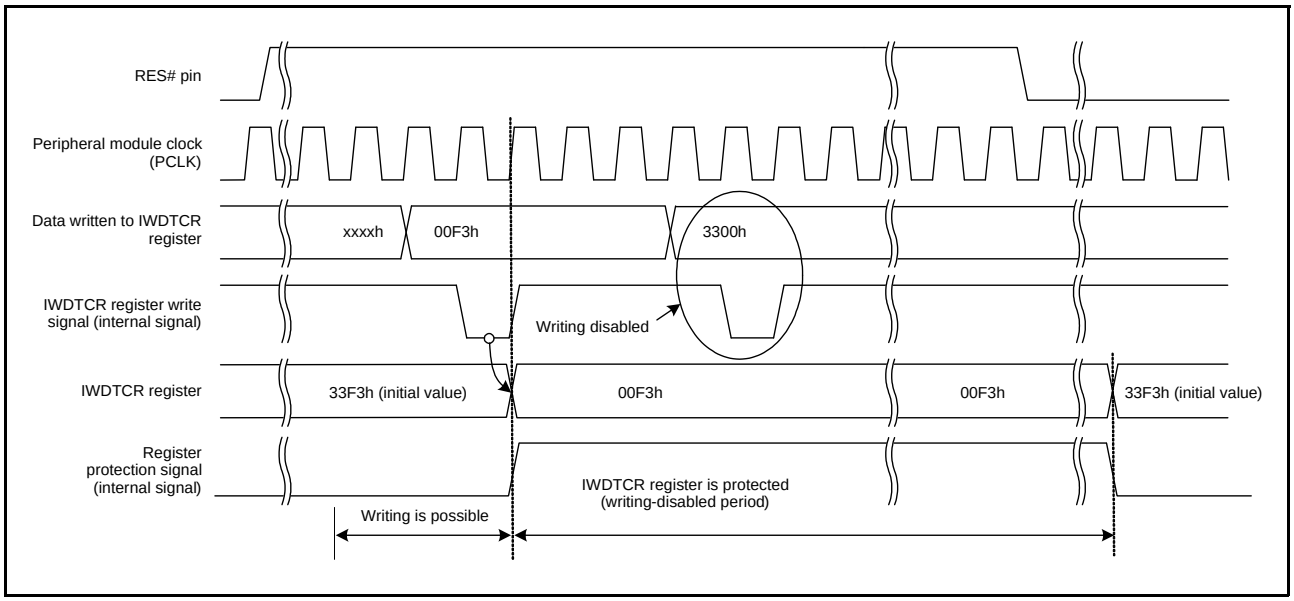


Figure 34.5 Control Waveforms Produced in Response to Writing to the IWDTCR Register

34.3.3 Refresh Operation

The counter is refreshed and starts operation (counting is started by refreshing) by writing the values 00h and then FFh to the IWDT refresh register (IWDTRR). If a value other than FFh is written after 00h, the counter is not refreshed. After such invalid writing, correct refreshing is performed by again writing 00h and then FFh to the IWDT refresh register (IWDTRR).
When writing is done in the order of 00h (first time) → 00h (second time), and if FFh is written after that, the writing order 00h → FFh is satisfied; writing 00h (n–1-th time) → 00h (nth time) → FFh is valid and correct refreshing will be done. Even when the first value written before 00h is not 00h, correct refreshing will be done if the operation contains the set of writing 00h → FFh. Moreover, even if a register other than the IWDTRR register is accessed or the IWDTRR register is read between writing 00h and writing FFh to the IWDTRR register, correct refreshing will be done.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h → FFh
- 00h (n–1-th time) → 00h (nth time) → FFh
- 00h → access to another register or read from the IWDTRR register → FFh

[Sample sequences of writing that are not valid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h → AAh (00h and a value other than FFh) → FFh

Even when 00h is written to the IWDTRR register outside the refresh-permitted period, if FFh is written to the IWDTRR register in the refresh-permitted period, the writing sequence is valid and refreshing will be done.
After FFh is written to the IWDTRR register, refreshing the counter requires up to four cycles of the signal for counting (the clock divide ratio selection bits (IWDTCR.CKS[3:0]) determine how many cycles of the IWDT-dedicated clock (IWDTCLK) make up one cycle for counting). Therefore, writing FFh to the IWDTRR register should be completed four-count cycles before the end position of the refresh-permitted period or a counter underflow. The value of the counter can be checked by the counter bits (IWDTSR.CNTVAL[13:0]).

[Sample refreshing timings]

- When the window start position is set to 1FFFh, even if 00h is written to the IWDTRR register before 1FFFh is reached (2002h, for example), refreshing is done if FFh is written to the IWDTRR register after the value of the IWDTSR.CNTVAL[13:0] bits has reached 1FFFh.
- When the window end position is set to 1FFFh, refreshing is done if 2003h (four-count cycles before 1FFFh) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to the IWDTRR register.
- When the refresh-permitted period continues until count 0000h, refreshing can be done immediately before an underflow. In this case, if 0003h (four-count cycles before an underflow) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to the IWDTRR register, no underflow occurs and refreshing is done.

Figure 34.6 shows the IWDT refresh-operation waveforms when PCLK > IWDTCLK and clock divide ratio = IWDTCLK.

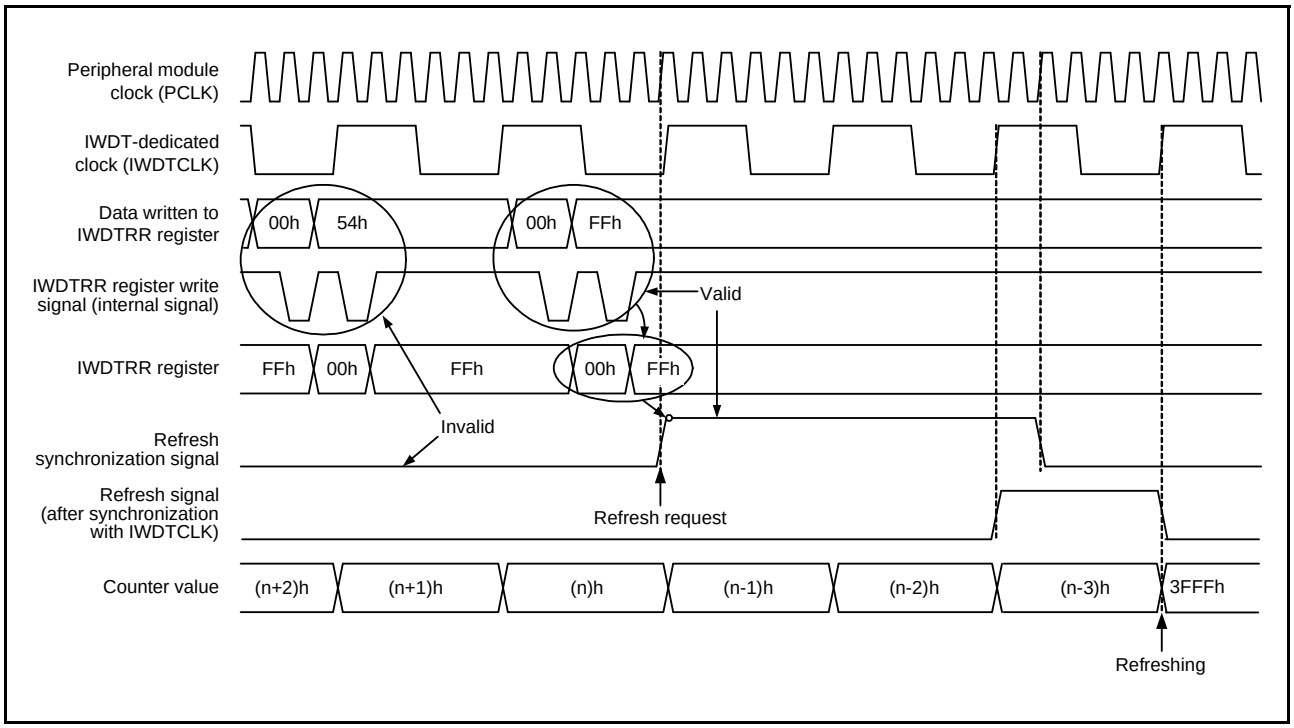


Figure 34.6 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

34.3.4 Status Flags

The refresh error (IWDTSR.REFEF) and underflow (IWDTSR.UNDFF) flags retain the source of the reset signal output from the IWDT or the source of the interrupt request from the IWDT.
Thus, after release from the reset state or interrupt request generation, read the IWDTSR.REFEF and IWDTSR.UNDFF flags to check for the reset or interrupt source.
For each flag, writing 0 clears the bit and writing 1 has no effect.
Leaving the status flags unchanged does not affect operation. If the flags are not cleared, at the time of the next reset or interrupt request from the IWDT, the earlier reset or interrupt source is cleared and the new reset or interrupt source is written.
After 0 is written to each flag, up to three IWDTCLK cycles and two PCLK cycles are required before the value is reflected.

34.3.5 Reset Output

When the reset interrupt selection bit (IWDTRCR.RSTIRQS) is set to 1 in register start mode or when the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) in option function select register 0 (OFS0) is set to 1 in auto-start mode, a reset signal is output when an underflow in the counter or a refresh error occurs.
In register start mode, the counter is initialized (all bits set to 0) and kept in that state after assertion of the reset signal. After the reset is released and the program is restarted, the counter is set up again and counting down is started by refreshing.
In auto-start mode, counting down automatically starts after the reset output.

34.3.6 Interrupt Sources

When the reset interrupt selection bit (IWDTRCR.RSTIRQS) is set to 0 in register start mode or when the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) in option function select register 0 (OFS0) is set to 0 in auto-start mode, an interrupt (IWUNI) signal is output when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt and an interrupt. For details, refer to section 15, Interrupt Controller (ICUA).

Table 34.4 IWDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
IWUNI	Counter underflow Refresh error	Not possible	Not possible

34.3.7 Reading the Counter Value

As the counter in IWDT-dedicated clock (IWDTCLK), the counter value cannot be read directly. The IWDT synchronizes the counter value with the peripheral module clock (PCLK) and stores it in the counter value bits (IWDTSR.CNTVAL[13:0]) of the IWDT status register. Thus, the counter value can be checked indirectly through the IWDTSR.CNTVAL[13:0] bits.

Reading the counter value requires multiple PCLK clock cycles (up to four clock cycles), and the read counter value may differ from the actual counter value by a value of one count.

Figure 34.7 shows the processing for reading the IWDT counter value when PCLK > IWDTCLK and clock divide ratio = IWDTCLK.

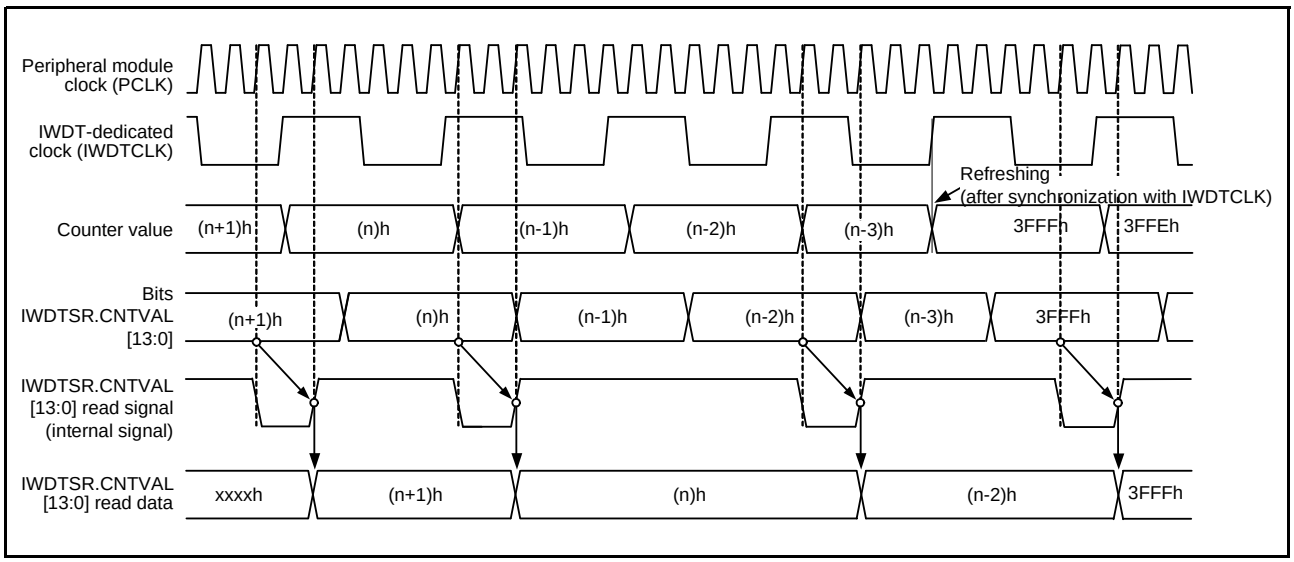


Figure 34.7 Processing for Reading IWDT Counter Value (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

34.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Table 34.5 lists the correspondence between option function select register 0 (OFS0) used in auto-start mode and the registers used in register start mode.
Do not change the OFS0 register setting during IWDT operation.
For details on option function select register 0 (OFS0), refer to section 7.2.3, Option Function Select Register 0 (OFS0).

Table 34.5 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Target of Control	Function	OFS0 Register (Enabled in Auto-Start Mode) OFS0.IWDTSTRT = 0	IWDT Registers (Enabled in Register Start Mode) OFS0.IWDTSTRT = 1
Counter	Timeout period selection	OFS0.IWDTTOPS[1:0]	IWDTCR.TOPS[1:0]
	Clock frequency divide ratio selection	OFS0.IWDTCKS[3:0]	IWDTCR.CKS[3:0]
	Window start position selection	OFS0.IWDTRPSS[1:0]	IWDTCR.RPSS[1:0]
	Window end position selection	OFS0.IWDTRPES[1:0]	IWDTCR.RPES[1:0]
Reset output or interrupt request output	Reset output or interrupt request output selection	OFS0.IWDTRSTIRQS	IWDTRCR.RSTIRQS
Count stop	Sleep mode count stop control	OFS0.IWDTSLCSTP	IWDTCSTPR.SLCSTP

34.4 Link Operation by ELC

The IWDT is capable of link operation for the previously specified module when interrupt request signal is used as an event signal by the event link controller (ELC). The event signal is output by the counter underflow and refresh error. An event signal is output regardless of the setting of the reset interrupt request selection bit (IWDTRCR.RSTIRQS) in register start mode or auto-start mode. An event signal can also be output upon generation of the next interrupt source while the refresh error flag (IWDTSR.REFEF) or underflow flag (IWDTSR.UNDFF) is 1.
For details, see section 21, Event Link Controller (ELC).

34.5 Usage Notes

34.5.1 Refresh Operations

When making the settings to control the timing of refreshing, consider variations in the range of errors due to the accuracy of the PCLK and IWDTCLK and set values which ensure that refreshing is possible.

34.5.2 Clock Divide Ratio Setting

Satisfy the frequency of the peripheral module clock (PCLK) ≥ 4 × (the frequency of the count source after divide).

35. Ethernet Controller (ETHERC)

35.1 Overview

This MCU has a two-channel Ethernet controller (ETHERC) compliant with the Ethernet or IEEE802.3 Media Access Control (MAC) layer protocol. Each ETHERC channel has one channel of the MAC layer interface, and connecting the MCU to the physical layer LSI (PHY-LSI) allows transmission and reception of frames compliant with the Ethernet/IEEE802.3 standard. The ETHERC is connected to the Ethernet controller direct memory access controller (EDMAC), so data can be transferred without using the CPU.

Table 35.1 lists the ETHERC specifications. Figure 35.1 shows the ETHERC configuration. Table 35.2 lists the ETHERC I/O pins.

Figure 35.2 and Figure 35.3 show examples of connecting the MCU to the PHY-LSI.

Table 35.1 ETHERC Specifications

Item	Description
Number of channels	Two channels
Protocol	Flow control compliant with IEEE802.3x
Data transmission/reception	Frames compliant with the Ethernet/IEEE802.3 standard can be transmitted and received.
Bit rate	Supports 10 Mbps and 100 Mbps
Operation modes	Supports full-duplex and half-duplex modes
Interfaces	Media Independent Interface (MII), Reduced Media Independent Interface (RMII), compliant with the IEEE802.3u standard
Functions	Magic Packet™ *1 detection, Wake-On-LAN (WOL) signal output

Note 1. Magic Packet is a trademark of Advanced Micro Devices, Inc.

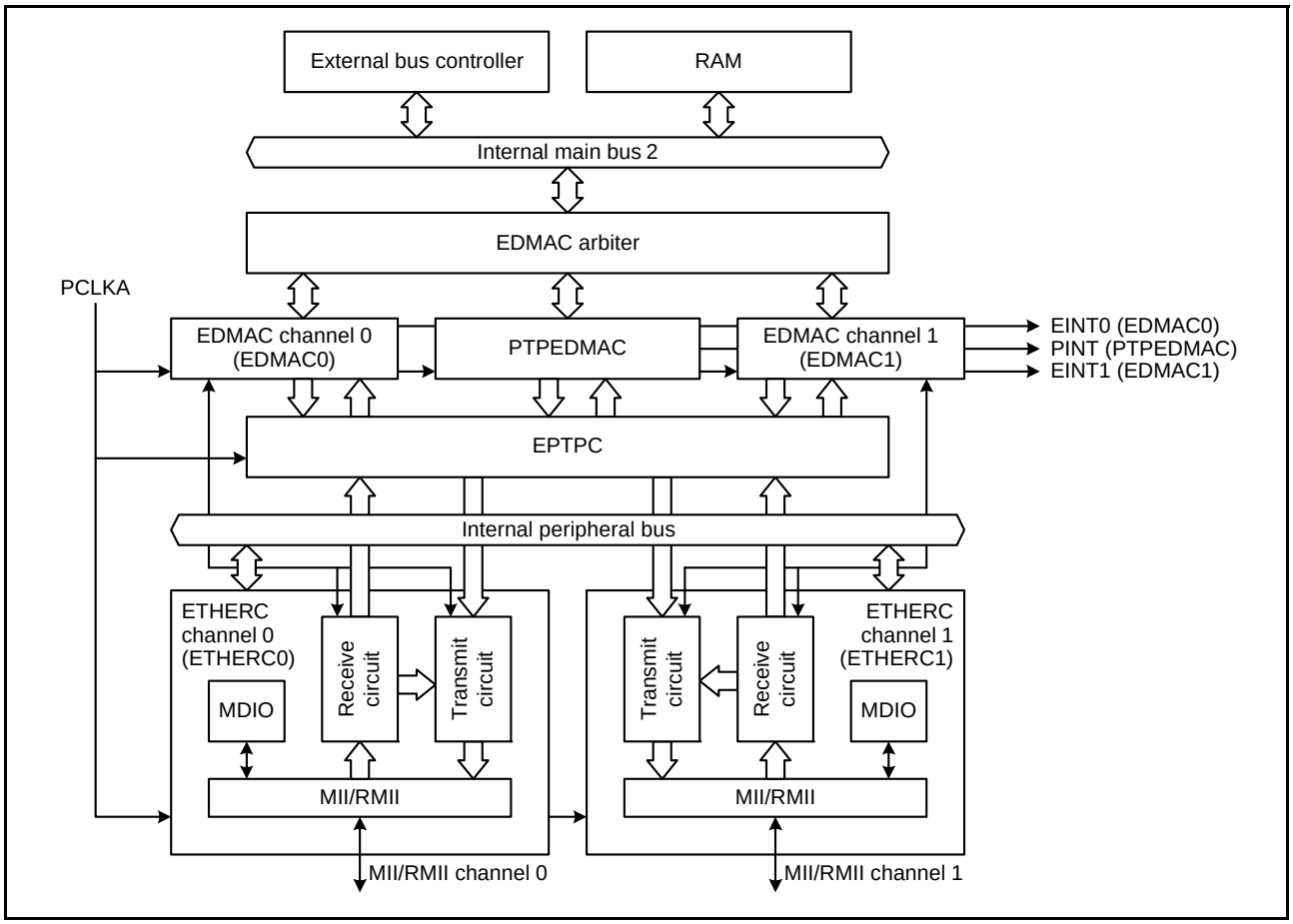


Figure 35.1 ETHERC Configuration

Table 35.2 ETHERC I/O Pins (n = 0, 1)

Operating Mode	Pin Name	I/O	Description
MII	ETn_TX_CLK *1	Input	Transmit clock Timing reference signal for outputting the ETn_TX_EN, ETn_ETXD3 to ETn_ETXD0, and ETn_TX_ER signals.
	ETn_RX_CLK *1	Input	Receive clock Timing reference signal for inputting the ETn_RX_DV, ETn_ERXD3 to ETn_ERXD0, and ETn_RX_ER signals.
	ETn_TX_EN *1	Output	Transmit data valid This signal indicates that valid transmit data has been output on pins ETn_ETXD3 to ETn_ETXD0.
	ETn_ETXD3 to ETn_ETXD0 *1	Output	4-bit transmit data
	ETn_TX_ER *1	Output	Transmit error This signal notifies the PHY-LSI that an error occurred during transmission.
	ETn_RX_DV *1	Input	Receive data valid This signal indicates that valid receive data is on pins ETn_ERXD3 to ETn_ERXD0.
	ETn_ERXD3 to ETn_ERXD0 *1	Input	4-bit receive data
	ETn_RX_ER *1	Input	Receive error This signal indicates that there is an error in a frame that is being transferred from the PHY-LSI to the ETHERC.
	ETn_CRS *1	Input	Carrier sense
	ETn_COL *1	Input	Collision detection signal
	ETn_MDC *1	Output	Management data clock Reference clock signal for transfer of information on the ETn_MDIO pin.
	ETn_MDIO *1	I/O	Management data I/O Bidirectional data signal for exchanging management data with the PHY-LSI.
	ETn_LINKSTA	Input	Link status input from the PHY-LSI
	ETn_EXOUT	Output	General output pin
	ETn_WOL	Output	Wake-On-LAN. This signal indicates that a Magic Packet has been received.
RMII	REF50CKn *2	Input	Reference clock Timing reference signal for pins RMIIn_TXD_EN, RMIIn_TXD1 to RMIIn_TXD0, RMIIn_CRS_DV, RMIIn_RXD1 to RMIIn_RXD0, and RMIIn_RX_ER.
	RMIIn_TXD_EN *2	Output	Transmit data valid This signal indicates that valid transmit data has been output on pins RMIIn_TXD1 and RMIIn_TXD0.
	RMIIn_TXD1 to RMIIn_TXD0 *2	Output	2-bit transmit data
	RMIIn_CRS_DV *2	Input	Carrier sense/receive data valid This signal indicates that valid receive data is on pins RMIIn_RXD1 and RMIIn_RXD0.
	RMIIn_RXD1 to RMIIn_RXD0 *2	Input	2-bit receive data
	RMIIn_RX_ER *2	Input	Receive error This signal indicates that there is an error in a frame that is being transferred from the PHY-LSI to the ETHERC.
	ETn_MDC *2	Output	Management data clock Reference clock signal for transfer of information on the ETn_MDIO pin
	ETn_MDIO *2	I/O	Management data I/O Bidirectional data signal for exchanging management data with the PHY-LSI.
	ETn_LINKSTA	Input	Link status input from the PHY-LSI.
	ETn_EXOUT	Output	General output pin
	ETn_WOL	Output	Wake-On-LAN. This signal indicates that a Magic Packet has been received.

Note 1. MII signal compliant with IEEE802.3u
Note 2. RMII signal compliant with IEEE802.3u

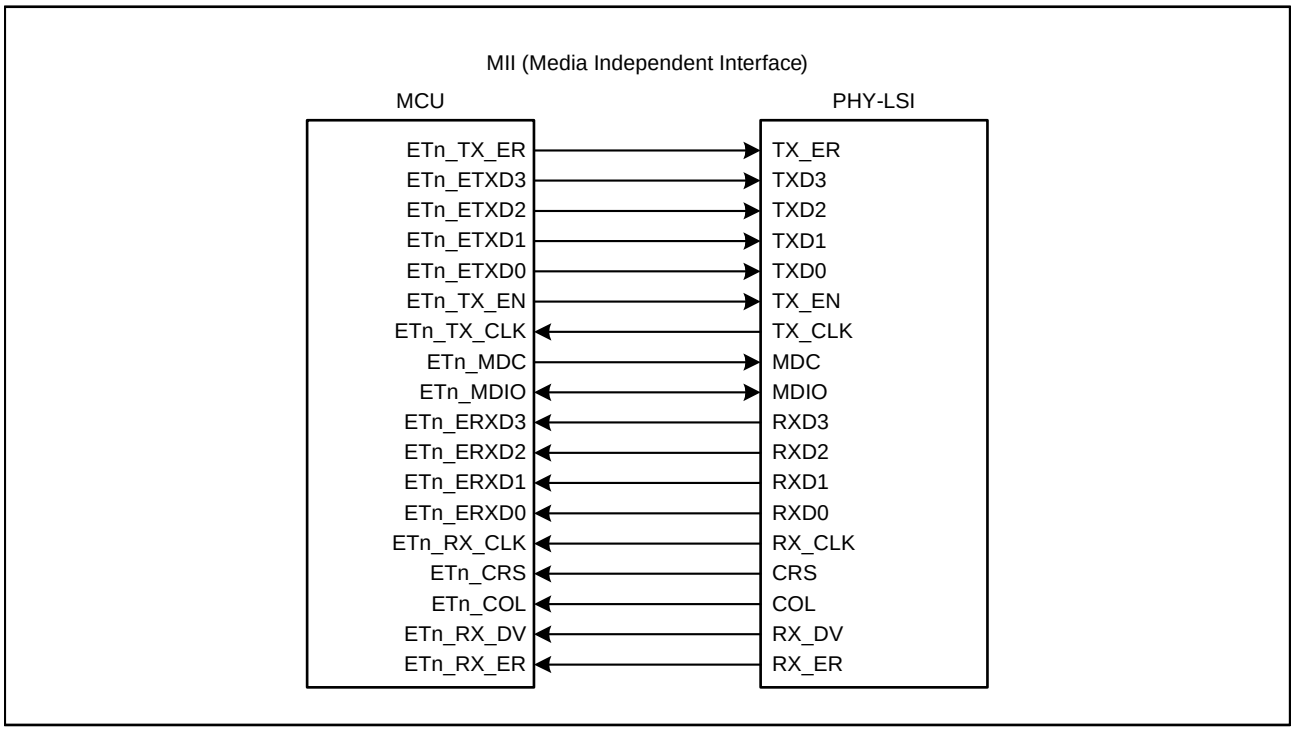


Figure 35.2 Example of Connection with the PHY-LSI for the MII

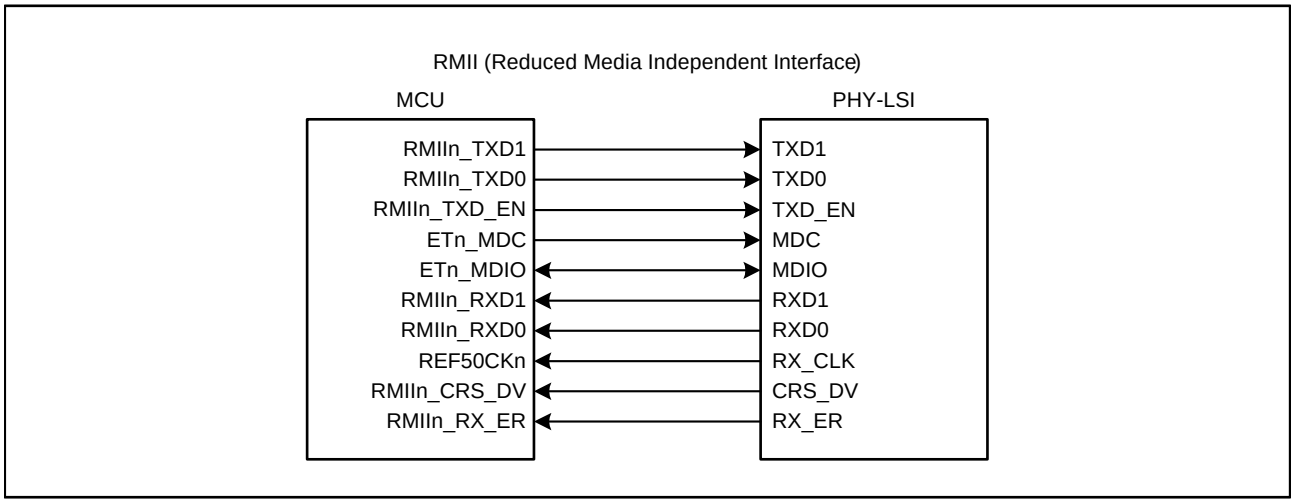


Figure 35.3 Example of Connection with the PHY-LSI for the RMII

35.2 Register Descriptions

35.2.1 ETHERC Mode Register (ECMR)

Address(es): ETHERC0.ECMR 000C 0100h, ETHERC1.ECMR 000C 0300h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	TPC	ZPF	PFR	RXF	TXF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PRCEF	—	—	MPDE	—	—	RE	TE	—	ILB	RTM	DM	PRM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRM	Promiscuous Mode	0: Promiscuous mode is disabled. 1: Promiscuous mode is enabled.	R/W
b1	DM	Duplex Mode	0: Half-duplex mode 1: Full-duplex mode	R/W
b2	RTM	Bit Rate	0: 10 Mbps 1: 100 Mbps	R/W
b3	ILB	Internal Loopback Mode	0: Normal data transmission or reception is performed. 1: Data is looped back in the ETHERC when full-duplex mode is selected.	R/W
b4	—	Reserved	The read value is 0. The write value should be 0.	R/W
b5	TE	Transmission Enable	0: Transmit function is disabled. 1: Transmit function is enabled.	R/W
b6	RE	Reception Enable	0: Receive function is disabled. 1: Receive function is enabled.	R/W
b8, b7	—	Reserved	The read value is 0. The write value should be 0.	R/W
b9	MPDE	Magic Packet Detection Enable	0: Magic Packet detection is disabled. 1: Magic Packet detection is enabled.	R/W
b11, b10	—	Reserved	The read value is 0. The write value should be 0.	R/W
b12	PRCEF	CRC Error Frame Receive Mode	0: EDMAC is notified of a CRC error. 1: EDMAC is not notified of a CRC error.	R/W
b15 to b13	—	Reserved	The read value is 0. The write value should be 0.	R/W
b16	TXF	Transmit Flow Control Operating Mode	0: Automatic PAUSE frame transmission is disabled. (PAUSE frame is not automatically transmitted.) 1: Automatic PAUSE frame transmission is enabled. (PAUSE frame is automatically transmitted as required.)	R/W
b17	RXF	Receive Flow Control Operating Mode	0: PAUSE frame detection is disabled. 1: PAUSE frame detection is enabled.	R/W
b18	PFR	PAUSE Frame Receive Mode	0: PAUSE frame is not transferred to the EDMAC. 1: PAUSE frame is transferred to the EDMAC.	R/W
b19	ZPF	0 Time PAUSE Frame Enable	0: PAUSE frame that contains the pause_time parameter of 0 is not used. 1: PAUSE frame that contains the pause_time parameter of 0 is used.	R/W
b20	TPC	PAUSE Frame Transmit	0: PAUSE frame is transmitted even during a PAUSE period. 1: PAUSE frame is not transmitted during a PAUSE period.	R/W
b31 to b21	—	Reserved	The read value is 0. The write value should be 0.	R/W

The ECMR register controls the ETHERC operation.

Set bits in the ECMR register, excluding bits TE and RE, during initialization after a reset. When rewriting this register

outside the initialization process, set the EDMACn.EDMR.SWR bit to 1 to reset the EDMAC and ETHERC, and then set this register again.

PRM Bit (Promiscuous Mode)

When the PRM bit is set to 1, the ETHERC operates in promiscuous mode where all Ethernet frames are received. In promiscuous mode, the ETHERC receives all valid frames regardless of whether or not the address matches the destination address or broadcast address and regardless of the multicast bit setting.

RTM Bit (Bit Rate)

The RTM bit sets the bit rate when the RMII is selected.

ILB Bit (Internal Loopback Mode)

When the ILB bit is set to 1, transmit frames can be looped back in the MCU. Set the DM bit to 1 (full-duplex mode) to perform a loopback test.

TE Bit (Transmission Enable)

When the TE bit is set to 1, the ETHERC transmit function is enabled.
When the TE bit is set to 0, the transmit function is disabled after the frame being processed is completely transmitted.

RE Bit (Reception Enable)

When the RE bit is set to 1, the ETHERC receive function is enabled.
When the RE bit is set to 0, the receive function is disabled after the frame being processed is completely received.

PRCEF Bit (CRC Error Frame Receive Mode)

When the PRCEF bit is set to 1, the EDMAC is not notified that a CRC error has occurred even when the error is detected in a receive frame. Accordingly, the EDMACn.EESR.CERF flag and RFS0 bit in receive descriptor 0 (RD0) do not become 1.

ZPF Bit (0 Time PAUSE Frame Enable)

When the ZPF bit is 1, a PAUSE frame that contains the pause_time parameter of 0 is transmitted when the PAUSE frame transmit request is canceled before the PAUSE time of the previously transmitted PAUSE frame has not elapsed. After the PAUSE frame that contains the pause_time parameter of 0 is received, the ETHERC is ready for transmission. When the ZPF bit is 0, even if the PAUSE frame transmit request from the receive FIFO is canceled, the next PAUSE frame is not transmitted until the PAUSE time of the previously transmitted PAUSE frame has elapsed. When a PAUSE frame that contains the pause_time parameter of 0 is received, the PAUSE frame is discarded.

35.2.2 Receive Frame Maximum Length Register (RFLR)

Address(es): ETHERC0.RFLR 000C 0108h, ETHERC1.RFLR 000C 0308h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	RFL[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	RFL[11:0]	Receive Frame Maximum Length	The set value becomes the maximum frame length. The minimum value that can be set is 1,518 bytes, and the maximum value that can be set is 2,048 bytes. Values that are less than 1,518 bytes are regarded as 1,518 bytes, and values larger than 2,048 bytes are regarded as 2,048 bytes.	R/W
b31 to b12	—	Reserved	The read value is 0. The write value should be 0.	R/W

The RFLR register sets the maximum frame length that can be received by the MCU. Set the length in bytes.
Do not rewrite this register while the ECMR.RE bit is 1 (receive function is enabled).

RFL[11:0] Bits (Receive Frame Maximum Length)

The RFL[11:0] bits set a frame length to be checked. When the number of bytes for fields from the destination address to the frame check sequence (FCS) of the received frame exceeds the RFL[11:0] bit value, the EDMAC is notified of a frame-too-long error.
When the received frame length exceeds the RFL[11:0] bit value, the excess data is discarded.

35.2.3 ETHERC Status Register (ECSR)

Address(es): ETHERC0.ECSR 000C 0110h, ETHERC1.ECSR 000C 0310h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	BFR	PSRTO	—	LCHNG	MPD	ICD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ICD	False Carrier Detect Flag	0: PHY-LSI has not detected a false carrier on the line. 1: PHY-LSI has detected a false carrier on the line.	R/W *1
b1	MPD	Magic Packet Detect Flag	0: Magic Packet has not been detected. 1: Magic Packet has been detected.	R/W *1
b2	LCHNG	Link Signal Change Flag	0: Change in the ETn_LINKSTA signal has not been detected. 1: Change in the ETn_LINKSTA signal has been detected (high to low, or low to high).	R/W *1
b3	—	Reserved	The read value is 0. The write value should be 0.	R/W
b4	PSRTO	PAUSE Frame Retransmit Over Flag	0: PAUSE frame retransmit count has not reached the upper limit. 1: PAUSE frame retransmit count has reached the upper limit.	R/W *1
b5	BFR	Continuous Broadcast Frame Reception Flag	0: The number of continuously received broadcast frames has not exceeded the value set in the BCFRR register. 1: The number of continuously received broadcast frames has exceeded the value set in the BCFRR register.	R/W *1
b31 to b6	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note 1. Write 1 to clear the flag.

The ECSR register indicates the status of the ETHERC.
When any flag in the ECSR register becomes 1 while the corresponding bit in the ECSIPR register is 1 (interrupt is notified), the EDMACn.EESR.ECI flag becomes 1.

ICD Flag (False Carrier Detect Flag)

The ICD flag indicates that the PHY-LSI has detected a false carrier on the line.
The ICD flag becomes 1 when a receive error signal shown in Figure 35.11 is received from the PHY-LSI. Note that the information may not be correct when signals input from the PHY-LSI change faster than software recognizes the change. Check the timing of the PHY-LSI.

LCHNG Flag (Link Signal Change Flag)

The LCHNG flag indicates that ETn_LINKSTA signal input from the PHY-LSI has changed from high to low, or from low to high.
Refer to the PSR.LMON flag for the current link status.

PSRTO Flag (PAUSE Frame Retransmit Over Flag)

The PSRTO flag indicates that the number of retransmissions reaches the value set in the TPAUSER register when retransmitting a PAUSE frame while automatic PAUSE frame transmission is enabled.

35.2.4 ETHERC Interrupt Enable Register (ECSIPR)

Address(es): ETHERC0.ECSIPR 000C 0118h, ETHERC1.ECSIPR 000C 0318h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	BFSIPR	PSRTOIP	—	LCHNGIP	MPDIP	ICDIP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ICDIP	False Carrier Detect Interrupt Enable	0: Notification of the false carrier detect interrupt is disabled. 1: Notification of the false carrier detect interrupt is enabled.	R/W
b1	MPDIP	Magic Packet Detect Interrupt Enable	0: Notification of the Magic Packet detect interrupt is disabled. 1: Notification of the Magic Packet detect interrupt is enabled.	R/W
b2	LCHNGIP	LINK Signal Change Interrupt Enable	0: Notification of ETn_LINKSTA signal change interrupt is disabled. 1: Notification of ETn_LINKSTA signal change interrupt is enabled.	R/W
b3	—	Reserved	The read value is 0. The write value should be 0.	R/W
b4	PSRTOIP	PAUSE Frame Retransmit Over Interrupt Enable	0: Notification of PAUSE frame retransmit over interrupt is disabled. 1: Notification of PAUSE frame retransmit over interrupt is enabled.	R/W
b5	BFSIPR	Continuous Broadcast Frame Reception Interrupt Enable	0: Notification of continuous broadcast frame reception interrupt is disabled. 1: Notification of continuous broadcast frame reception interrupt is enabled.	R/W
b31 to b6	—	Reserved	The read value is 0. The write value should be 0.	R/W

The ECSIPR register selects whether or not to notify the EDMAC of the status indicated by the ECSR register. Each bit corresponds to the flag in the ECSR register that has the same bit number.

35.2.5 PHY Interface Register (PIR)

Address(es): ETHERC0.PIR 000C 0120h, ETHERC1.PIR 000C 0320h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	MDI	MDO	MMD	MDC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MDC	MII/RMII Management Data Clock	The MDC bit value is output from the ETn_MDC pin to supply the management data clock to the MII or RMII.	R/W
b1	MMD	MII/RMII Management Mode	0: Read 1: Write	R/W
b2	MDO	MII/RMII Management Data-Out	The MDO bit value is output from the ETn_MDIO pin when the MMD bit is 1 (write). The value is not output when the MMD bit is 0 (read).	R/W
b3	MDI	MII/RMII Management Data-In	This bit indicates the level of the ETn_MDIO pin. The write value should be 0.	R/W
b31 to b4	—	Reserved	The read value is 0. The write value should be 0.	R/W

The PIR register is used to access registers in the PHY-LSI via the MII or RMII. The management clock and management data are controlled by software.
Refer to section 35.3.4, Accessing MII/RMII Registers for details on accessing MII and RMII registers.

35.2.6 PHY Status Register (PSR)

Address(es): ETHERC0.PSR 000C 0128h, ETHERC1.PSR 000C 0328h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LMON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

Bit	Symbol	Bit Name	Description	R/W
b0	LMON	ETn_LINKSTA Pin Status Flag	The link status can be read by connecting the link signal output from the PHY-LSI to the ETn_LINKSTA pin. For details on the polarity, refer to the specifications of the connected PHY-LSI.	R
b31 to b1	—	Reserved	The read value is 0. The write value should be 0.	R

The PSR register is used to monitor interface signals from the PHY-LSI.

35.2.7 Random Number Generation Counter Limit Setting Register (RDMLR)

Address(es): ETHERC0.RDMLR 000C 0140h, ETHERC1.RDMLR 000C 0340h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	RMD[19:16]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RMD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b19 to b0	RMD[19:0]	Random Number Generation Counter	00000h: Normal operation 00001h to FFFFFh: Setting prohibited	R/W
b31 to b20	—	Reserved	The read value is 0. The write value should be 0.	R/W

The RDMLR register sets the maximum value for the counter used in the random number generator.
Do not rewrite this register while the ECMR.TE bit is 1 (transmit function is enabled) or while the ECMR.RE bit is 1 (receive function is enabled).

35.2.8 Interpacket Gap Register (IPGR)

Address(es): ETHERC0.IPGR 000C 0150h, ETHERC1.IPGR 000C 0350h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	IPG[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	IPG[4:0]	Interpacket Gap	00h: 16 bit time 01h: 20 bit time : : 14h: 96 bit time (initial value) : : 1Fh: 140 bit time	R/W
b31 to b5	—	Reserved	The read value is 0. The write value should be 0.	R/W

The IPGR register sets the interpacket gap (IPG) value.
Do not rewrite this register while the ECMR.TE bit is 1 (transmit function is enabled) or while the ECMR.RE bit is 1 (receive function is enabled).
Refer to section 35.3.6, Adjusting Transmission Efficiency by Changing the IPG for details on the IPG.

35.2.9 Automatic PAUSE Frame Register (APR)

Address(es): ETHERC0.APR 000C 0154h, ETHERC1.APR 000C 0354h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	AP[15:0]	Automatic PAUSE Time Setting	These bits set the value of the pause_time parameter for a PAUSE frame that is automatically transmitted. Transmission is not performed until the set value multiplied by 512 bit time has elapsed.	R/W
b31 to b16	—	Reserved	The read value is 0. The write value should be 0.	R/W

The APR register sets the PAUSE time of the PAUSE frame that is automatically transmitted. The value set in the APR register is used for the pause_time parameter of the PAUSE frame.
Do not rewrite this register while the ECMR.TE bit is 1 (transmit function is enabled) or while the ECMR.RE bit is 1 (receive function is enabled).

35.2.10 Manual PAUSE Frame Register (MPR)

Address(es): ETHERC0.MPR 000C 0158h, ETHERC1.MPR 000C 0358h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	MP[15:0]	Manual PAUSE Time Setting	These bits set the value of the pause_time parameter for a PAUSE frame that is manually transmitted. Transmission is not performed until the set value multiplied by 512 bit time has elapsed. The read value is undefined.	R/W
b31 to b16	—	Reserved	The read value is 0. The write value should be 0.	R/W

The MPR register sets the PAUSE time of the PAUSE frame that is manually transmitted. The value set in the MPR register is used for the pause_time parameter of the PAUSE frame.
When a value is set to this register, a PAUSE frame is transmitted. Rewrite this register while the ECMR.TE bit is 1 (transmit function is enabled).

35.2.11 Received PAUSE Frame Counter (RFCF)

Address(es): ETHERC0.RFCF 000C 0160h, ETHERC1.RFCF 000C 0360h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	RPAUSE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	RPAUSE[7:0]	Received PAUSE Frame Count	Number of received PAUSE frames	R
b31 to b8	—	Reserved	The read value is 0.	R

The RFCF register is a counter indicating the number of received PAUSE frames. The counter is reset after this register is read.

35.2.12 PAUSE Frame Retransmit Count Setting Register (TPAUSER)

Address(es): ETHERC0.TPAUSER 000C 0164h, ETHERC1.TPAUSER 000C 0364h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TPAUSE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TPAUSE[15:0]	Automatic PAUSE Frame Retransmit Setting	0000h: Number of retransmissions is unlimited 0001h: Maximum number of retransmissions is 1 : : FFFFh: Maximum number of retransmissions is 65,535	R/W
b31 to b16	—	Reserved	The read value is 0. The write value should be 0.	R/W

The TPAUSER register selects the maximum number of times a PAUSE frame is automatically transmitted.
Do not rewrite this register while the ECMR.TE bit is 1 (transmit function is enabled).

35.2.13 PAUSE Frame Retransmit Counter (TPAUSECR)

Address(es): ETHERC0.TPAUSECR 000C 0168h, ETHERC1.TPAUSECR 000C 0368h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TXP[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TXP[7:0]	PAUSE Frame Retransmit Count	Number of times a PAUSE frame was retransmitted	R
b31 to b8	—	Reserved	The read value is 0.	R

The TPAUSECR register is a counter indicating the number of times a PAUSE frame was automatically retransmitted.
The counter is reset after this register is read.

35.2.14 Broadcast Frame Receive Count Setting Register (BCFRR)

Address(es): ETHERC0.BCFRR 000C 016Ch, ETHERC1.BCFRR 000C 036Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BCF[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	BCF[15:0]	Broadcast Frame Continuous Receive Count Setting	0000h: Number of receptions is unlimited. 0001h: Receive 1 frame. : : FFFFh: Receive 65,535 frames.	R/W
b31 to b16	—	Reserved	The read value is 0. The write value should be 0.	R/W

The BCFRR register sets the number of times broadcast frames can be received continuously. When the number of received frames exceeds the BCF[15:0] bit value, the ECSR.BFR flag becomes 1 and the excess broadcast frames are discarded. The internal counter that counts the number of continuously received broadcast frames is reset when receiving any other frame than broadcast frame.
Do not rewrite this register while the ECMR.RE bit is 1 (receive function is enabled).

35.2.15 MAC Address Upper Bit Register (MAHR)

Address(es): ETHERC0.MAHR 000C 01C0h, ETHERC1.MAHR 000C 03C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The MAHR register sets the upper 32 bits (b47 to b16) of the 48-bit MAC address. For example, if the MAC address is 01-23-45-67-89-AB, set the register to 0123 4567h.
Set the MAHR register during initialization after a reset. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function is enabled) or while the ECMR.RE bit is 1 (receive function is enabled). When rewriting this register, set the EDMACn.EDMR.SWR bit to 1 to reset the EDMAC and ETHERC and then set this register again.

35.2.16 MAC Address Lower Bit Register (MALR)

Address(es): ETHERC0.MALR 000C 01C8h, ETHERC1.MALR 000C 03C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	—	These bits set the lower 16 bits of the MAC address.	R/W
b31 to b16	—	Reserved	The read value is 0. The write value should be 0.	R/W

The MALR register sets the lower 16 bits of the 48-bit MAC address. For example, if the MAC address is 01-23-45-67-89-AB, set the register to 0000 89ABh.
Set the MALR register during initialization after a reset. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function is enabled) or while the ECMR.RE bit is 1 (receive function is enabled). When rewriting this register, set the EDMACn.EDMR.SWR bit to 1 to reset the EDMAC and ETHERC and then set this register again.

35.2.17 Transmit Retry Over Counter Register (TROCR)

Address(es): ETHERC0.TROCR 000C 01D0h, ETHERC1.TROCR 000C 03D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The TROCR register is a counter indicating the number of frames that fail to be retransmitted.
The TROCR register is incremented by 1 when a frame fails to be retransmitted 15 times. The counter stops when the TROCR register value becomes FFFF FFFFh. The counter value becomes 0 by writing any value to the TROCR register.

35.2.18 Late Collision Detect Counter Register (CDCR)

Address(es): ETHERC0.CDCR 000C 01D4h, ETHERC1.CDCR 000C 03D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CDCR register is a counter indicating the number of late collisions that have been detected after transmission starts. When the CDCR register value becomes FFFF FFFFh, the counter stops. The counter value becomes 0 by writing any value to the CDCR register.

35.2.19 Lost Carrier Counter Register (LCCR)

Address(es): ETHERC0.LCCR 000C 01D8h, ETHERC1.LCCR 000C 03D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The LCCR register is a counter indicating the number of times a loss of carrier is detected during frame transmission. When the LCCR register value becomes FFFF FFFFh, the counter stops. The counter value becomes 0 by writing any value to the LCCR register.

35.2.20 Carrier Not Detect Counter Register (CNDCR)

Address(es): ETHERC0.CNDCR 000C 01DCh, ETHERC1.CNDCR 000C 03DCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CNDCR register is a counter indicating the number of times a carrier is not detected during preamble transmission. When the CNDCR register value becomes FFFF FFFFh, the counter stops. The counter value becomes 0 by writing any value to the CNDCR register.

35.2.21 CRC Error Frame Receive Counter Register (CEFCR)

Address(es): ETHERC0.CEFCR 000C 01E4h, ETHERC1.CEFCR 000C 03E4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CEFCR register is a counter indicating the number of received frames where a CRC error has been detected. When the CEFCR register value becomes FFFF FFFFh, the counter stops. The counter value becomes 0 by writing any value to the CEFCR register.

35.2.22 Frame Receive Error Counter Register (FRECR)

Address(es): ETHERC0.FRECR 000C 01E8h, ETHERC1.FRECR 000C 03E8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FRECR register is a counter indicating the number of times a frame receive error has occurred. The PHY-LSI notifies the ETHERC of the frame receive error using the ETn_RX_ER pin.
The FRECR register is incremented each time the ETn_RX_ER pin becomes high. When the FRECR register value becomes FFFF FFFFh, the counter stops. The counter value becomes 0 by writing any value to the FRECR register.
When a frame receive error occurs, the EPTPCn.SYSR.INFABT flag may become 1 regardless of use of the EPTPC. In this case, reset the EPTPC, PTPEDMAC, and the corresponding channels ETHERC and EDMAC. Refer to section 35.5.3, Handling Errors in Control Information for the procedure to reset these modules.

35.2.23 Too-Short Frame Receive Counter Register (TSFRCR)

Address(es): ETHERC0.TSFRCR 000C 01ECh, ETHERC1.TSFRCR 000C 03ECh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The TSFRCR register is a counter indicating the number of times a short frame that is shorter than 64 bytes has been received. When the TSFRCR register value becomes FFFF FFFFh, the counter stops. The counter value becomes 0 by writing any value to the TSFRCR register.

35.2.24 Too-Long Frame Receive Counter Register (TLFRCR)

Address(es): ETHERC0.TLFRCR 000C 01F0h, ETHERC1.TLFRCR 000C 03F0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The TLFRCR register is a counter indicating the number of times a long frame that is longer than the RFLR register value has been received.
When the TLFRCR register value becomes FFFF FFFFh, the counter stops. The counter value becomes 0 by writing any value to the TLFRCR register.
Note that the TLFRCR register is not incremented when a frame is received with the alignment error. In this case, the RFCR register is incremented.

35.2.25 Received Alignment Error Frame Counter Register (RFCR)

Address(es): ETHERC0.RFCR 000C 01F4h, ETHERC1.RFCR 000C 03F4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The RFCR register is a counter indicating the number of times a frame has been received with the alignment error (frame is not an integral number of octets).
When the RFCR register value becomes FFFF FFFFh, the counter stops. The counter value becomes 0 by writing any value to the RFCR register.
When a frame is received with an alignment error, the EPTPCn.SYSR.INFABT flag may become 1 regardless of use of the EPTPC. In this case, reset the EPTPC, PTPEDMAC, and the corresponding channels ETHERC and EDMAC. Refer to section 35.5.3, Handling Errors in Control Information for the procedure to reset these modules.

35.2.26 Multicast Address Frame Receive Counter Register (MAFCR)

Address(es): ETHERC0.MAFCR 000C 01F8h, ETHERC1.MAFCR 000C 03F8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

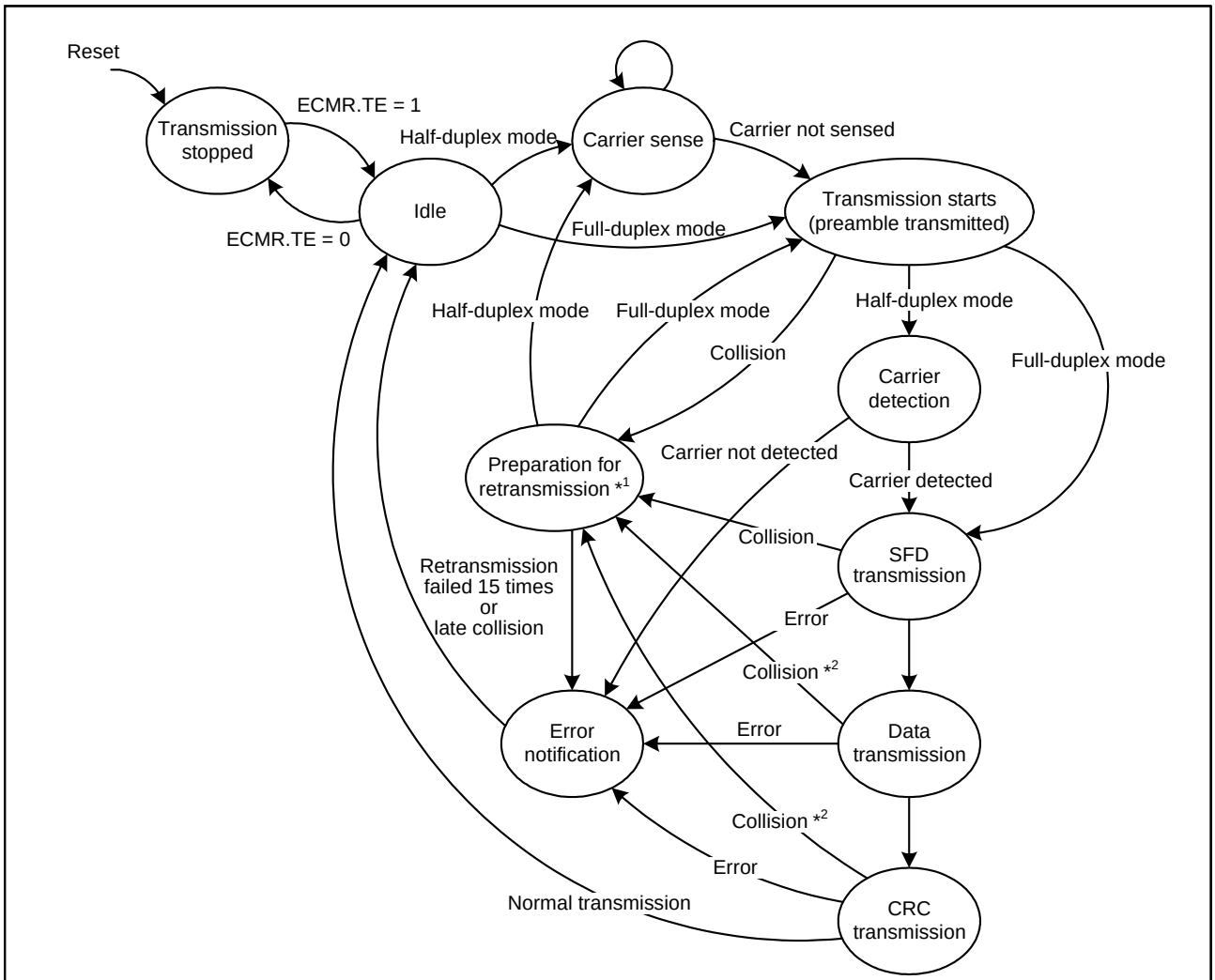
The MAFCR register is a counter indicating the number of times a frame where the multicast address is set has been received.
When the RFCR register value becomes FFFF FFFFh, the counter stops. The counter value becomes 0 by writing any value to the RFCR register.

35.3 Operation

This section is an overview of the ETHERC operations. The ETHERC supports flow control compliant with IEEE802.3x, and can transmit and receive PAUSE frames.

35.3.1 Transmission

The ETHERC transmitter assembles transmit data into a frame and outputs it to the MII/RMII when a transmit request is received from the EDMAC. The frame transmitted via the MII/RMII is transmitted on the line by the PHY-LSI. Figure 35.4 shows the state transitions of the ETHERC transmitter.



Note 1. Preparation for retransmission includes transmitting a jam signal and adjusting transmission time intervals using the backoff algorithm after a collision is detected.

Note 2. Preparation for retransmission is performed only when a collision is detected within 512 bit time after preamble transmission starts. When a collision is detected after 512 bit time, only a jam signal is transmitted and processing using the backoff algorithm is not performed.

Figure 35.4 ETHERC Transmitter State Transitions

1. When setting the ECMR.TE bit to 1, the ETHERC enters the transmit idle state.
2. When a transmit request is received from the EDMAC, the ETHERC enters the carrier sense state. The ETHERC waits for the interpacket gap and then transmits a preamble to the MII/RMII. When full-duplex mode is selected, it is not required to sense a carrier, so the ETHERC transmits a preamble immediately after receiving a transmit request from the EDMAC.

3. The ETHERC transmits the Start Frame Delimiter (SFD), transmit data, and CRC sequentially. When the transmission is completed successfully, the ETHERC notifies the EDMAC of successful completion, and the EDMAC sets the EDMACn.EESR.TC flag to 1. When a late collision or loss of carrier is detected during data transmission, the ETHERC stops the transmission and notifies the EDMAC of the error.
4. After the time for the interpacket gap has elapsed, the ETHERC enters the idle state and continues transmission when transmit data remains.

35.3.2 Reception

The ETHERC receiver separates the frame input from the MII/RMII into the preamble, SFD, receive data, and CRC, and transmits only receive data (destination address, source address, type/length, data/LLC). Figure 35.5 shows the state transitions of the ETHERC receiver.

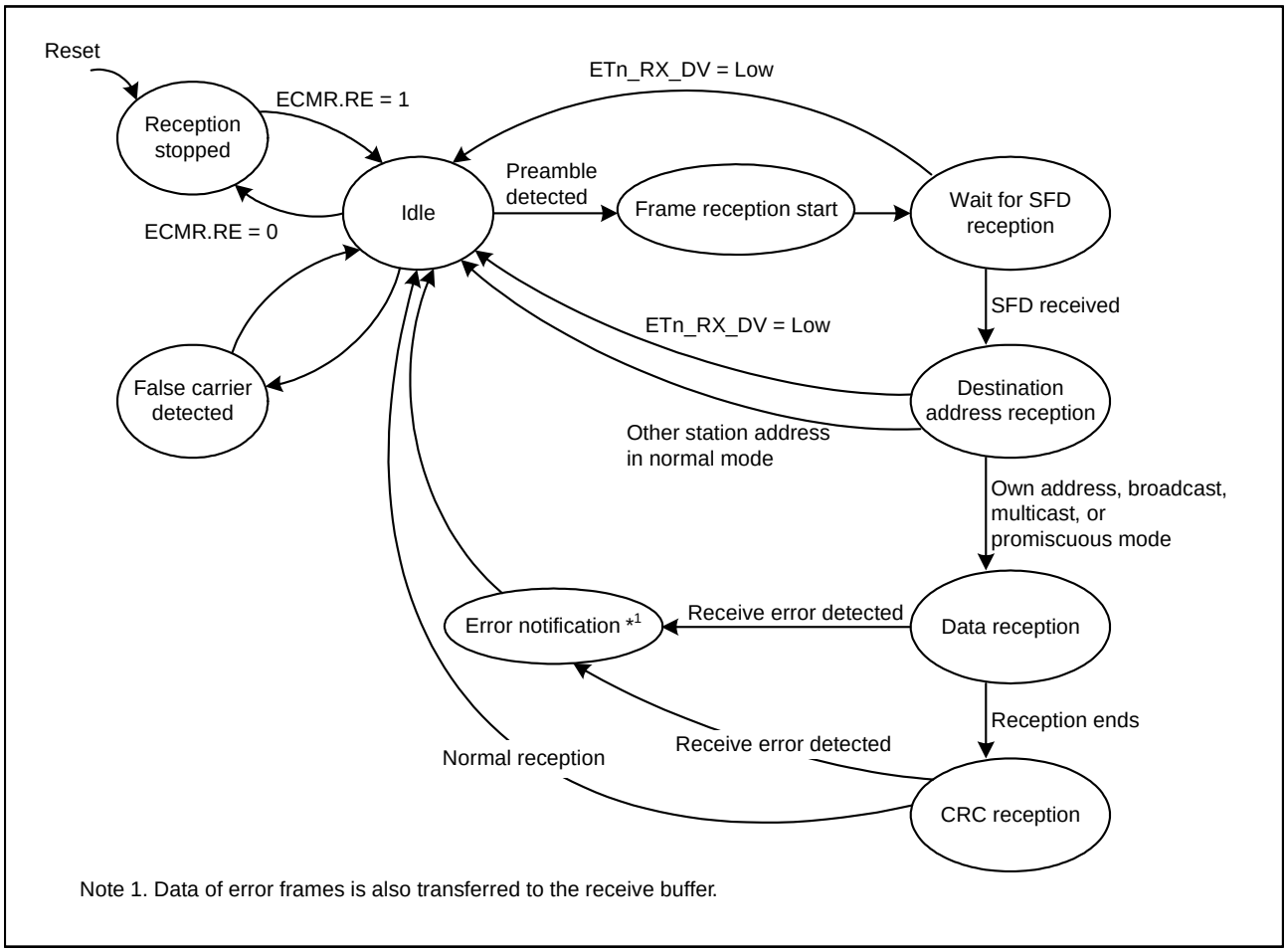


Figure 35.5 ETHERC Receiver State Transitions

1. When setting the ECMR.RE bit to 1, the ETHERC enters the receive idle state.
2. When the SFD following the preamble of the receive packet is detected, the ETHERC starts reception. If the received SFD is invalid, the ETHERC discards the frame.
3. In normal mode, the ETHERC starts data reception when the destination address of the receive frame is the address of the MCU or the receive frame is broadcast frame or multicast frame. In promiscuous mode, the ETHERC starts data reception regardless of the receive frame type.
4. After receiving data from the MII/RMII, the ETHERC performs the CRC check. The ETHERC notifies the EDMAC of the CRC check result. After the received data is transferred to the receive buffer, the CRC check result is written back to the receive descriptor as a status. The result is also reflected in the EDMACn.EESR.CERF flag.
5. When the ECMR.RE bit is 1 after one frame has been received, the ETHERC prepares to receive the next frame.

35.3.3 Frame Timing

35.3.3.1 MII Frame Timing

Figure 35.6 to Figure 35.11 show the MII frame timing.

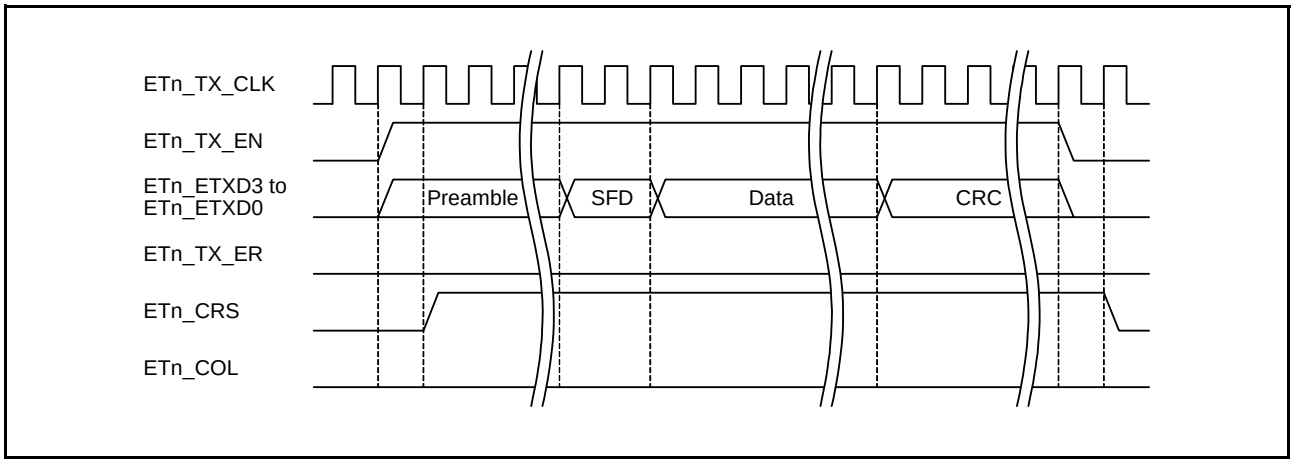


Figure 35.6 MII Frame Transmit Timing During Normal Transmission

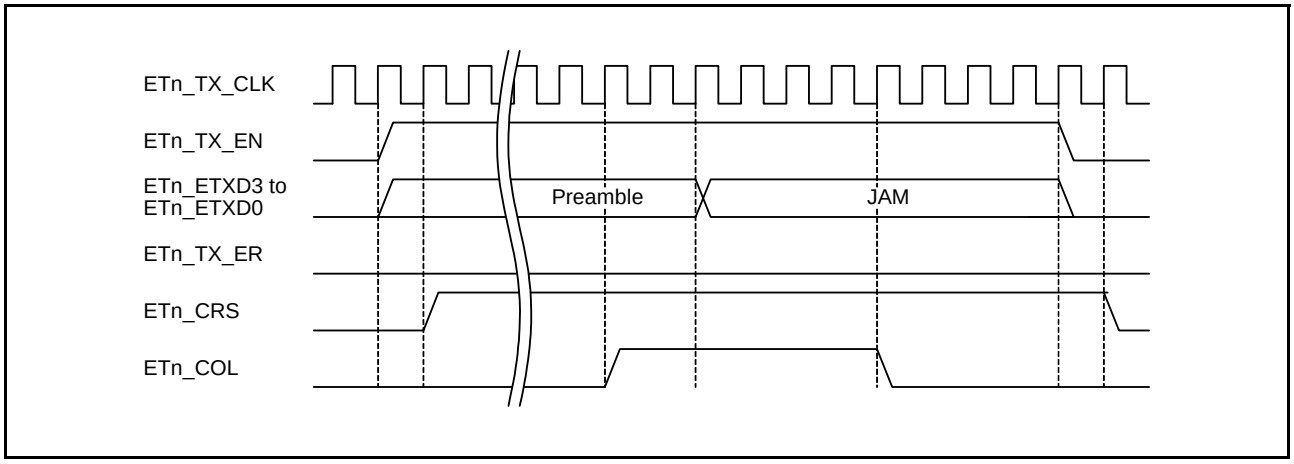


Figure 35.7 MII Frame Transmit Timing When Collision Occurs

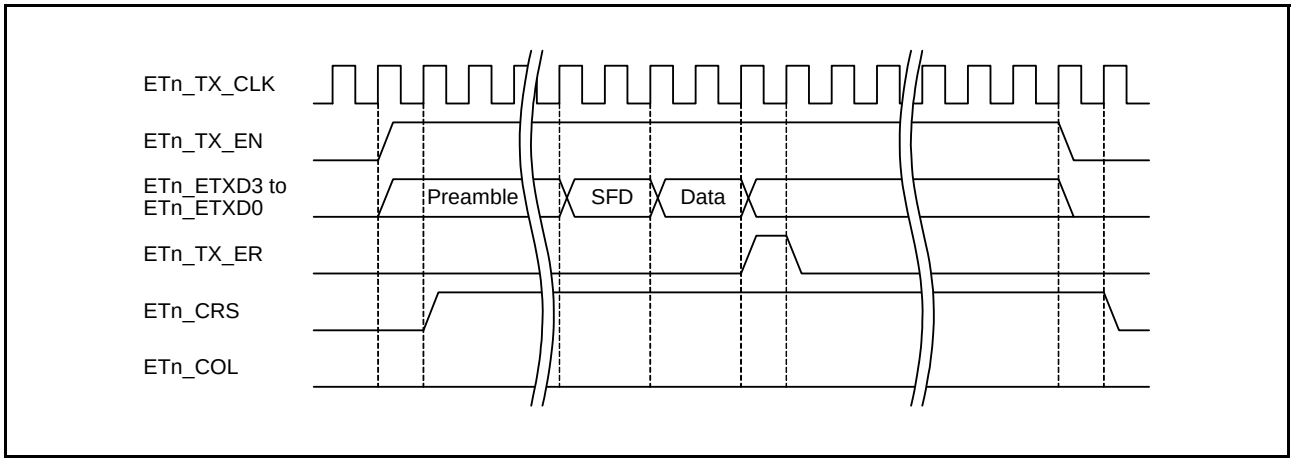


Figure 35.8 MII Frame Transmit Timing When Transmit Error Occurs

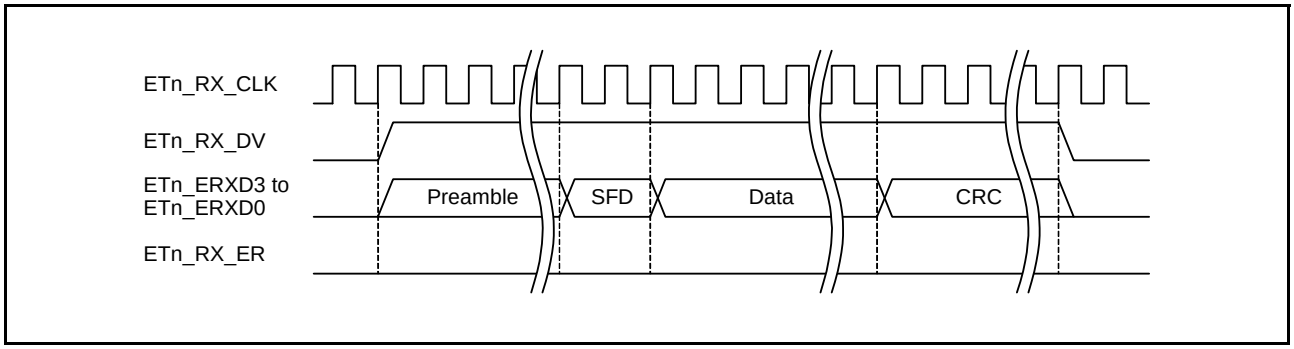


Figure 35.9 MII Frame Receive Timing During Normal Reception

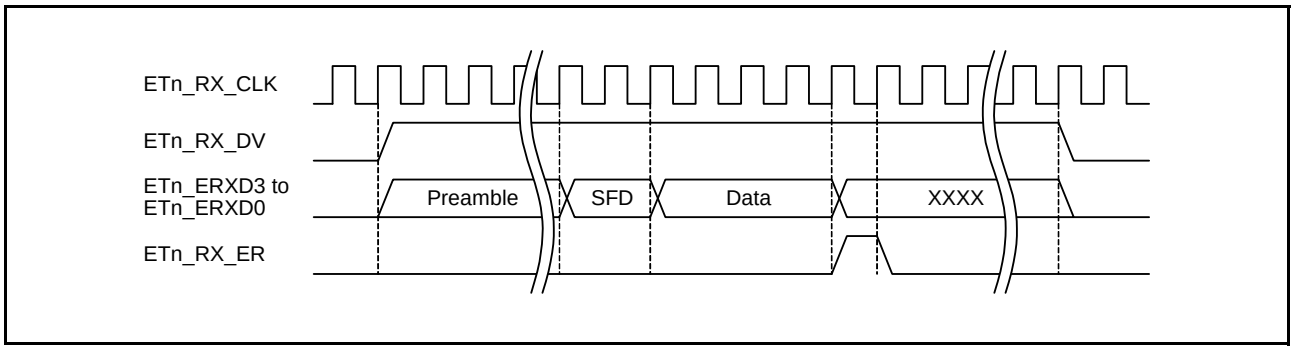


Figure 35.10 MII Frame Receive Timing for Receive Error Notification

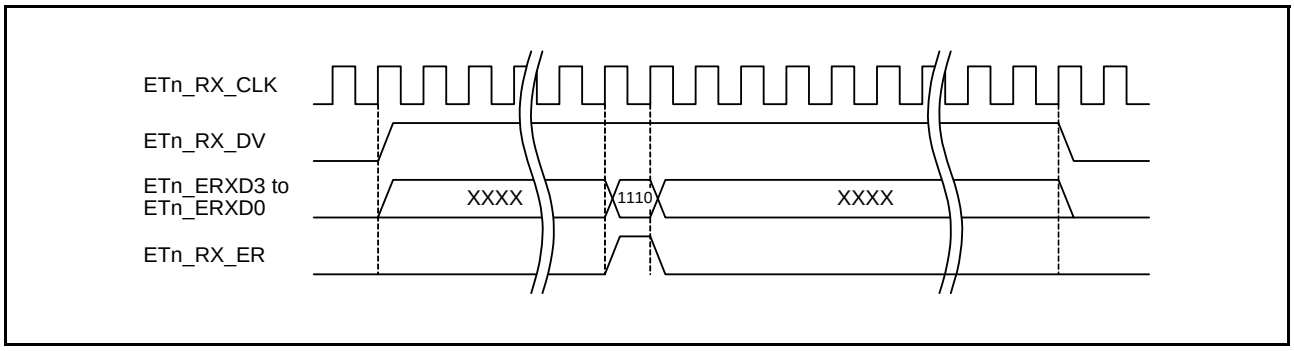


Figure 35.11 MII Fame Receive Timing for False Carrier Notification

35.3.3.2 RMII Frame Timing

The RMII frame timing is shown in Figure 35.12 to Figure 35.14.

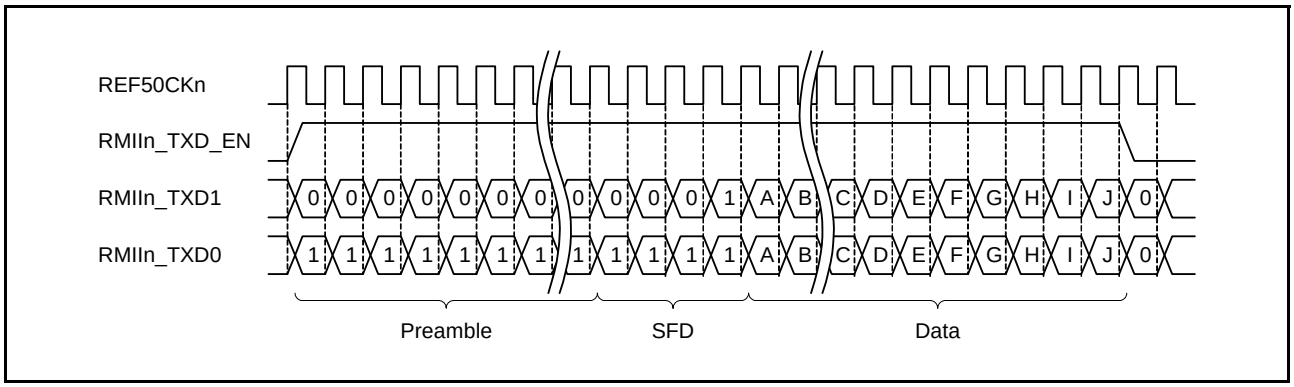


Figure 35.12 RMII Frame Transmit Timing During Normal Transmission

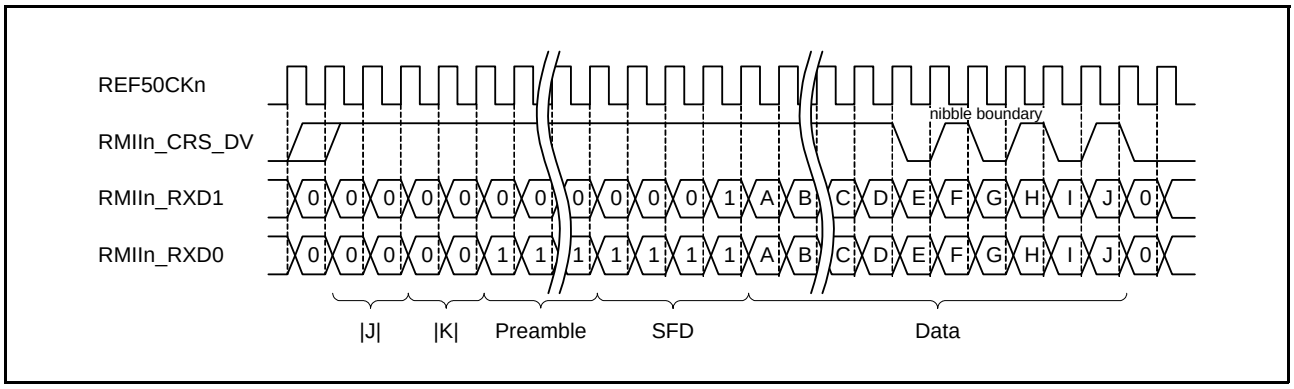


Figure 35.13 RMII Frame Receive Timing During Normal Reception

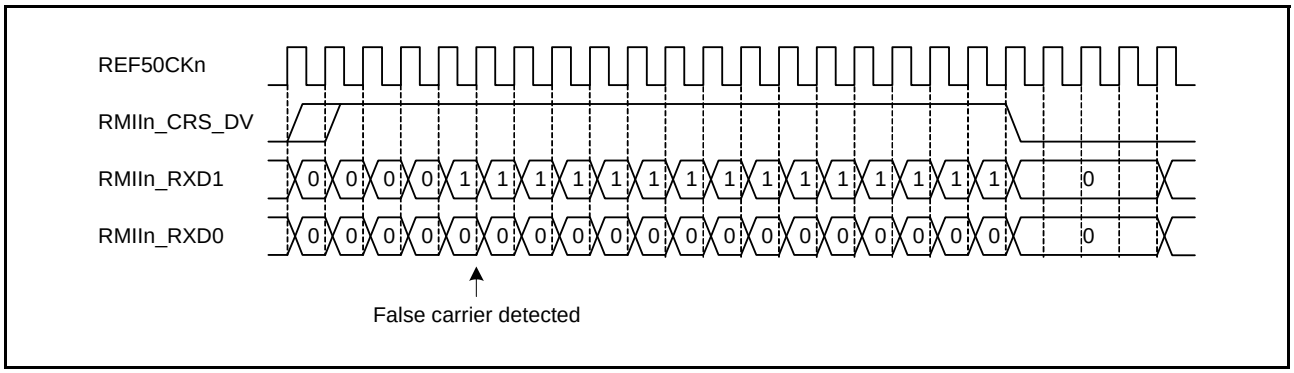


Figure 35.14 RMII Frame Receive Timing When False Carrier Is Detected

35.3.4 Accessing MII/RMII Registers

Use the PIR register to access the MII/RMII registers in the PHY-LSI. Serial data in the MII/RMII management frame format is transmitted and received via the ETn_MDC and ETn_MDIO pins controlled by software.

35.3.4.1 MII/RMII Management Frame Format

Table 35.3 lists the MII/RMII management frame format.

Table 35.3 MII/RMII Management Frame Format

Access Type		MII/RMII Management Frame							
	Item	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
	Number of bits	32	2	2	5	5	2	16	1
Read		1...1	01	10	00001	RRRRR	Z0	DDDDDDDDDDDDDDDD	Z
Write		1...1	01	01	00001	RRRRR	10	DDDDDDDDDDDDDDDD	Z

PRE (preamble): Send 32 consecutive 1's.
ST (start of frame): Send 01b.
OP (operation code): Send 10b for read or 01b for write.
PHYAD (PHY address): Up to 32 PHY-LSIs can be connected to one MAC. PHY-LSIs are selected with these 5 bits.
When the PHY-LSI address is 1, send 00001b.
REGAD (register address): One register is selected from up to 32 registers in the PHY-LSI.
When the register address is 1, send 00001b.
TA (turnaround): 2-bit turnaround time to avoid contention between the register address and data during a read operation
(a) Send 10b during a write operation.
(b) Release the bus for 1 bit during a read operation (Z is output)
(indicated as Z0 because 0 is output from the PHY-LSI on the next clock cycle).
DATA (data): 16-bit data. Sequentially send or receive starting from the MSB.
IDLE (IDLE condition): Wait time to input the next MII/RMII management format
(a) Release the bus during a write operation (Z is output).
(b) No control is required since a bus has already been released during a read operation.

35.3.4.2 MII/RMII Register Access Procedure

Access to the MII/RMII registers includes writing data in 1-bit units, reading data in 1-bit units, and releasing the bus. Figure 35.15 to Figure 35.18 show examples of the MII/RMII register access timing. The access timing differ with the PHY-LSI type.

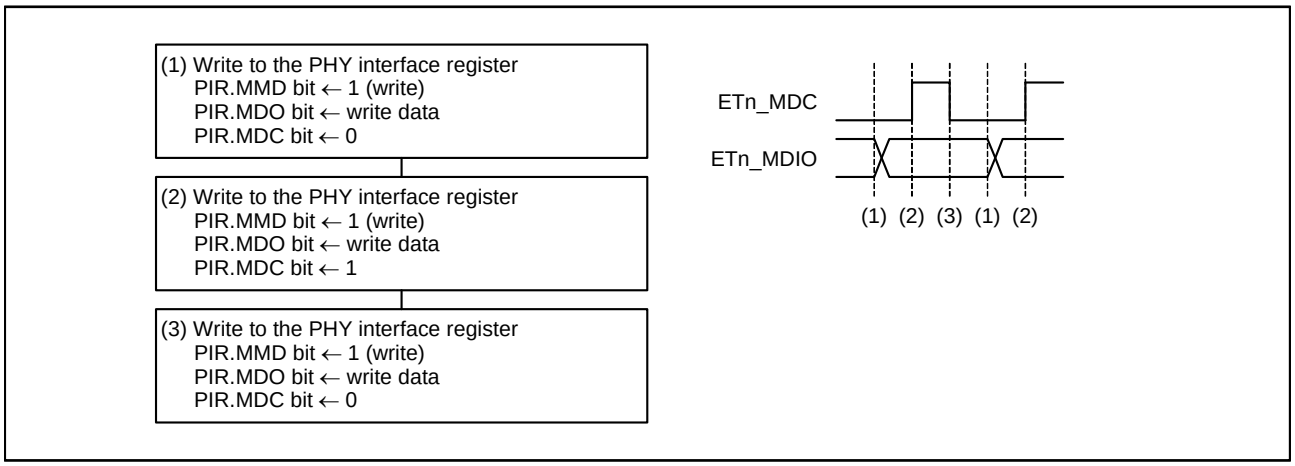


Figure 35.15 1-Bit Data Write Flow

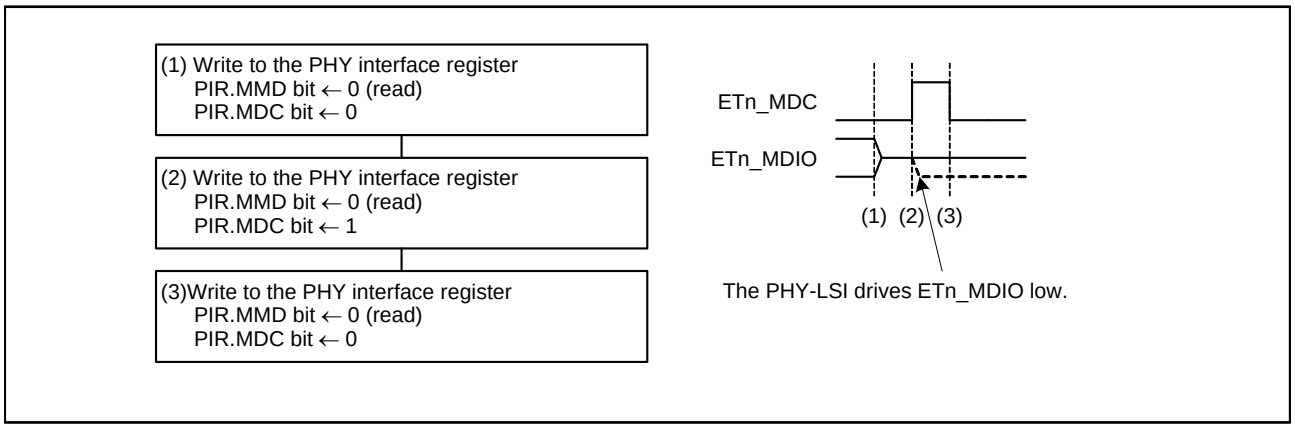


Figure 35.16 Bus Release Flow (TA in Read Operation in Table 35.3)

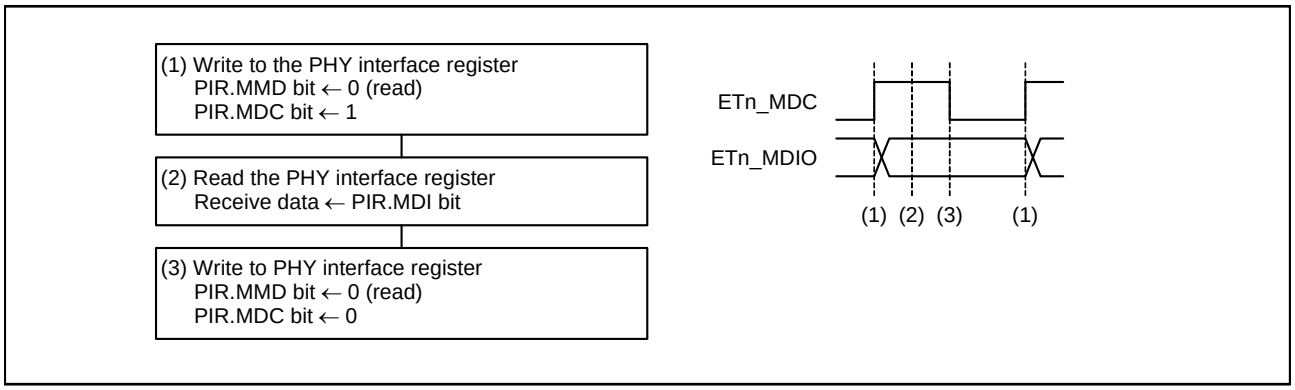


Figure 35.17 1-Bit Data Read Flow

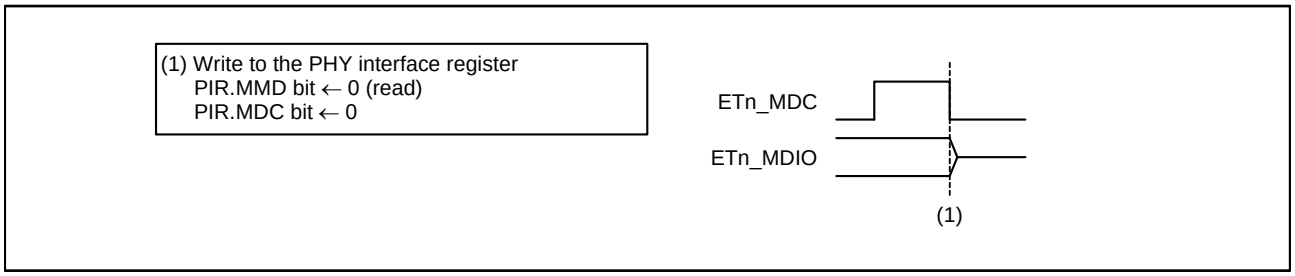


Figure 35.18 Bus Release Flow (IDLE in Write Operation in Table 35.3)

35.3.5 Magic Packet Detection

The ETHERC supports Wake-On-LAN (WOL). WOL is a function to detect a Magic Packet transmitted from a host device or other device and exit a low power consumption state such as sleep mode. When the ETHERC detects a Magic Packet, high is output from the ETn_WOL pin. Write 1 to the EDMACn.EDMR.SWR bit to set the ETn_WOL pin to low.

Since a Magic Packet is transmitted in broadcast mode, the Magic Packet is received regardless of the destination MAC address selected in the format. The ETHERC outputs high from the ETn_WOL pin only when the destination MAC address matches its own MAC address. Refer to the technical documentation provided by Advanced Micro Devices, Inc. for details on the Magic Packet. The following describes an example of the procedure to use WOL in the MCU.

1. Set the ICU to disable the EINTn interrupt request.
2. Set the ECMR.MPDE bit to 1 to enable Magic Packet detection. Set the ECMR.RE bit to 1 to enable reception.
3. Set the ECSIPR.MPDIP bit to 1 to enable notification of the Magic Packet detect interrupt.
4. Set the EDMACn.EESIPR.ECIIP bit to 1 to enable the ETHERC status register source interrupt.
5. Set the ICU to enable the EINTn interrupt request.
6. Change the CPU operating mode to sleep mode or place unused peripherals in the module stop state as needed.
7. When a Magic Packet is detected, an interrupt request is sent to the CPU. High is output from the ETn_WOL pin to notify peripheral devices that the Magic Packet has been detected.

35.3.5.1 Notes on Magic Packet Detection

The ETHERC receives packets, including broadcast packets, even when waiting to receive a Magic Packet. Therefore, receive data may have been stored in the receive FIFO of the EDMAC when a Magic Packet is detected. Also, flags in registers ECSR and EDMACn.EESR may have been changed. When returning to normal operation by detecting a Magic Packet, set the EDMACn.EDMR.SWR bit to 1 to reset the ETHERC and EDMAC.

35.3.6 Adjusting Transmission Efficiency by Changing the IPG

The IPG is a non-transmit period between transmit frames. The ETHERC can change the value of the IPG. Transmission efficiency can be increased or decreased by setting the IPGR register. The typical value of the IPG is specified by the IEEE802.3 standard. When changing the setting, confirm that all devices operate normally in the same network.

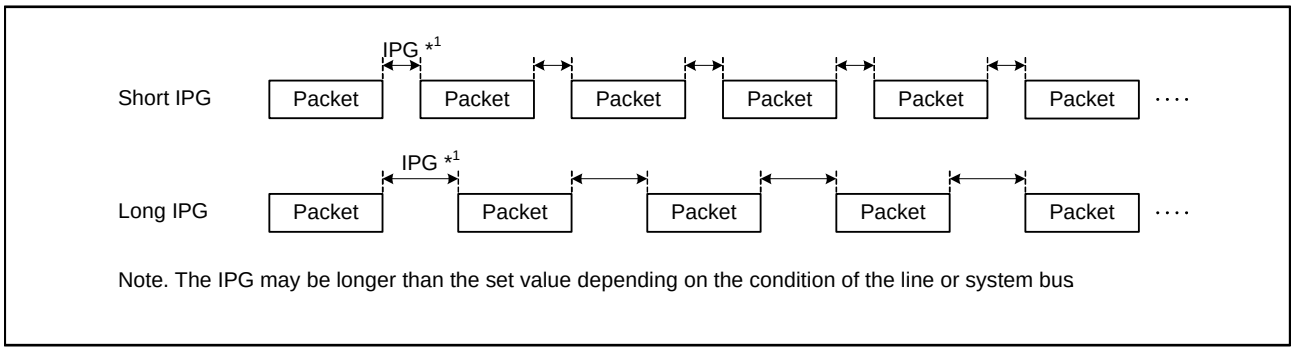


Figure 35.19 Differences in Transmission Efficiency Based on Changes in the IPG

35.3.7 Flow Control

The ETHERC can perform flow control compliant with IEEE802.3x in full-duplex mode, and the receiver and transmitter can be set individually. PAUSE frames can be transmitted automatically or manually.

35.3.7.1 Automatic PAUSE Frame Transmission

When the ECMR.TXF bit is set to 1, automatic PAUSE frame transmission is enabled. A PAUSE frame is automatically transmitted by a PAUSE frame transmit request from the EDMAC. The APR.AP[15:0] bit value is used for the pause_time parameter of the PAUSE frame.

After a PAUSE frame is transmitted, if the EDMAC is still requesting PAUSE frame transmission when the PAUSE time elapses, a PAUSE frame is transmitted again. The maximum number of PAUSE frame retransmissions can be set in the TPAUSER.TPAUSE[15:0] bits. If the maximum number of retransmissions is reached, subsequent PAUSE frames are not transmitted.

Figure 35.20 shows the procedure to set automatic PAUSE frame transmission.

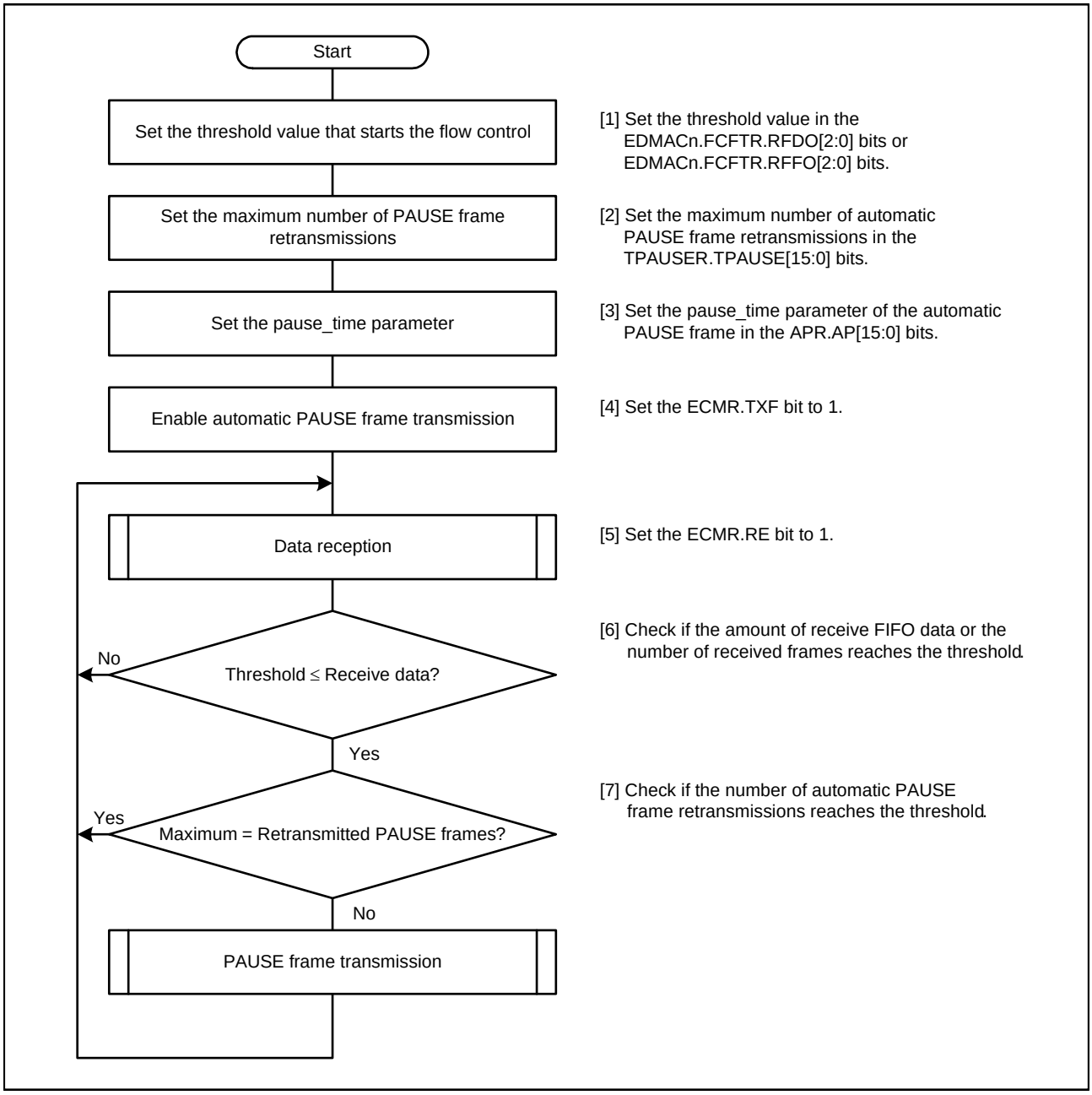


Figure 35.20 Example of Procedure to Set Automatic PAUSE Frame Transmission

35.3.7.2 Manual PAUSE Frame Transmission

A PAUSE frame can be manually transmitted at any time. When writing the pause_time parameter of the PAUSE frame to the MPR.MP[15:0] bits by software, the ETHERC transmits a PAUSE frame once. When transmitting a PAUSE frame more than once, write to the MPR.MP[15:0] bits for each transmission.

35.3.7.3 PAUSE Frame Reception

When setting the ECMR.RXF bit to 1, PAUSE frame detection is enabled. After a PAUSE frame is received, the ETHERC completes transmitting the current frame and waits for the PAUSE time of the received PAUSE frame to elapse before the next frame can be transmitted. Also, the ETHERC increments the RFCF.RPAUSE[7:0] bit value.
However, while waiting for the PAUSE time to elapse, if a PAUSE frame that contains a pause_time parameter of 0 is received and the ECMR.ZPF bit is 1, the ETHERC becomes ready to transmit.

35.4 Interrupts

When a flag in the ECSR register becomes 1 and the corresponding bit in the ECSIPR register is 1, the ETHERC notifies the EDMAC of the status as an interrupt source. After receiving the notification, the EDMAC sets the EDMACn.EESR.ECI flag to 1. When the EDMACn.EESIPR.ECIIP bit is 1, the EDMAC sends an EINTn interrupt request to the CPU. Refer to section 37, DMA Controller for the Ethernet Controller (EDMACa) for details.

35.5 Usage Notes

35.5.1 Conditions for the LCHNG Flag to Become 1

The ECSR.LCHNG flag may become 1 even when the input level of the ETn_LINKSTA pin remains the same. In this case, high is input to the ETn_LINKSTA pin when setting the MPC.PmnPFS register to assign the ETn_LINKSTA signal to a port or when releasing the ETHERC and EDMAC software reset using the EDMACn.EDMR.SWR bit. The ECSR.LCHNG flag becomes 1 because the ETn_LINKSTA signal in the ETHERC is fixed low regardless of the input level to the external pin while the MPC does not assign the ETn_LINKSTA signal or during the ETHERC and EDMAC software reset.
To avoid wrongly generating a link signal change interrupt, clear the ECSR.LCHNG flag and then set the ECSIPR.LCHNGIP bit to 1.

35.5.2 Input to the RMIIn_RX_ER Pin While the RMII is Selected

When the width of a reception error signal received from the PHY-LSI is only one cycle of the REF50CKn clock (50 MHz) while the RMII is selected, the signal is not recognized as an error signal.

35.5.3 Handling Errors in Control Information

When a frame is received with an alignment error or a frame receive error occurs, the EPTPCn.SYSR.INFABT flag may become 1 regardless of use of the EPTPC. In this case, reset the EPTPC, PTPEDMAC, and the corresponding channels ETHERC and EDMAC.

Follow the procedure below to reset these modules:

(1) Write 0000 0001h to the EPTPC.PTRSTR register.
(2) Write 0000 0001h to the PTPEDMAC.EDMR register.
(3) Write 0000 0001h to the EDMACn.EDMR register of the corresponding channels.
(4) Wait for 64 cycles of the PCLKA until initialization is completed.
(5) Write 0000 0000h to the EPTPC.PTRSTR register.

Refer to sections section 36.2.83, PTP Reset Register (PTRSTR) and section 37.2.1, EDMAC Mode Register (EDMR) for details of resets.

36. PTP Module for the Ethernet Controller (EPTPC)

36.1 Overview

This MCU has an on-chip Precision Time Protocol (PTP) module for the Ethernet controller (EPTPC). The module applies the PTP defined in version 2 of the IEEE 1588-2008 standard to handle timing and synchronization between devices. The EPTPC is composed of synchronization frame processing units (SYNFP0 and SYNFP1), a packet relation controller unit (PRC-TC), and a statistical time correction algorithm unit (STCA).
Use the EPTPC in combination with the on-chip Ethernet controller (ETHERC) and the DMA controller for the PTP Ethernet controller (PTPEDMAC).
Table 36.1 lists the EPTPC specifications, and Figure 36.1 shows the EPTPC configuration.

Table 36.1 EPTPC Specifications

Item	Description
Protocol	Compliant with the Precision Time Protocol (PTP) defined in IEEE 1588
Synchronization frame processing units (SYNFP0 and SYNFP1)	• Transmits and receives PTP messages as a master or slave. • The following four clock devices are supported: Ordinary clock (OC) Boundary clock (BC) End-to-end transparent clock (E2E TC) Peer-to-peer transparent clock (P2P TC) • Calculates the meanPathDelay and offsetFromMaster values defined in IEEE 1588. • Capable of generating a master clock. • Hardware filtering of received multicast packets with a MAC address • Capable of hardware filtering in accord with the type of PTP message • Supports PTP message frames in layer 4 (IPv4 and UDP) and layer 2 (Ethernet frames). • Can be used as a normal Ethernet port when time synchronization is not in use.
Packet relation controller unit (PRC-TC)	• Relaying of received data between Ethernet ports 0 and 1 • Setting the same MAC address for Ethernet ports 0 and 1 allows transmission of data from the two ports or from only one of them. • Store-and-forward method or cut-through method selectable for the relaying of packets
Statistical time correction algorithm unit (STCA)	• Frequency of the clock signal supplied to the statistical time correction algorithm unit is selectable as 20, 25, 50, or 100 MHz. • In slave operation, the synchronized state can be indicated from the offsetFromMaster value staying below a previously specified threshold. Additionally, the threshold can be calculated statistically from collected positive and negative gradient values (worst-10 acquisition). • The local clock counter holds corrected time information obtained from a master clock. • The STCA clock can be used as the clock source for generating pulse signals from pulse output timer m (m = 0 to 5). • Peripheral modules such as MTU3 and GPT can be started or stopped on the edge of pulses synchronized with the master clock in response to interrupt requests by the pulse output timer or the output of event signals to the ELC.
Interrupt sources	MINT interrupt • Requested when the state of the individual modules is changed. • Requested on rising edges of the pulse signal generated by the pulse output timer. IPLS interrupt • Requested on rising or falling edges of the pulse signal generated by the previously selected pulse output timer group. • Can be requested on every edge or only once.
Event linking	• An event signal is output to the ELC on a rising or falling edge of the pulse signal generated by the pulse output timer. • An event signal can be output on every edge or only once.

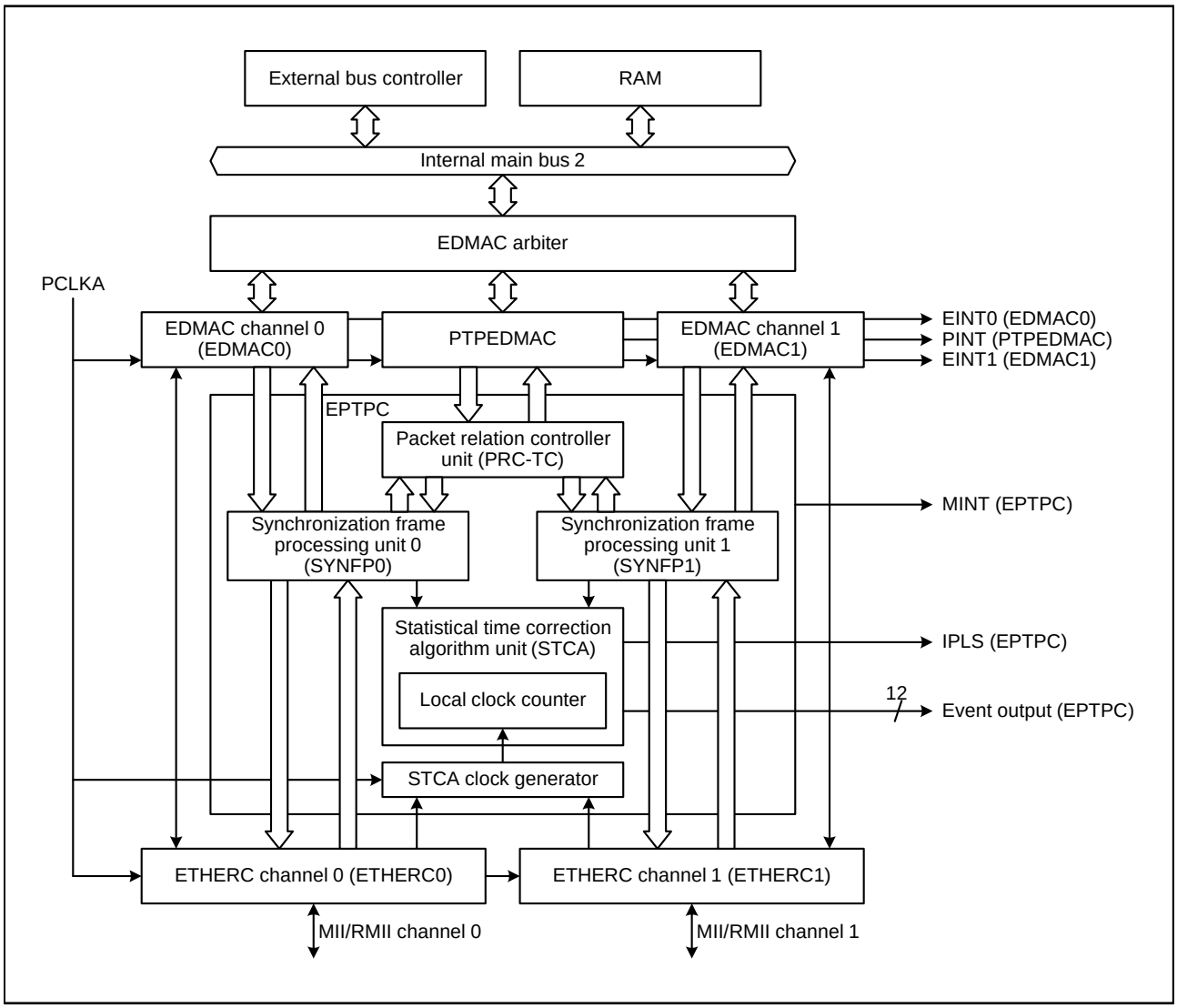


Figure 36.1 EPTPC Configuration

In this section, channels may not be mentioned in overall descriptions of modules that have multiple channels. Examples of the notation are listed in Table 36.2.

Table 36.2 Examples of the Notation

Module Name	Channel	Meaning
SYNFPn module	n = 0, 1	Synchronization processing units 0 and 1 (SYNFP0 and SYNFP1)
Pulse output timer m	m = 0 to 5	Pulse output timer channels 0 to 5

36.1.1 Combination of Clock Device and Ethernet Port

The EPTPC supports operation as three types of clock device: an ordinary clock (OC), boundary clock (BC), or transparent clock (TC). Furthermore, it supports both end-to-end (E2E) and peer-to-peer (P2P) operation for all three clock devices.
Available combinations for usage of Ethernet ports 0 and 1 are listed in Table 36.3.

Table 36.3 Combination of Clock Device and Ethernet Port

Clock Device	Ethernet Port 0		Ethernet Port 1	
No control by EPTPC	PTP packets are not handled.			
Ordinary Clock (OC) Only Ethernet port 0 is used for handling PTP packets	Master	End-to-end (E2E)	PTP packets are not handled.	
		Peer-to-peer (P2P)		
	Slave	E2E		
		P2P		
Ordinary Clock (OC) Only Ethernet port 1 is used for handling PTP packets	PTP packets are not handled.		Master	E2E
				P2P
			Slave	E2E
				P2P
Boundary clock (OC)	Master	E2E	Master	E2E
		E2E		P2P
		P2P		E2E
		P2P		P2P
	Master	E2E	Slave	E2E
		E2E		P2P
		P2P		E2E
		P2P		P2P
	Slave	E2E	Master	E2E
		E2E		P2P
		P2P		E2E
		P2P		P2P
Transparent clock (TC)	E2E TC			
	P2P TC			

36.1.2 Frame Format of PTP Messages

The frame format of PTP messages can be selected from the two types by setting the FORM0 and FORM1 bits in the SYNFP frame format setting register (SYFORMR).
Figure 36.2 shows the PTP message formats for transmission and reception by the EPTPC.

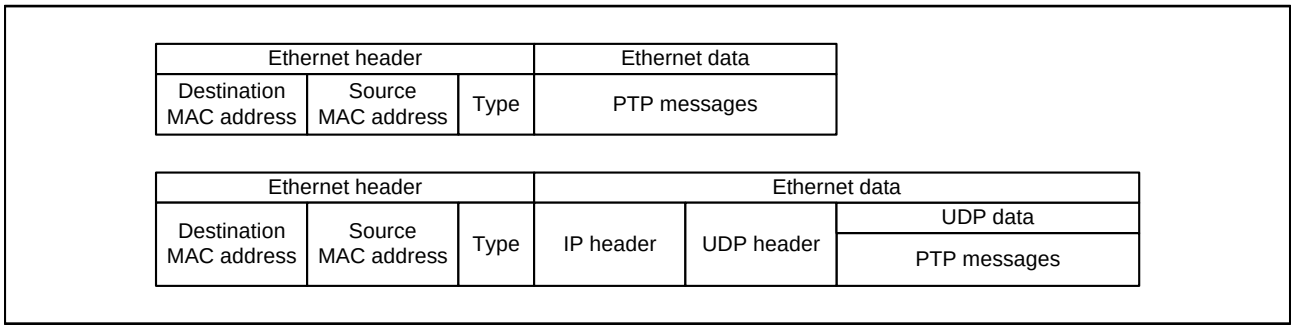


Figure 36.2 Frame Format of PTP Messages

The EPTPC module is capable of transmitting PTP messages. When it sends a PTP message, multicast addresses defined by IEEE 1588 are normally specified as the destination MAC address and IP address, depending on the type of the PTP message to be sent.
Furthermore, the port number when a PTP message is encapsulated for use with the UDP must also be specified in accord with the type of message as stipulated by IEEE 1588.
Table 36.4 gives a summary of the information to specify the Ethernet frame format for PTP messages.

Table 36.4 Type of PTP Message for Multicast and Information to Specify the Ethernet Frame Format

Type of PTP Message			Ethernet II Frame Format (SYFORMR.FORM0 Bit = 0)		IP Address (IPv4)	UDP Port No. *1
			MAC Address	Ethertype		
PTP-primary	Event messages	Sync	01-1B-19-00-00-00	88F7h	224.0.1.129	319
		Delay_Req				
PTP-pdelay		Pdelay_Req	01-80-C2-00-00-0E		224.0.0.107	
		Pdelay_Resp				
	General messages	Pdelay_Resp_Follow_Up				320
PTP-primary		Announce	01-1B-19-00-00-00		224.0.1.129	
		Follow_Up				
		Delay_Resp				
		Signaling				
		Management				

Note 1. The port number must be specified only when a PTP message is encapsulated for use with the UDP (i.e. the SYFORMR.FORM1 bit = 1).

36.1.3 Type of PTP Message and Details of Processing

Table 36.5 and Table 36.6 respectively give details of processing by the EPTPC for receiving and transmitting PTP messages.

Table 36.5 List of PTP Messages for Reception by the EPTPC

Message Type	Message	Details of Processing
Event	Sync	Calculates the value of offsetFromMaster if twoStepFlag in flagField is FALSE.
	Delay_Req	Responds to Delay_Resp.
	Pdelay_Req	Responds to Pdelay_Resp.
	Pdelay_Resp	Calculates the value of meanPathDelay if twoStepFlag in flagField is FALSE.
General	Announce	—
	Follow_Up	Calculates the value of offsetFromMaster if twoStepFlag in flagField of the most recently received Sync message was TRUE and the value of meanPathDelay is fixed.
	Delay_Resp	Calculates the value of meanPathDelay.
	Pdelay_Resp_Follow_Up	Calculates the value of meanPathDelay if twoStepFlag in flagField of the most recently received Pdelay_Resp message was TRUE.
	Management	—
	Signaling	—

Table 36.6 List of PTP Messages for Transmission by the EPTPC

Message Type	Message	Details of Processing
Event	Sync	Sync messages are transmitted at the fixed interval specified by the SYTLIR.SYNC[7:0] bits.
	Delay_Req	Transmission proceeds with an interval from 0 to twice the interval set by the value of the SYTLIR.DREQ[7:0] bits and determined by a random number.
	Pdelay_Req	Pdelay_Req messages are transmitted at the fixed interval specified by the SYTLIR.DREQ[7:0] bits.
	Pdelay_Resp	Transmission of responses to Pdelay_Req.
General	Announce	Announce messages are transmitted at the fixed interval specified by the SYTLIR.ANCE[7:0] bits.
	Follow_Up	—
	Delay_Resp	Transmission of responses to Delay_Req
	Pdelay_Resp_Follow_Up	—
	Management	—
	Signaling	—

36.2 Register Descriptions

36.2.1 MINT Interrupt Source Status Register (MIESR)

Address(es): EPTPC.MIESR 000C 4000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	CYC5	CYC4	CYC3	CYC2	CYC1	CYC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PRC	SY1	SY0	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ST	STCA Status Flag	0: No change in the state of the STCA module 1: A change in the state of the STCA module	R
b1	SY0	SYNFP0 Status Flag	0: No change in the state of the SYNFP0 module 1: A change in the state of the SYNFP0 module	R
b2	SY1	SYNFP1 Status Flag	0: No change in the state of the SYNFP1 module 1: A change in the state of the SYNFP1 module	R
b3	PRC	PRC-TC Status Flag	0: No change in the state of the PRC-TC module 1: A change in the state of the PRC-TC module	R
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	CYC0	Pulse Output Timer 0 Rising Edge Detection Flag	0: A rising edge in the periodic pulse signal from pulse output timer 0 is not detected. 1: A rising edge in the periodic pulse signal from pulse output timer 0 is detected.	R/W*1
b17	CYC1	Pulse Output Timer 1 Rising Edge Detection Flag	0: A rising edge in the periodic pulse signal from pulse output timer 1 is not detected. 1: A rising edge in the periodic pulse signal from pulse output timer 1 is detected.	R/W*1
b18	CYC2	Pulse Output Timer 2 Rising Edge Detection Flag	0: A rising edge in the periodic pulse signal from pulse output timer 2 is not detected. 1: A rising edge in the periodic pulse signal from pulse output timer 2 is detected.	R/W*1
b19	CYC3	Pulse Output Timer 3 Rising Edge Detection Flag	0: A rising edge in the periodic pulse signal from pulse output timer 3 is not detected. 1: A rising edge in the periodic pulse signal from pulse output timer 3 is detected.	R/W*1
b20	CYC4	Pulse Output Timer 4 Rising Edge Detection Flag	0: A rising edge in the periodic pulse signal from pulse output timer 4 is not detected. 1: A rising edge in the periodic pulse signal from pulse output timer 4 is detected.	R/W*1
b21	CYC5	Pulse Output Timer 5 Rising Edge Detection Flag	0: A rising edge in the periodic pulse signal from pulse output timer 5 is not detected. 1: A rising edge in the periodic pulse signal from pulse output timer 5 is detected.	R/W*1
b31 to b22	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writing 1 clears the flag. Writing 0 does not affect the flag value.

The MIESR register indicates changes in the states of the STCA, SYNFPn, and PRC-TC modules which act as sources of MINT interrupts and the detection of rising edges of pulse output timers m (m = 0 to 5).

For the MINT interrupt, see section 36.4, Interrupts.

ST Flag (STCA Status Flag)

This flag indicates changes in the state of the STCA module.
[Setting condition]

- A change in the state of a flag of the STSR register for which the setting in the STIPR register enables notification

[Clearing conditions]
When any of the following conditions is met:

- The flags in the STSR register are all 0.
- The bits in the STIPR register are all 0.
- There is a bit set to 1 in the STIPR register, but the corresponding flag in the STSR register is 0.

SYn Flag (SYNFPn Status Flag)

This flag indicates changes in the state of the SYNFPn module.
[Setting condition]

- A change in the state of a flag of the SYSR register for which the setting in the SYIPR register enables notification

[Clearing conditions]
When any of the following conditions is met:

- The flags in the SYSR register are all 0.
- The bits in the SYIPR register are all 0.
- There is a bit set to 1 in the SYIPR register, but the corresponding flag in the SYSR register is 0.

PRC Flag (PRC-TC Status Flag)

This flag indicates changes in the state of the PRC-TC module.
[Setting condition]

- A change in the state of a flag of the PRSR register for which the setting in the PRIPR register enables notification

[Clearing conditions]
When any of the following conditions is met:

- The flags in the PRSR register are all 0.
- The bits in the PRIPR register are all 0.
- There is a bit set to 1 in the PRIPR register, but the corresponding bit in the PRSR register is 0.

CYCm Flag (Pulse Output Timer m Rising Edge Detection Flag)

This flag indicates detection of a rising edge in the periodic pulse signal produced by the corresponding pulse output timer m (m = 0 to 5).
[Setting condition]

- Detection of a rising edge in the periodic pulse signal produced by a pulse output timer for which the setting in the MITSELR register enables notification

[Clearing condition]

- 1 being written to this flag.
  After the flag is cleared to 0, it is set to 1 again on detection of a rising edge in the periodic pulse signal from the corresponding pulse output timer.

36.2.2 MINT Interrupt Request Enable Register (MIEIPR)

Address(es): EPTPC.MIEIPR 000C 4004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	CYC5	CYC4	CYC3	CYC2	CYC1	CYC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PR	SY1	SY0	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ST	STCA Status Interrupt Request Enable	0: Disables the generation of MINT interrupt requests by the STCA status flag. 1: Enables the generation of MINT interrupt requests by the STCA status flag.	R/W
b1	SY0	SYNFP0 Status Interrupt Request Enable	0: Disables the generation of MINT interrupt requests by the SYNFP0 status flag. 1: Enables the generation of MINT interrupt requests by the SYNFP0 status flag.	R/W
b2	SY1	SYNFP1 Status Interrupt Request Enable	0: Disables the generation of MINT interrupt requests by the SYNFP1 status flag. 1: Enables the generation of MINT interrupt requests by the SYNFP1 status flag.	R/W
b3	PR	PRC-TC Status Interrupt Request Enable	0: Disables the generation of MINT interrupt requests by the PRC-TC status flag. 1: Enables the generation of MINT interrupt requests by the PRC-TC status flag.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	CYC0	Pulse Output Timer 0 Rising Edge Detection Interrupt Request Enable	0: Disables the generation of MINT interrupt requests in response to detection of a rising edge of pulse output timer 0. 1: Enables the generation of MINT interrupt requests in response to detection of a rising edge of pulse output timer 0.	R/W
b17	CYC1	Pulse Output Timer 1 Rising Edge Detection Interrupt Request Enable	0: Disables the generation of MINT interrupt requests in response to detection of a rising edge of pulse output timer 1. 1: Enables the generation of MINT interrupt requests in response to detection of a rising edge of pulse output timer 1.	R/W
b18	CYC2	Pulse Output Timer 2 Rising Edge Detection Interrupt Request Enable	0: Disables the generation of MINT interrupt requests in response to detection of a rising edge of pulse output timer 2. 1: Enables the generation of MINT interrupt requests in response to detection of a rising edge of pulse output timer 2.	R/W
b19	CYC3	Pulse Output Timer 3 Rising Edge Detection Interrupt Request Enable	0: Disables the generation of MINT interrupt requests in response to detection of a rising edge of pulse output timer 3. 1: Enables the generation of MINT interrupt requests in response to detection of a rising edge of pulse output timer 3.	R/W
b20	CYC4	Pulse Output Timer 4 Rising Edge Detection Interrupt Request Enable	0: Disables the generation of MINT interrupt requests in response to detection of a rising edge of pulse output timer 4. 1: Enables the generation of MINT interrupt requests in response to detection of a rising edge of pulse output timer 4.	R/W
b21	CYC5	Pulse Output Timer 5 Rising Edge Detection Interrupt Request Enable	0: Disables the generation of MINT interrupt requests in response to detection of a rising edge of pulse output timer 5. 1: Enables the generation of MINT interrupt requests in response to detection of a rising edge of pulse output timer 5.	R/W
b31 to b22	—	Reserved	These bits are read as 0. The write value should be 0.	R

The MIEIPR register is used to enable or disable the generation of MINT interrupt requests when MINT interrupt source conditions are satisfied.

36.2.3 ELC Output/IPLS Interrupt Request Enable Register (ELIPPR)

Address(es): EPTPC.ELIPPR 000C 4010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	PLSN	—	—	—	—	—	—	—	PLSP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CYCN5	CYCN4	CYCN3	CYCN2	CYCN1	CYCN0	—	—	CYCP5	CYCP4	CYCP3	CYCP2	CYCP1	CYCP0
Value after reset:	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	CYCP0	Pulse Output Timer 0 Rising Edge Detection Event Output Enable	0: Rising edges of the signal from pulse output timer 0 are not conveyed to the ELC as event signals. 1: Rising edges of the signal from pulse output timer 0 are conveyed to the ELC as event signals.	R/W
b1	CYCP1	Pulse Output Timer 1 Rising Edge Detection Event Output Enable	0: Rising edges of the signal from pulse output timer 1 are not conveyed to the ELC as event signals. 1: Rising edges of the signal from pulse output timer 1 are conveyed to the ELC as event signals.	R/W
b2	CYCP2	Pulse Output Timer 2 Rising Edge Detection Event Output Enable	0: Rising edges of the signal from pulse output timer 2 are not conveyed to the ELC as event signals. 1: Rising edges of the signal from pulse output timer 2 are conveyed to the ELC as event signals.	R/W
b3	CYCP3	Pulse Output Timer 3 Rising Edge Detection Event Output Enable	0: Rising edges of the signal from pulse output timer 3 are not conveyed to the ELC as event signals. 1: Rising edges of the signal from pulse output timer 3 are conveyed to the ELC as event signals.	R/W
b4	CYCP4	Pulse Output Timer 4 Rising Edge Detection Event Output Enable	0: Rising edges of the signal from pulse output timer 4 are not conveyed to the ELC as event signals. 1: Rising edges of the signal from pulse output timer 4 are conveyed to the ELC as event signals.	R/W
b5	CYCP5	Pulse Output Timer 5 Rising Edge Detection Event Output Enable	0: Rising edges of the signal from pulse output timer 5 are not conveyed to the ELC as event signals. 1: Rising edges of the signal from pulse output timer 5 are conveyed to the ELC as event signals.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	CYCN0	Pulse Output Timer 0 Falling Edge Detection Event Output Enable	0: Falling edges of the signal from pulse output timer 0 are not conveyed to the ELC as event signals. 1: Falling edges of the signal from pulse output timer 0 are conveyed to the ELC as event signals.	R/W
b9	CYCN1	Pulse Output Timer 1 Falling Edge Detection Event Output Enable	0: Falling edges of the signal from pulse output timer 1 are not conveyed to the ELC as event signals. 1: Falling edges of the signal from pulse output timer 1 are conveyed to the ELC as event signals.	R/W
b10	CYCN2	Pulse Output Timer 2 Falling Edge Detection Event Output Enable	0: Falling edges of the signal from pulse output timer 2 are not conveyed to the ELC as event signals. 1: Falling edges of the signal from pulse output timer 2 are conveyed to the ELC as event signals.	R/W

Bit	Symbol	Bit Name	Description	R/W
b11	CYCN3	Pulse Output Timer 3 Falling Edge Detection Event Output Enable	0: Falling edges of the signal from pulse output timer 3 are not conveyed to the ELC as event signals. 1: Falling edges of the signal from pulse output timer 3 are conveyed to the ELC as event signals.	R/W
b12	CYCN4	Pulse Output Timer 4 Falling Edge Detection Event Output Enable	0: Falling edges of the signal from pulse output timer 4 are not conveyed to the ELC as event signals. 1: Falling edges of the signal from pulse output timer 4 are conveyed to the ELC as event signals.	R/W
b13	CYCN5	Pulse Output Timer 5 Falling Edge Detection Event Output Enable	0: Falling edges of the signal from pulse output timer 5 are not conveyed to the ELC as event signals. 1: Falling edges of the signal from pulse output timer 5 are conveyed to the ELC as event signals.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	PLSP	Pulse Output Timer Rising Edge Detection IPLS Interrupt Request Enable	0: Disables IPLS interrupt requests due to rising edges of signals from the selected pulse output timer. 1: Enables IPLS interrupt requests due to rising edges of signals from the selected pulse output timer.	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	PLSN	Pulse Output Timer Falling Edge Detection IPLS Interrupt Request Enable	0: Disables IPLS interrupt requests due to falling edges of signals from the selected pulse output timer. 1: Enables IPLS interrupt requests due to falling edges of signals from the selected pulse output timer.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R

The ELIPPR register determines whether or not rising and falling edges of the periodic pulse signals produced by pulse output timers m are output to the ELC. The register also enables and disables IPLS interrupts due to rising or falling edges of signals from the pulse output timer selected in the IPTSELR register.

Peripheral modules such as the MTU3 or GPT can be controlled with the clock synchronized by the PTP by using the ELC linking function to set a periodic pulse generated by pulse output timer m as a trigger for operations of the peripheral module.

The ELIPACR register can be used to set up the one-time-only output of event signals to the ELC or of IPLS interrupt requests.

For the IPLS interrupt, see section 36.4, Interrupts.

36.2.4 ELC Output/IPLS Interrupt Enable Automatic Clearing Register (ELIPACR)

Address(es): EPTPC.ELIPACR 000C 4014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	PLSN	—	—	—	—	—	—	—	PLSP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CYCN5	CYCN4	CYCN3	CYCN2	CYCN1	CYCN0	—	—	CYCP5	CYCP4	CYCP3	CYCP2	CYCP1	CYCP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CYCP0	ELIPPR.CYCP0 Bit Automatic Clearing	0: Disables automatic clearing of the enable bit for the output of rising edges of pulse output timer 0. 1: Enables automatic clearing of the enable bit for the output of rising edges of pulse output timer 0.	R/W
b1	CYCP1	ELIPPR.CYCP1 Bit Automatic Clearing	0: Disables automatic clearing of the enable bit for the output of rising edges of pulse output timer 1. 1: Enables automatic clearing of the enable bit for the output of rising edges of pulse output timer 1.	R/W
b2	CYCP2	ELIPPR.CYCP2 Bit Automatic Clearing	0: Disables automatic clearing of the enable bit for the output of rising edges of pulse output timer 2. 1: Enables automatic clearing of the enable bit for the output of rising edges of pulse output timer 2.	R/W
b3	CYCP3	ELIPPR.CYCP3 Bit Automatic Clearing	0: Disables automatic clearing of the enable bit for the output of rising edges of pulse output timer 3. 1: Enables automatic clearing of the enable bit for the output of rising edges of pulse output timer 3.	R/W
b4	CYCP4	ELIPPR.CYCP4 Bit Automatic Clearing	0: Disables automatic clearing of the enable bit for the output of rising edges of pulse output timer 4. 1: Enables automatic clearing of the enable bit for the output of rising edges of pulse output timer 4.	R/W
b5	CYCP5	ELIPPR.CYCP5 Bit Automatic Clearing	0: Disables automatic clearing of the enable bit for the output of rising edges of pulse output timer 5. 1: Enables automatic clearing of the enable bit for the output of rising edges of pulse output timer 5.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	CYCN0	ELIPPR.CYCN0 Bit Automatic Clearing	0: Disables automatic clearing of the enable bit for the output of falling edges of pulse output timer 0. 1: Enables automatic clearing of the enable bit for the output of falling edges of pulse output timer 0.	R/W
b9	CYCN1	ELIPPR.CYCN1 Bit Automatic Clearing	0: Disables automatic clearing of the enable bit for the output of falling edges of pulse output timer 1. 1: Enables automatic clearing of the enable bit for the output of falling edges of pulse output timer 1.	R/W
b10	CYCN2	ELIPPR.CYCN2 Bit Automatic Clearing	0: Disables automatic clearing of the enable bit for the output of falling edges of pulse output timer 2. 1: Enables automatic clearing of the enable bit for the output of falling edges of pulse output timer 2.	R/W
b11	CYCN3	ELIPPR.CYCN3 Bit Automatic Clearing	0: Disables automatic clearing of the enable bit for the output of falling edges of pulse output timer 3. 1: Enables automatic clearing of the enable bit for the output of falling edges of pulse output timer 3.	R/W

Bit	Symbol	Bit Name	Description	R/W
b12	CYCN4	ELIPPR.CYCN4 Bit Automatic Clearing	0: Disables automatic clearing of the enable bit for the output of falling edges of pulse output timer 4. 1: Enables automatic clearing of the enable bit for the output of falling edges of pulse output timer 4.	R/W
b13	CYCN5	ELIPPR.CYCN5 Bit Automatic Clearing	0: Disables automatic clearing of the enable bit for the output of falling edges of pulse output timer 5. 1: Enables automatic clearing of the enable bit for the output of falling edges of pulse output timer 5.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	PLSP	ELIPPR.PLSP Bit Automatic Clearing	0: Disables automatic clearing of the enable bit for IPLS interrupt requests in response to detection of rising edges of the pulse output timer. 1: Enables automatic clearing of the enable bit for IPLS interrupt requests in response to detection of rising edges of the pulse output timer.	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	PLSN	ELIPPR.PLSN Bit Automatic Clearing	0: Disables automatic clearing of the enable bit for IPLS interrupt requests in response to detection of falling edges of the pulse output timer. 1: Enables automatic clearing of the enable bit for IPLS interrupt requests in response to detection of falling edges of the pulse output timer.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R

The ELIPACR register is used to enable one-time output of each event to the ELC or each IPLS interrupt request triggered by detecting edges of the periodic pulses of pulse output timer m.
Normally, an event is output to the ELC or an IPLS interrupt request is generated at each edge of the periodic pulses of pulse output timer m while the corresponding bit in the ELIPPR register is 1 (enabled),
When a bit in the ELIPPR register is 1 while the corresponding bit in the ELIPACR register is also 1, the bit in the ELIPPR register automatically becomes 0 when the event signal for the ELC or IPLS interrupt request is generated.
For the IPLS interrupt, see section 36.4, Interrupts.

36.2.5 STCA Status Register (STSR)

Address(es): EPTPC.STSR 000C 4040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	W10D	SYNTOUT	—	SYNCOUT	SYNC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SYNC	Synchronized State Detection Flag	0: Synchronization is not detected. 1: Synchronization is detected.	R/W*1
b1	SYNCOUT	Synchronization Loss Detection Flag	0: A loss of synchronization is not detected. 1: A loss of synchronization is detected.	R/W*1
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R
b3	SYNTOUT	Sync Message Reception Timeout Detection Flag	0: A Sync message reception timeout is not detected. 1: A Sync message reception timeout is detected.	R/W*1
b4	W10D	Worst 10 Acquisition Completion Flag	0: The worst 10 values have not been acquired yet. 1: The worst 10 values have been acquired.	R/W*1
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: When the SYNSTARTR.STR bit is 0, the value of the corresponding flag remains.
Note 1. Writing 1 clears the flag. Writing 0 does not affect the flag value.

The STSR register indicates the state of the STCA module.

SYNC Flag (Synchronized State Detection Flag)

This flag indicates that synchronization has occurred more than the number of times specified by the STMR.SYTH[3:0] bits in succession when the STMR.ALEN0 bit is 1. When the STMR.ALEN0 bit is 0, the SYNC flag is not set to 1 even if synchronization has occurred more than the specified number of times in succession.

SYNCOUT Flag (Synchronization Loss Detection Flag)

This flag indicates that loss of synchronization has occurred more than the number of times specified by the STMR.DVTH[3:0] bits in succession when the STMR.ALEN0 bit is 1.
Since the time is not synchronized immediately after time synchronization is started (SYNSTARTR.STR bit is set to 1), the SYNCOUT becomes 1 regardless of the STMR.ALEN0 bit setting. When using the SYNTOUT flag, set the SYNTOUT flag to 0 immediately after starting time synchronization.
When the STMR.ALEN0 bit is 0, the SYNCOUT flag does not become 1 even if loss of synchronization has occurred more than the specified number of times in succession after time synchronization is started and the SYNTOUT flag is immediately set to 0.

SYNTOUT Flag (Sync Message Reception Timeout Detection Flag)

This flag indicates that a Sync message has not been received during the period specified by the SYNTOR register when the STMR.ALEN1 bit is 1. The SYNTOUT flag becomes 1 immediately if time synchronization is started (SYNSTARTR.STR bit is set to 1) when no Sync message has been received after the EPTPC starts. When using the SYNTOUT flag, set the SYNTOUT flag to 0 immediately after starting time synchronization.

W10D Flag (Worst 10 Acquisition Completion Flag)

This flag indicates that acquisition of the worst 10 has been completed.

36.2.6 STCA Status Notification Enable Register (STIPR)

Address(es): EPTPC.STIPR 000C 4044h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	W10D	SYNTOUT	—	SYNCOUT	SYNC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SYNC	SYNC Status Notification Enable	0: Disables notification of the state of STSR.SYNC. 1: Enables notification of the state of STSR.SYNC.	R/W
b1	SYNCOUT	SYNCOUT Status Notification Enable	0: Disables notification of the state of STSR.SYNCOUT. 1: Enables notification of the state of STSR.SYNCOUT.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R
b3	SYNTOUT	SYNTOUT Status Notification Enable	0: Disables notification of the state of STSR.SYNTOUT. 1: Enables notification of the state of STSR.SYNTOUT.	R/W
b4	W10D	W10D Status Notification Enable	0: Disables notification of the state of STSR.W10D. 1: Enables notification of the state of STSR.W10D.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R

The STIPR register specifies whether the MIESR.ST flag does or does not reflect changes in the state of the STCA module.

36.2.7 STCA Clock Frequency Setting Register (STCFR)

Address(es): EPTPC.STCFR 000C 4050h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STCF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	STCF[1:0]	STCA Clock Frequency	b1 b0 0 0: 20 MHz 0 1: 25 MHz 1 0: 50 MHz 1 1: 100 MHz	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: Set a value in this register before starting the EDMAC, ETHERC, and PTPEDMAC. Do not change the settings during operations.

The STCFR register is used to select the frequency of the clock source for the STCA module (STCA clock).
The setting in the STCFR register must be for the same frequency as that selected by the setting of the STCSELR register.

STCF[1:0] Bits (STCA Clock Frequency)

These bits select the frequency of the STCA clock.
To enable synchronous control in compliance with IEEE 1588, the STCA clock frequency must be selected from 20 MHz, 25 MHz, 50 MHz, or 100 MHz. If the frequency selected by these bits differs from the clock frequency actually input to the STCA module, operation is not guaranteed.

36.2.8 STCA Operating Mode Register (STMR)

Address(es): EPTPC.STMR 000C 4054h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	ALEN1	ALEN0	—	—	—	—	DVTH[3:0]				SYTH[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	W10S	—	CMOD	—	—	—	—	—	WINT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	WINT[7:0]	Worst 10 Acquisition Time	00h: The worst 10 values are not acquired. 01h: Sync message reception: 1 time : FFh: Sync message reception: 255 times	R/W
b12 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R
b13	CMOD	Time Synchronization Correction Mode	0: Mode 1 1: Mode 2	R/W
b14	—	Reserved	This bit is read as 0. The write value should be 0.	R
b15	W10S	Worst 10 Acquisition Control Select	0: Measurement is started by hardware and the value acquired in the PW10VR or MW10R register is used as the limit for filtering. 1: Measurement is started by the GETW10R.GW10 bit. Also, the value set in the PLIMITR or MLIMITR register is used as the limit for filtering.	R/W
b19 to b16	SYTH[3:0]	Synchronized State Detection Threshold Setting	0h: None *1 1h: 1 time : Fh: 15 times	R/W
b23 to b20	DVTH[3:0]	Synchronization Loss Detection Threshold Setting	0h: None *2 1h: 1 time : Fh: 15 times	R/W
b27 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R
b28	ALEN0	Alarm Detection Enable 0	0: The STSR.SYNC or SYNCOUT flag is not set to 1 on detection of synchronization or loss of synchronization. 1: The STSR.SYNC or SYNCOUT flag is set to 1 on detection of synchronization or loss of synchronization.	R/W
b29	ALEN1	Alarm Detection Enable 1	0: The STSR.SYNTOUT flag is not set to 1 on detection of the Sync message reception timeout interrupt. 1: The STSR.SYNTOUT flag is not set to 1 on detection of the Sync message reception timeout interrupt.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. The STSR.SYNC flag does not become 1 regardless of the ALEN0 bit setting.
Note 2. The STSR.SYNCOUT flag does not become 1 regardless of the ALEN0 bit setting.

The STMR register is used to set the operating mode of the STCA module.

WINT[7:0] Bits (Worst 10 Acquisition Time)

These bits set the time for acquiring the worst 10 gradients (the number of times Sync messages are received). Usually, it is recommended to set the number of times of Sync message receptions to 32 or more.

CMOD Bit (Time Synchronization Correction Mode)

Mode 1 or mode 2 can be selected to correct the local time information when the EPTPC operates as a slave clock. Select the appropriate mode in consideration of system configuration, etc.
Table 36.7 is a summary of the two individual correction modes.

Table 36.7 Types and Features of Correction Mode

Correction Mode	Function	Features	Notes
Mode 1	Mode for correcting the counter every Sync message reception by using the current offsetFromMaster (operation is in mode 1 after the start of correction, and then shifts to the specified mode).	The time information of the master clock is set as the local time information at a time.	Synchronization cannot be guaranteed if calculating offsetFromMaster is not possible (e.g. packets are temporarily being discarded due to a failure of communications).
Mode 2	Mode where the gradient value calculated from offsetFromMaster (worst-10 control) is retained and used in correcting the local time information so that it approximates the time information of the master clock.	Even if calculating offsetFromMaster is not possible, a certain level of synchronization can be guaranteed in this mode (the counter is still corrected from the gradient information).	Establishing synchronization takes longer.

W10S Bit (Worst 10 Acquisition Control Select)

This bit selects the value used for measurement and filtering of worst 10 gradients. When this bit is set to 0, the values acquired in registers PW10VRU, PW10VRM, and PW10VRL and registers MW10RU, MW10RM, and MW10RL are used as the limit for the filter. When the bit is set to 1, the values set in registers PLIMITRU, PLIMITRM, and PLIMITRL and registers MLIMITRU, MLIMITRM, and MLIMITRL are used as the limit for the filter.

SYTH[3:0] Bits (Synchronized State Detection Threshold Setting)

These bits specify a value for the number of consecutive times being within the thresholds will be judged as synchronization. Specifically, the value falling within the thresholds set in registers SYNTDBRU and SYNTDBRL as many times as the value of these bits indicates the synchronized state. When the ALEN0 bit is 1, the STSR.SYNCOUT flag becomes 1.

DVTH[3:0] Bits (Synchronization Loss Detection Threshold Setting)

These bits specify a value for the number of consecutive times exceeding a threshold will be judged as a loss of synchronization. Specifically, synchronization is judged to have been lost when the value in registers SYNTDARU and SYNTDARL is consecutively exceeded as many times as the value of these bits. When the ALEN0 bit is 1, the STSR.SYNCOUT flag becomes 1.

ALEN0 Bit (Alarm Detection Enable 0)

When this bit is 1, the STSR.SYNC or SYNCOUT flag is set to 1 on detection of synchronization or loss of synchronization. When this bit is 0, the SYNC or SYNCOUT flag is not set to 1 even if synchronization or loss of synchronization is detected.

ALEN1 Bit (Alarm Detection Enable 1)

When this bit is 1, the STSR.SYNTOUT flag is set to 1 if a Sync message is not received within the time specified by the SYNTOR register. When this bit is 0, the SYNTOUT flag is not set to 1 even if the reception timeout has occurred.

36.2.9 Sync Message Reception Timeout Register (SYNTOR)

Address(es): EPTPC.SYNTOR 000C 4058h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	A Sync message not being received within 1024 × n (ns), where n is the setting, leads to a timeout for reception of Sync messages, leading to the STSR.SYNTOUT flag being set to 1.	R/W

The SYNTOR register is used to specify the timeout period for reception of Sync messages. The timeout period is 1024 times the setting of these bits, in units of nanoseconds.

A Sync message not being received within the period specified by these bits is judged to represent a timeout.

When the SYNTOR register is 0, the STSR.SYNTOUT flag does not become 1.

36.2.10 IPLS Interrupt Request Timer Select Register (IPTSELR)

Address(es): EPTPC.IPTSELR 000C 4060h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	IPTSEL5	IPTSEL4	IPTSEL3	IPTSEL2	IPTSEL1	IPTSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IPTSEL0	Pulse Output Timer 0 Select	0: Pulse output timer 0 is not selected as a source of IPLS interrupt requests. 1: Pulse output timer 0 is selected as a source of IPLS interrupt requests.	R/W
b1	IPTSEL1	Pulse Output Timer 1 Select	0: Pulse output timer 1 is not selected as a source of IPLS interrupt requests. 1: Pulse output timer 1 is selected as a source of IPLS interrupt requests.	R/W
b2	IPTSEL2	Pulse Output Timer 2 Select	0: Pulse output timer 2 is not selected as a source of IPLS interrupt requests. 1: Pulse output timer 2 is selected as a source of IPLS interrupt requests.	R/W
b3	IPTSEL3	Pulse Output Timer 3 Select	0: Pulse output timer 3 is not selected as a source of IPLS interrupt requests. 1: Pulse output timer 3 is selected as a source of IPLS interrupt requests.	R/W
b4	IPTSEL4	Pulse Output Timer 4 Select	0: Pulse output timer 4 is not selected as a source of IPLS interrupt requests. 1: Pulse output timer 4 is selected as a source of IPLS interrupt requests.	R/W
b5	IPTSEL5	Pulse Output Timer 5 Select	0: Pulse output timer 5 is not selected as a source of IPLS interrupt requests. 1: Pulse output timer 5 is selected as a source of IPLS interrupt requests.	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

The IPTSELR register is used to select the pulse output timer that generates IPLS interrupt requests.
Each pulse output timer m (m = 0 to 5) takes the clock signal from the STCA as its clock source and produces pulses with a specified period and duty cycle. An IPLS interrupt is requested on rising edges if the ELIPPR.PLSP bit is set to 1 and on falling edges if the PLSN bit in the same register is set to 1. The interrupt request signal becomes the logical OR of the interrupt requests from the given channels when bits for multiple channels in register IPTSELR are set to 1. For the IPLS interrupt, see section 36.4, Interrupts.

36.2.11 MINT Interrupt Request Timer Select Register (MITSELR)

Address(es): EPTPC.MITSELR 000C 4064h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	MINTEN5	MINTEN4	MINTEN3	MINTEN2	MINTEN1	MINTEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MINTEN0	Pulse Output Timer 0 MINT Interrupt Output Enable	0: Output of rising edges by pulse output timer 0 is not reflected by the MIESR.CYC0 flag as a MINT interrupt source. 1: Output of rising edges by pulse output timer 0 is reflected by the MIESR.CYC0 flag as a MINT interrupt source.	R/W
b1	MINTEN1	Pulse Output Timer 1 MINT Interrupt Output Enable	0: Output of rising edges by pulse output timer 1 is not reflected by the MIESR.CYC1 flag as a MINT interrupt source. 1: Output of rising edges by pulse output timer 1 is reflected by the MIESR.CYC1 flag as a MINT interrupt source.	R/W
b2	MINTEN2	Pulse Output Timer 2 MINT Interrupt Output Enable	0: Output of rising edges by pulse output timer 2 is not reflected by the MIESR.CYC2 flag as a MINT interrupt source. 1: Output of rising edges by pulse output timer 2 is reflected by the MIESR.CYC2 flag as a MINT interrupt source.	R/W
b3	MINTEN3	Pulse Output Timer 3 MINT Interrupt Output Enable	0: Output of rising edges by pulse output timer 3 is not reflected by the MIESR.CYC3 flag as a MINT interrupt source. 1: Output of rising edges by pulse output timer 3 is reflected by the MIESR.CYC3 flag as a MINT interrupt source.	R/W
b4	MINTEN4	Pulse Output Timer 4 MINT Interrupt Output Enable	0: Output of rising edges by pulse output timer 4 is not reflected by the MIESR.CYC4 flag as a MINT interrupt source. 1: Output of rising edges by pulse output timer 4 is reflected by the MIESR.CYC4 flag as a MINT interrupt source.	R/W
b5	MINTEN5	Pulse Output Timer 5 MINT Interrupt Output Enable	0: Output of rising edges by pulse output timer 5 is not reflected by the MIESR.CYC5 flag as a MINT interrupt source. 1: Output of rising edges by pulse output timer 5 is reflected by the MIESR.CYC5 flag as a MINT interrupt source.	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

The MITSELR register is used to select pulse output timer m that generates MINT interrupt requests.
Each pulse output timer m (m = 0 to 5) takes the clock signal from the STCA as its clock source and produces pulses with a specified period and duty cycle. A MINT interrupt is requested on rising edges of the pulse signal from the corresponding pulse output timer m if the setting of the MIEIPR.CYCm bit is 1.
For the MINT interrupt, see section 36.4, Interrupts.

36.2.12 ELC Output Timer Select Register (ELTSELR)

Address(es): EPTPC.ELTSELR 000C 4068h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	ELTDIS5	ELTDIS4	ELTDIS3	ELTDIS2	ELTDIS1	ELTDIS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ELTDIS0	Pulse Output Timer 0 Event Generation Disable	0: Pulse output timer 0 is used for the generation of event signals for the ELC. 1: Pulse output timer 0 is not used for the generation of event signals for the ELC.	R/W
b1	ELTDIS1	Pulse Output Timer 1 Event Generation Disable	0: Pulse output timer 1 is used for the generation of event signals for the ELC. 1: Pulse output timer 1 is not used for the generation of event signals for the ELC.	R/W
b2	ELTDIS2	Pulse Output Timer 2 Event Generation Disable	0: Pulse output timer 2 is used for the generation of event signals for the ELC. 1: Pulse output timer 2 is not used for the generation of event signals for the ELC.	R/W
b3	ELTDIS3	Pulse Output Timer 3 Event Generation Disable	0: Pulse output timer 3 is used for the generation of event signals for the ELC. 1: Pulse output timer 3 is not used for the generation of event signals for the ELC.	R/W
b4	ELTDIS4	Pulse Output Timer 4 Event Generation Disable	0: Pulse output timer 4 is used for the generation of event signals for the ELC. 1: Pulse output timer 4 is not used for the generation of event signals for the ELC.	R/W
b5	ELTDIS5	Pulse Output Timer 5 Event Generation Disable	0: Pulse output timer 5 is used for the generation of event signals for the ELC. 1: Pulse output timer 5 is not used for the generation of event signals for the ELC.	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

The ELTSELR register is used to select pulse output timer m that outputs event signals to the ELC.
Each pulse output timer m (m = 0 to 5) takes the clock signal from the STCA as its clock source and produces pulses with a specified period and duty cycle. An event signal is output to the ELC on rising edges if the ELIPPR.CYCPm bit is set to 1 and on falling edges if the CYCNm bit in the same register is set to 1.
For the output of event signals to the ELC, see section 36.4, Interrupts.

36.2.13 Time Synchronization Channel Select Register (STCHSELR)

Address(es): EPTPC.STCHSELR 000C 406Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SYSEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SYSEL	Timer Information Input Select	0: Time information from the SYNFP0 module is used. 1: Time information from the SYNFP1 module is used.	R/W*1
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Do not change the value of this bit while the SYNSTARTR.STR bit is 1.

The STCHSELR register selects the time information input to the STCA module.

36.2.14 Slave Time Synchronization Start Register (SYNSTARTR)

Address(es): EPTPC.SYNSTARTR 000C 4080h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STR	Slave Time Synchronization Control	0: Slave time synchronization is stopped. 1: Slave time synchronization is started.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SYNSTARTR register is used to start or stop time synchronization. This register is used when the EPTPC is operating as a slave node.

36.2.15 Local Clock Counter Initial Value Load Directive Register (LCIVLDR)

Address(es): EPTPC.LCIVLDR 000C 4084h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LOAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	LOAD	Local Clock Counter Initial Value Load Directive	0: The initial value is not loaded into the local clock counter. 1: The initial value is loaded into the local clock counter.	W*1
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Do not change the value of this bit while the SYNSTARTR.STR bit is 1.

The LCIVLDR register is used to set the value of registers LCIVRU, LCIVRM, and LCIVRL as the initial value of the local clock counter.

36.2.16 Synchronization Loss Detection Threshold Registers (SYNTDARU, SYNTDARL)

Address(es): EPTPC.SYNTDARU 000C 4090h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the higher-order 32 bits of the threshold for detection of loss of synchronization.	R/W

Address(es): EPTPC.SYNTDARL 000C 4094h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the lower-order 32 bits of the threshold for detection of loss of synchronization.	R/W

The settings of the SYNTDARU and SYNTDARL registers specify the threshold value for offsetFromMaster to be used in judging loss of synchronization. When setting a threshold value, write the higher-order 32 bits to SYNTDARU and the lower-order 32 bits to SYNTDARL, in that order and in consecutive operations.

If the offsetFromMaster value exceeds the value specified in SYNTDARU and SYNTDARL, a loss of synchronization is detected. Set a value in SYNTDARU and SYNTDARL in units of nanoseconds.

SYNTDARU and SYNTDARL are not used when the device is operating as a master clock.

36.2.17 Synchronization Detection Threshold Registers (SYNTDBRU, SYNTDBRL)

Address(es): EPTPC.SYNTDBRU 000C 4098h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the higher-order 32 bits of the threshold for detection of synchronization.	R/W

Address(es): EPTPC.SYNTDBRL 000C 409Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the lower-order 32 bits of the threshold for detection of synchronization.	R/W

The settings of the SYNTDBRU and SYNTDBRL registers specify the threshold value for offsetFromMaster to be used in judging synchronization. When setting a threshold value, write the higher-order 32 bits to SYNTDBRU and the lower-order 32 bits to SYNTDBRL, in that order and in consecutive operations.
If the offsetFromMaster value is less than the value specified in SYNTDBRU and SYNTDBRL, synchronization is detected. Set a value in SYNTDBRU and SYNTDBRL in units of nanoseconds.
SYNTDBRU and SYNTDBRL are not used when the device is operating as a master clock.

36.2.18 Local Clock Counter Initial Value Registers (LCIVRU, LCIVRM, LCIVRL)

Address(es): EPTPC.LCIVRU 000C 40B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	—	These bits hold the setting for the higher-order 16 bits of the integer portion of the initial value for the local clock counter.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Address(es): EPTPC.LCIVRM 000C 40B4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the lower-order 32 bits of the integer portion of the initial value for the clock counter.	R/W

Address(es): EPTPC.LCIVRL 000C 40B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the fractional portion of the initial value of the clock counter in nanoseconds.	R/W

Registers LCIVRU, LCIVRM, and LCIVRL specify the initial value in seconds of the clock counter. When setting an initial value, write the higher-order 16 bits of the integer portion to LCIVRU, the lower-order 32 bits of the integer portion to LCIVRM, and the fractional portion in nanoseconds to LCIVRL, in that order and in consecutive operations. The value of registers LCIVRU, LCIVRM, and LCIVRL can be used as the initial value of the local clock counter. When setting the value of registers LCIVRU, LCIVRM, and LCIVRL in the local clock counter, set the LCIVLDR.LOAD bit to 1.

Example) When 2.000000025s is to be set as the initial value, write the following values to each of the registers.
LCIVRU: 0000 0000h
LCIVRM: 0000 0002h
LCIVRL: 0000 0019h

36.2.19 Worst 10 Acquisition Directive Register (GETW10R)

Address(es): EPTPC.GETW10R 000C 4124h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GW10
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	GW10	Worst 10 Acquisition Directive	0: The worst-10 values are not acquired. 1: Starts acquisition of the worst-10 values.	R/W*1
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Do not set this bit to 1 while the STMR.W10S bit is 0.

The GETW10R register is used in software control to start calculation of gradient values for use in selecting the worst-10 values.
A gradient value is the amount by which the timer counter of a slave is incremented when a given interval elapses. Setting the GW10 bit to 1 while the value of the STMR.W10S bit is 1 selects calculation of a gradient value by the EPTPC each time it receives a Sync message. Gradient values are calculated the number of times specified by the STMR.WINT[7:0] bits. The GW10 bit is cleared to 0 on completion of this number of calculations. The GETW10R register is not used in operation as a master clock.

36.2.20 Positive Gradient Limit Registers (PLIMITRU, PLIMITRM, PLIMITRL)

Address(es): EPTPC.PLIMITRU 000C 4128h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b30 to b0	—	—	These bits hold the setting for the higher-order 31 bits of the limit for the positive gradient.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R

Address(es): EPTPC.PLIMITRM 000C 412Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the middle-order 32 bits of the limit for the positive gradient.	R/W

Address(es): EPTPC.PLIMITRL 000C 4130h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the lower-order 32 bits of the limit for the positive gradient.	R/W

Registers PLIMITRU, PLIMITRM, and PLIMITRL are used to specify an upper limit on gradient (= positive gradient) for use in time synchronization. When setting an upper limit, write values consecutively to PLIMITRU, PLIMITRM, and PLIMITRL in that order.
The gradients that exceed the value specified in registers PLIMITRU, PLIMITRM, and PLIMITRL are not used in time synchronization. Registers PLIMITRU, PLIMITRM, PLIMITRL are not used when the device is operating as a master clock. Registers PLIMITRU, PLIMITRM, and PLIMITRL are effective while the STMR.CMOD and W10S bits are 1.

The gradient value to be set in the register is calculated by using the following expression.

PLIMITRU, PLIMITRM, and PLIMITRL register values = A (s)/T (s) × 2^{32}

A: The time (s) by which the slave local clock counter advances during the interval between received Sync messages

T: The actual time (s) between received Sync messages

For example, if the interval between Sync messages is 0.5 seconds and the local clock counter advances by 0.7 seconds during that time, and this is to be set as the limit, then the setting for PLIMITR = 0.7/0.5 × 2^{32} = 6 012 954 214 = 1 6666 6666h,
and the settings for the individual registers are as follows: PLIMITRU = 0000 0000h, PLIMITRM = 0000 0001h, and PLIMITRL = 6666 6666h.

The minimum setting depends on the STCA clock frequency as the clock source for counting by the local clock counter. For example, if the STCA clock frequency is 50 MHz, then the minimum allowable setting for registers PLIMITRU, PLIMITRM, and PLIMITRL = (1/50 (MHz)) (s)/0.5 (s) × 2^{32} = 172 = ACh,
and the settings for the individual registers are as follows: PLIMITRU = 0000 0000h, PLIMITRM = 0000 0000h, and PLIMITRL = 0000 00ACh.
The gradient limit values to be set are valid when time synchronization correction mode is mode 2 (the STMR.CMOD bit is 1) and the gradient is controlled by software (the STMR.W10S bit is 1).

36.2.21 Negative Gradient Limit Registers (MLIMITRU, MLIMITRM, MLIMITRL)

Address(es): EPTPC.MLIMITRU 000C 4134h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b30 to b0	—	—	These bits hold the setting for the higher-order 31 bits of the limit for the negative gradient.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R

Address(es): EPTPC.MLIMITRM 000C 4138h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the middle-order 32 bits of the limit for the negative gradient.	R/W

Address(es): EPTPC.MLIMITRL 000C 413Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the lower-order 32 bits of the limit for the negative gradient.	R/W

Registers MLIMITRU, MLIMITRM, and MLIMITRL are used to specify a lower limit for the gradient (= negative gradient) for use in time synchronization. Use two's complement to set a lower limit. When setting a lower limit, write values consecutively to MLIMITRU, MLIMITRM, and MLIMITRL in that order.
The gradients that are less than the value specified in registers MLIMITRU, MLIMITRM, and MLIMITRL are not used in time synchronization. Registers MLIMITRU, MLIMITRM, and MLIMITRL are not used when the device is operating as a master clock. Registers MLIMITRU, MLIMITRM, and MLIMITRL are effective while the STMR.CMOD and W10S bits are 1.
The procedure for setting a value and the minimum value that can be set are the same as for registers PLIMITRU, PLIMITRM, and PLIMITRL.

36.2.22 Statistical Information Retention Control Register (GETINFOR)

Address(es): EPTPC.GETINFOR 000C 4140h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INFO
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	INFO	Information Retention Control	[When written] 0: Has no effects. 1: Information is retained. [When read] 0: Information retention is completed. 1: Processing for information retention is in progress. Once information fetching is directed, values of various statistical information read before completion of information fetching are not guaranteed.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

The GETINFOR register controls retention of the following statistical information.

- Registers LCCVRU, LCCVRM, and LCCVRL
- Registers PW10VRU, PW10VRM, and PW10VRL
- Registers MW10RU, MW10RM, and MW10RL

The only value that is writable to the INFO bit is 1. When setting a value in registers PW10VRU, PW10VRM, and PW10VRL, or registers MW10RU, MW10RM, and MW10RL, set the INFO bit to 1 only while the STMR.W10S bit is 1. If the INFO bit is set to 1 before acquisition of the worst-10 values is completed, whether the information retained in registers PW10VRU, PW10VRM, and PW10VRL and registers MW10RU, MW10RM, MW10RL is correct or not is not guaranteed. Use the GETW10R.GW10 bit to confirm that acquisition has been completed before setting the INFO bit to 1. The INFO bit is automatically returned to 0 on completion of information fetching.

36.2.23 Local Clock Counters (LCCVRU, LCCVRM, LCCVRL)

Address(es): EPTPC.LCCVRU 000C 4170h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	—	These bits are for reading the higher-order 16 bits of the integer portion of the local clock counter's value.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

Address(es): EPTPC.LCCVRM 000C 4174h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits are for reading the lower-order 32 bits of the integer portion of the local clock counter's value.	R

Address(es): EPTPC.LCCVRL 000C 4178h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits are for reading the fractional portion of the local clock counter's value (in nanoseconds).	R

Registers LCCVRU, LCCVRM, and LCCVRL indicate the local clock counter's value.
When the GETINFOR.INFO bit is set to 1, the value of the local clock counter at that time is stored in registers LCCVRU, LCCVRM, and LCCVRL. The higher-order 16 bits of the integer portion in seconds are saved in LCCVRU, the lower-order 32 bits of the integer portion in seconds are saved in LCCVRM, and the fractional portion in nanoseconds is saved in LCCVRL.
For example, if the local time information is 14:25, 44 seconds, 10 milliseconds, 23 microseconds, and 39 nanoseconds, registers LCCVRU, LCCVRM, and LCCVRL have $14 \times 3600 + 25 \times 60 + 44 = 51944$ (s) = 0000 0000 CAE8h as the setting of the higher-order 48 bits and $10 \times 10^6 + 23 \times 10^3 + 39 = 10023039$ (ns) = 0098 F07Fh as the setting of the lower-order 32 bits.

36.2.24 Positive Gradient Worst 10 Value Registers (PW10VRU, PW10VRM, PW10VRL)

Address(es): EPTPC.PW10VRU 000C 4210h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits are for reading the higher-order 32 bits of the positive gradient value.	R

Address(es): EPTPC.PW10VRM 000C 4214h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits are for reading the middle-order 32 bits of the positive gradient value.	R

Address(es): EPTPC.PW10VRL 000C 4218h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits are for reading the lower-order 32 bits of the positive gradient value.	R

Registers PW10VRU, PW10VRM, and PW10VRL indicate the worst 10 of the positive gradient values.
When the GETINFOR.INFO bit is set to 1, the worst 10 value at that time is stored in registers PW10VRU, PW10VRM, and PW10VRL.

The format of the worst 10 gradients stored in registers PW10VRU, PW10VRM, and PW10VRL are the same as for registers PLIMITRU, PLIMITRM, and PLIMITRL. For details on the format of the values, see the description of the PLIMITR registers.
Registers PW10VRU, PW10VRM, and PW10VRL are not used when the device is used as a master clock.

36.2.25 Negative Gradient Worst 10 Value Registers (MW10RU, MW10RM, MW10RL)

Address(es): EPTPC.MW10RU 000C 42D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits are for reading the higher-order 32 bits of the negative gradient value.	R

Address(es): EPTPC.MW10RM 000C 42D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits are for reading the middle-order 32 bits of the negative gradient value.	R

Address(es): EPTPC.MW10RL 000C 42D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits are for reading the lower-order 32 bits of the negative gradient value.	R

Registers MW10RU, MW10RM, and MW10RL indicate the worst 10 of the negative gradient values.
When the GETINFOR.INFO bit is set to 1, the worst 10 value at that time is stored in registers MW10RU, MW10RM, and MW10RL.
The format of the worst 10 gradients stored in registers MW10RU, MW10RM, and MW10RL are the same as for registers MLIMITRU, MLIMITRM, and MLIMITRL. For details on the format of the values, see the description of the MLIMITR registers.
Registers MW10RU, MW10RM, and MW10RL are not used when the device is used as a master clock.

36.2.26 Timer Start Time Setting Registers (TMSTTRUm, TMSTTRLm) (m = 0 to 5)

Address(es): EPTPC.TMSTTRU0 000C 4300h, EPTPC.TMSTTRU1 000C 4310h, EPTPC.TMSTTRU2 000C 4320h, EPTPC.TMSTTRU3 000C 4330h, EPTPC.TMSTTRU4 000C 4340h, EPTPC.TMSTTRU5 000C 4350h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the higher-order 32 bits of the start time of the pulse output timer in nanoseconds.	R/W

Address(es): EPTPC.TMSTTRL0 000C 4304h, EPTPC.TMSTTRL1 000C 4314h, EPTPC.TMSTTRL2 000C 4324h, EPTPC.TMSTTRL3 000C 4334h, EPTPC.TMSTTRL4 000C 4344h, EPTPC.TMSTTRL5 000C 4354h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the lower-order 32 bits of the start time of the pulse output timer in nanoseconds.	R/W

Registers TMSTTRUm and TMSTTRLm are used to set the start time of pulse output timer m.

Set the start time of pulse output timer m (64 bits) in units of nanoseconds. Though the setting is in units of nanoseconds, the start time of pulse output timer m depends on the resolution of the STCA clock.

For example, if the STCA clock is running at 50 MHz, one cycle takes 20 ns, so the time at which the timer starts may differ from the time set in registers TMSTTRUm and TMSTTRLm by up to 20 ns.

When writing to registers TMSTTRUm and TMSTTRLm, write values consecutively in order of TMSTTRUm and then TMSTTRLm while the TMSTARTR.ENm bit is 0.

Note, however, that the format for setting times in registers TMSTTRUm and TMSTTRLm differs from that described in section 36.2.23, Local Clock Counters (LCCVRU, LCCVRM, LCCVRL).

36.2.27 Timer Cycle Setting Registers m (TMCYCRm) (m = 0 to 5)

Address(es): EPTPC.TMCYCR0 000C 4308h, EPTPC.TMCYCR1 000C 4318h, EPTPC.TMCYCR2 000C 4328h, EPTPC.TMCYCR3 000C 4338h, EPTPC.TMCYCR4 000C 4348h, EPTPC.TMCYCR5 000C 4358h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b29 to b0	—	—	These bits set the cycle of the pulse output timer in nanoseconds. Set a value that is equivalent to at least four cycles of the STCA clock.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

A TMCYCRm register specifies the period of the output signal generated by the corresponding pulse output timer m. Set a value in nanoseconds that is equivalent to at least four cycles of the STCA clock while the value of the TMSTARTR.ENm bit is 0.

Though the settings of TMCYCRm registers are in units of nanoseconds, the period of the output signal generated by pulse output timer m and the time at which the timer starts depend on the period of the STCA clock. For example, one cycle of the STCA clock running at 50 MHz takes 20 ns, so the clock source for counting by pulse output timer m may differ from the period set in the TMCYCRm registers by up to 19 ns. The SYNFP module handles calculations to correct this difference.

For example, if the setting for the timer period is 81 ns and the STCA clock is running at 50 MHz, the only available settings close to the actual timer period are for 80 ns or 100 ns. By setting the timer period in the SYNFP module to 80 ns for 19 and to 100 ns for 1 of every 20 cycles, we are able to adjust the average period to 81 ns.

(80 (ns) × 19 + 100 (ns) × 1)/20 = 81 (ns)

The minimum value that can be set in a TMCYCRm register is four cycles of the STCA clock. For example, if the STCA clock is running at 50 MHz, the minimum setting corresponds to 80 ns. Timer operation is not guaranteed if a value set in one of these registers is less than this value.

36.2.28 Timer Pulse Width Setting Register m (TMPLSRm) (m = 0 to 5)

Address(es): EPTPC.TMPLSR0 000C 430Ch, EPTPC.TMPLSR1 000C 431Ch, EPTPC.TMPLSR2 000C 432Ch, EPTPC.TMPLSR3 000C 433Ch, EPTPC.TMPLSR4 000C 434Ch, EPTPC.TMPLSR5 000C 435Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b28 to b0	—	—	These bits set the width at high level of the pulse signal from the timer in nanoseconds. Set a value that is equivalent to at least two cycles of the STCA clock.	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R

A TMPLSRm register specifies the width at high level of the output signal generated by the corresponding pulse output timer m.

When the TMSTARTR.ENm bit is 0, set a value corresponding to a time no shorter than two cycles of the STCA clock in units of nanoseconds.

The higher-order 3 bits of the TMPLSRm register are reserved. These bits are read as 000b. When writing, write 000b to these bits.

Though the settings of the TMPLSRm register are in units of nanoseconds, the width at high level of the signal from the timer depends on the period of the STCA clock. The method for correcting the width at high level of the signal from the timer is the same as that for correcting the timer periods set by the TMCYCRm register.

36.2.29 Timer Start Register (TMSTARTR)

Address(es): EPTPC.TMSTARTR 000C 437Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	EN5	EN4	EN3	EN2	EN1	EN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EN0	Pulse Output Timer 0 Start	0: Stops pulse output timer 0. 1: Starts pulse output timer 0.	R/W
b1	EN1	Pulse Output Timer 1 Start	0: Stops pulse output timer 1. 1: Starts pulse output timer 1.	R/W
b2	EN2	Pulse Output Timer 2 Start	0: Stops pulse output timer 2. 1: Starts pulse output timer 2.	R/W
b3	EN3	Pulse Output Timer 3 Start	0: Stops pulse output timer 3. 1: Starts pulse output timer 3.	R/W
b4	EN4	Pulse Output Timer 4 Start	0: Stops pulse output timer 4. 1: Starts pulse output timer 4.	R/W
b5	EN5	Pulse Output Timer 5 Start	0: Stops pulse output timer 5. 1: Starts pulse output timer 5.	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

The TMSTARTR register is used to start and stop pulse output timer m.

36.2.30 PRC-TC Status Register (PRSR)

Address(es): EPTPC.PRSR 000C 4400h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	URE1	URE0	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MACE	—	—	—	—	OVRE3	OVRE2	OVRE1	OVRE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OVRE0	Relay Packet Overflow Detection Flag 0	0: No overflow in transfer of data from SYNFP1 to PTPEDMAC 1: An overflow has been detected in transfer of data from SYNFP1 to PTPEDMAC.	R/W*1
b1	OVRE1	Relay Packet Overflow Detection Flag 1	0: No overflow in transfer of data from SYNFP0 to PTPEDMAC 1: An overflow has been detected in transfer of data from SYNFP0 to PTPEDMAC.	R/W*1
b2	OVRE2	Relay Packet Overflow Detection Flag 2	0: No overflow in transfer of data from SYNFP1 to SYNFP0 1: An overflow has been detected in transfer of data from SYNFP1 to SYNFP0.	R/W*1
b3	OVRE3	Relay Packet Overflow Detection Flag 3	0: No overflow in transfer of data from SYNFP0 to SYNFP1 1: An overflow has been detected in transfer of data from SYNFP0 to SYNFP1.	R/W*1
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	MACE	Source MAC Address Mismatch Detection Flag	0: A MAC address mismatch has not been detected. 1: A MAC address mismatch has been detected.	R/W*1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b27 to b16	—	Reserved	These bits are read as undefined. The write value should be 0.	R/W
b28	URE0	Relay Packet Underflow Detection Flag 0	0: No underflow in transfer of data from SYNFP1 to SYNFP0 1: An underflow has been detected in transfer of data from SYNFP1 to SYNFP0.	R/W*1
b29	URE1	Relay Packet Underflow Detection Flag 1	0: No underflow in transfer of data from SYNFP0 to SYNFP1 1: An underflow has been detected in transfer of data from SYNFP0 to SYNFP1.	R/W*1
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writing 1 clears the flag. Writing 0 does not affect the flag value.

The PRSR register indicates the state of the PRC-TC module.
The PRC-TC module transmits and receives Ethernet packets through Ethernet port 0, Ethernet port 1, and the PTPEDMAC.
The PRSR register indicates the states as overflows, underflows, and non-matching MAC addresses in the transmission and reception of Ethernet packets.

OVRE0 Flag (Relay Packet Overflow Detection Flag 0)

This flag indicates that the FIFO buffer has overflowed while packets were being transferred from the SYNFP1 module to the PTPEDMAC. The data received in the PTPEDMAC may be incorrect if the setting of the OVRE0 flag is 1.

OVRE1 Flag (Relay Packet Overflow Detection Flag 1)

This flag indicates that the FIFO buffer has overflowed while packets were being transferred from the SYNFP0 module to the PTPEDMAC. The data received in the PTPEDMAC may be incorrect if the setting of the OVRE1 flag is 1.

OVRE2 Flag (Relay Packet Overflow Detection Flag 2)

This flag indicates that the FIFO buffer has overflowed while packets were being transferred from the SYNFP1 module to the SYNFP0 module.

OVRE3 Flag (Relay Packet Overflow Detection Flag 3)

This flag indicates that the FIFO buffer has overflowed while packets were being transferred from the SYNFP0 module to the SYNFP1 module.

MACE Flag (Source MAC Address Mismatch Detection Flag)

This flag becomes 1 if the source MAC address of a packet for transmission from the PTPEDMAC matches none of registers PRMACRU0, PRMACRL0, PRMACRU1, and PRMACRL1.

URE0 Flag (Relay Packet Underflow Detection Flag 0)

This flag indicates that the FIFO buffer has underflowed while packets were being transferred from the SYNFP1 module to the SYNFP0 module.

URE1 Flag (Relay Packet Underflow Detection Flag 1)

This flag indicates that the FIFO buffer has underflowed while packets were being transferred from the SYNFP0 module to the SYNFP1 module.

36.2.31 PRC-TC Status Notification Enable Register (PRIPR)

Address(es): EPTPC.PRIPR 000C 4404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	URE1	URE0	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MACE	—	—	—	—	OVRE3	OVRE2	OVRE1	OVRE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OVRE0	PRSR.OVRE0 Status Notification Enable	0: Disables notification of the state of PRSR.OVRE0. 1: Enables notification of the state of PRSR.OVRE0.	R/W
b1	OVRE1	PRSR.OVRE1 Status Notification Enable	0: Disables notification of the state of PRSR.OVRE1. 1: Enables notification of the state of PRSR.OVRE1.	R/W
b2	OVRE2	PRSR.OVRE2 Status Notification Enable	0: Disables notification of the state of PRSR.OVRE2. 1: Enables notification of the state of PRSR.OVRE2.	R/W
b3	OVRE3	PRSR.OVRE3 Status Notification Enable	0: Disables notification of the state of PRSR.OVRE3. 1: Enables notification of the state of PRSR.OVRE3.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	MACE	PRSR.MACE Status Notification Enable	0: Disables notification of the state of PRSR.MACE 1: Enables notification of the state of PRSR.MACE	R/W
b27 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b28	URE0	PRSR.URE0 Status Notification Enable	0: Disables notification of the state of PRSR.URE0. 1: Enables notification of the state of PRSR.URE0.	R/W
b29	URE1	PRSR.URE1 Status Notification Enable	0: Disables notification of the state of PRSR.URE1. 1: Enables notification of the state of PRSR.URE1.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: Refer to section 36.4, Interrupts, for details on interrupt system.

The PRIPR register specifies whether the MIESR.PRC flag does or does not reflect changes in the state of the PRC-TC module.

36.2.32 Channel 0 Local MAC Address Registers (PRMACRU0, PRMACRL0)

Address(es): EPTPC.PRMACRU0 000C 4410h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	—	These bits hold the setting for the higher-order 24 bits of the local MAC address for Ethernet port 0.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

Address(es): EPTPC.PRMACRL0 000C 4414h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	—	These bits hold the setting for the lower-order 24 bits of the local MAC address for Ethernet port 0.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

Registers PRMACRU0 and PRMACRL0 are used to set the local MAC address for Ethernet port 0. Set the higher-order 24 bits and the lower-order 24 bits of the MAC address in PRMACRU0 and PRMACRL0, respectively.
These registers are used in transmission from the PTPEDMAC to send frames from transmission sources with MAC addresses matching the value of this register to Ethernet port 0. Set registers PRMACRU0 and PRMACRL0 to the same value as registers SYMACRU and SYMACRL for the EPTPC0 module.
Rewrite registers PRMACRU0 and PRMACRL0 before starting the EDMAC, ETHERC, and PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.33 Channel 1 Local MAC Address Registers (PRMACRU1, PRMACRL1)

Address(es): EPTPC.PRMACRU1 000C 4418h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	—	These bits hold the setting for the higher-order 24 bits of the local MAC address for Ethernet port 1.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

Address(es): EPTPC.PRMACRL1 000C 441Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	—	These bits hold the setting for the lower-order 24 bits of the local MAC address for Ethernet port 1.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

Registers PRMACRU1 and PRMACRL1 are used to set the local MAC address for Ethernet port 1. Set the higher-order 24 bits and the lower-order 24 bits of the MAC address in registers PRMACRU1 and PRMACRL1, respectively.
These registers are used in transmission from the PTPEDMAC to send frames from transmission sources with MAC addresses matching the value of this register to Ethernet port 1. Set registers PRMACRU1 and PRMACRL1 to the same value as registers SYMACRU and SYMACRL for the EPTPC1 module.
Rewrite registers PRMACRU1 and PRMACRL1 before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.34 Packet Transmission Control Register (TRNDISR)

Address(es): EPTPC.TRNDISR 000C 4420h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TDIS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TDIS[1:0]	Packet Transmission Control	b1 b0 0 0: PTP packets are transmitted through both Ethernet port 0 and Ethernet port 1. 0 1: PTP packets are only transmitted through Ethernet port 0. 1 0: PTP packets are only transmitted through Ethernet port 1. 1 1: Setting prohibited	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

The TRNDISR register controls transmission of PTP packets when Ethernet port 0 and Ethernet port 1 have the same local MAC address.

When registers PRMACRU0 and PRMACRL0 and registers PRMACRU1 and PRMACRL1 have the same local MAC address setting, the EPTPC can selects whether PTP packets with the matching address are transmitted through both ports or are only transmitted through one port or the other.

If Ethernet port 0 and Ethernet port 1 have different local MAC address settings, set the TDIS[1:0] bits to 00b. If the setting is other than 00b, the transmission of frames with an source MAC address matching that of the Ethernet port may be blocked.

Rewrite the TRNDISR register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.35 Relay Mode Register (TRNMR)

Address(es): EPTPC.TRNMR 000C 4430h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	FWD1	FWD0	—	—	—	—	—	—	—	MOD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MOD	Cut-Through Mode	0: Store-and-forward 1: Cut-through	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	FWD0	Channel 0 Relay Enable	0: Unicast, multicast (other than PTP packets), and broadcast messages from the other node are not relayed from port 0 to port 1. 1: Unicast, multicast (other than PTP packets), and broadcast messages from the other node are relayed from port 0 to port 1.	R/W
b9	FWD1	Channel 1 Relay Enable	0: Unicast, multicast (other than PTP packets), and broadcast messages from the other node are not relayed from port 1 to port 0. 1: Unicast, multicast (other than PTP packets), and broadcast messages from the other node are relayed from port 1 to port 0.	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R

The TRNMR register is used to control relaying of packets between Ethernet port 0 and Ethernet port 1 by the PRC-TC module. For example, as Figure 36.3 shows, if the network is in a daisy-chain configuration, packets can be transferred through all three devices by enabling relaying of packets between Ethernet port 0 and Ethernet port 1 in the center of the figure.

For recalculating the CRC value in packet transfer, in the case of cut-through mode, a packet that had an abnormal CRC value may be transferred with a normal CRC value. Use store-and-forward mode when abnormal packets need to be discarded. There is no difference in the latency of transfer in cut-through mode and store-and-forward mode when the length of the packets is shorter than 96 bytes.

Rewrite the TRNMR register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

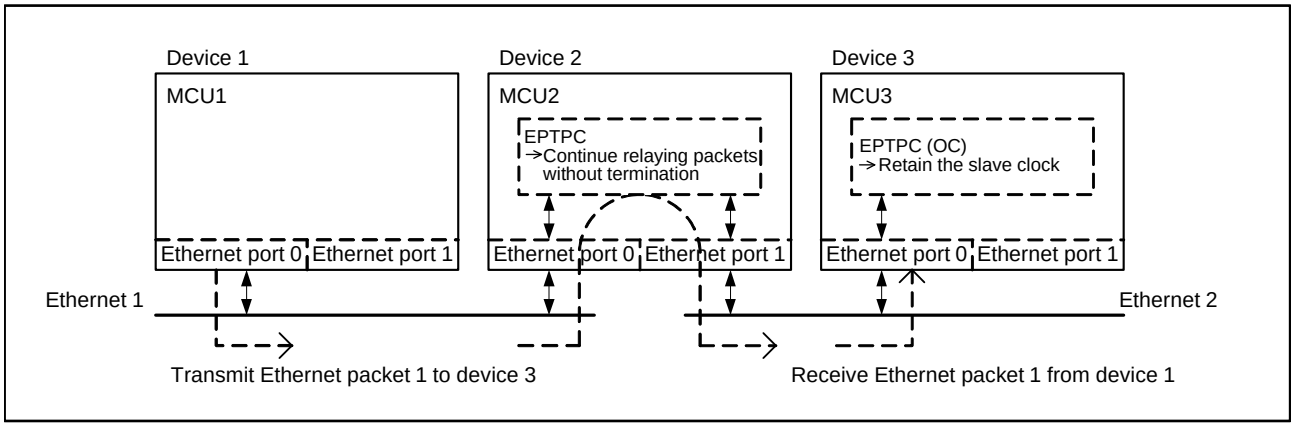


Figure 36.3 Network Configuration Example

36.2.36 Cut-Through Transfer Start Threshold Register (TRNCTTDR)

Address(es): EPTPC.TRNCTTDR 000C 4434h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	THVAL[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	THVAL[10:0]	FIFO Read Start Threshold	Threshold for starting to read data from the relay FIFO in cut-through mode (specified as the number of bytes)*1, *2	R
b10 to b2				R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. A value cannot be set in the lower-order 2 bits. These bits are fixed to 0.
Note 2. A value of less than 96 bytes cannot be set.

The TRNCTTDR register is used to set the threshold for the transmitting port starting to read data from the relay FIFO when the cut-through method is selected for relaying between the Ethernet ports.
When the TRNMR.MOD bit is 1, data transfer can start without waiting for all frame data to be stored in the relay FIFO. The setting of the THVAL[10:0] bits is the amount of data, in bytes, that must be stored in the relay FIFO before transmission starts. Set a multiple of four as the threshold value.
Reading from the relay FIFO starts when either of the following conditions is met.

- The relay FIFO holds at least as much data as the number of bytes specified by the THVAL[10:0] bits.
- The relay FIFO holds at least one frame.

Rewrite the TRNCTTDR register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.37 SYNFP Status Register (SYSR)

Address(es): EPTPC0.SYSR 000C 4800h, EPTPC1.SYSR 000C 4C00h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GENDN	RESDN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	INFABT	—	RECLP	—	—	—	—	—	DRQOVR	INTDEV	DRPTO	—	MPDUD	INTCHG	OFMUD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OFMUD	offsetFromMaster Value Update Flag	0: The offsetFromMaster value has not been updated. 1: The offsetFromMaster value has been updated.	R/W*1
b1	INTCHG	Receive logMessageInterval Value Change Detection Flag	0: No change in the received logMessageInterval value. 1: A change in the received logMessageInterval value.	R/W*1
b2	MPDUD	meanPathDelay Value Update Flag	0: The meanPathDelay value has not been updated. 1: The meanPathDelay value has been updated.	R/W*1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	DRPTO	Delay_Resp/Pdelay_Resp Reception Timeout Detection Flag	0: A Delay_Resp/Pdelay_Resp timeout has not occurred. 1: A Delay_Resp/Pdelay_Resp timeout has occurred.	R/W*1
b5	INTDEV	Receive logMessageInterval Value Out-of-Range Flag	0: The received logMessageInterval value is within the range. 1: The received logMessageInterval value is out of the range.	R/W*1
b6	DRQOVR	Delay_Req Reception FIFO Overflow Detection Flag	0: The received Delay_Req has not caused the reception FIFO to overflow. 1: The received Delay_Req has caused the reception FIFO to overflow.	R/W*1
b11 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	RECLP	Loop Reception Detection Flag	0: A received message has not returned through a loop. 1: A received message has returned through a loop.	R/W*1
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	INFABT	Control Information Abnormality Detection Flag	0: No abnormality in control information 1: Abnormality in control information	R/W*1
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R
b16	RESDN	Response Stop Completion Detection Flag	0: Stopping responses has not been completed. 1: Stopping responses has been completed.	R/W*1
b17	GENDN	Generation Stop Completion Detection Flag	0: Stopping generation has not been completed. 1: Stopping generation has been completed.	R/W*1
b23 to b18	—	Reserved	These bits are read as undefined. The write value should be 0.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writing 1 clears the flag. Writing 0 does not affect the flag value.

The SYSR register indicates the state of the SYNFP module.

OFMUD Flag (offsetFromMaster Value Update Flag)

This flag indicates that the value of offsetFromMaster has been updated.

INTCHG Flag (Receive logMessageInterval Value Change Detection Flag)

This flag indicates that the logMessageInterval value of the Delay_Resp, Sync or Announce message differs from the previously received value.

MPDUD Flag (meanPathDelay Value Update Flag)

This flag indicates that the value of meanPathDelay has been updated.

DRPTO Flag (Delay_Resp/Pdelay_Resp Reception Timeout Detection Flag)

This flag indicates that a Delay_Resp or Pdelay_Resp message has not been received within the period set in the RSTOUTR register.

INTDEV Flag (Receive logMessageInterval Value Out-of-Range Flag)

This flag indicates the reception of a Delay_Resp message with a logMessageInterval value outside the range -7 to 6.

DRQOVR Flag (Delay_Req Reception FIFO Overflow Detection Flag)

This flag indicates that the FIFO buffer for storing information from received Delay_Req messages holds 32 or more entries.

RECLP Flag (Loop Reception Detection Flag)

This flag indicates that the value of the sourcePortIdentity field in a received PTP message matches the local PortIdentity (as set by registers SYCIDRU, SYCIDRL, and SYPNUMR).

INFABT Flag (Control Information Abnormality Detection Flag)

This flag indicates that the control information included a mismatch.
When this bit becomes 1, reset the EPTPC, PTPEDMAC, and the corresponding channels ETHERC and EDMAC.
Follow the procedure below to reset these modules:

(1) Write 0000 0001h to the EPTPC.PTRSTR register.
(2) Write 0000 0001h to the PTPEDMAC.EDMR register.
(3) Write 0000 0001h to the EDMACn.EDMR register of the corresponding channels.
(4) Wait for 64 cycles of the PCLKA until initialization is completed.
(5) Write 0000 0000h to the EPTPC.PTRSTR register.

Refer to section 36.2.83, PTP Reset Register (PTRSTR) and section 37.2.1, EDMAC Mode Register (EDMR) for details of resets.

RESDN Flag (Response Stop Completion Detection Flag)

This flag indicates the end of processing for transmission of a Delay_Resp or Pdelay_Resp as response messages when the handling of a received Delay_Req or Pdelay_Req by the SYNFP module has been disabled by the setting of the SYRFL1R or SYRVLDR register.

GENDN Flag (Generation Stop Completion Detection Flag)

This flag indicates the end of processing for transmission of messages of a type disabled in the SYTRENR or SYRVLDR register.

36.2.38 SYNFP Status Notification Enable Register (SYIPR)

Address(es): EPTPC0.SYIPR 000C 4804h, EPTPC1.SYIPR 000C 4C04h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GENDN	RESDN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	INFABT	—	RECLP	—	—	—	—	—	DRQOVR	INTDEV	DRPTO	—	MPDUD	INTCHG	OFMUD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OFMUD	SYSR.OFMUD Status Notification Enable	0: Disables notification of the state of SYSR.OFMUD. 1: Enables notification of the state of SYSR.OFMUD.	R/W
b1	INTCHG	SYSR.INTCHG Status Notification Enable	0: Disables notification of the state of SYSR.INTCHG. 1: Enables notification of the state of SYSR.INTCHG.	R/W
b2	MPDUD	SYSR.MPDUD Status Notification Enable	0: Disables notification of the state of SYSR.MPDUD. 1: Enables notification of the state of SYSR.MPDUD.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	DRPTO	SYSR.DRPTO Status Notification Enable	0: Disables notification of the state of SYSR.DRPTO. 1: Enables notification of the state of SYSR.DRPTO.	R/W
b5	INTDEV	SYSR.INTDEV Status Notification Enable	0: Disables notification of the state of SYSR.INTDEV. 1: Enables notification of the state of SYSR.INTDEV.	R/W
b6	DRQOVR	SYSR.DRQOVR Status Notification Enable	0: Disables notification of the state of SYSR.DRQOVR. 1: Enables notification of the state of SYSR.DRQOVR.	R/W
b11 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	RECLP	SYSR.RECLP Status Notification Enable	0: Disables notification of the state of SYSR.RECLP. 1: Enables notification of the state of SYSR.RECLP.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	INFABT	SYSR.INFABT Status Notification Enable	0: Disables notification of the state of SYSR.INFABT. 1: Enables notification of the state of SYSR.INFABT.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R
b16	RESDN	SYSR.RESDN Status Notification Enable	0: Disables notification of the state of SYSR.RESDN. 1: Enables notification of the state of SYSR.RESDN.	R/W
b17	GENDN	SYSR.GENDN Status Notification Enable	0: Disables notification of the state of SYSR.GENDN. 1: Enables notification of the state of SYSR.GENDN.	R/W
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SYIPR register specifies whether the MIESR.SYn flag does or does not reflect changes in the state of the SYNFPn module.

36.2.39 SYNFP MAC Address Registers (SYMACRU, SYMACRL)

Address(es): EPTPC0.SYMACRU 000C 4810h, EPTPC1.SYMACRU 000C 4C10h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	—	These bits hold the setting for the higher-order 24 bits of the local MAC address.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Address(es): EPTPC0.SYMACRL 000C 4814h, EPTPC1.SYMACRL 000C 4C14h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	—	These bits hold the setting for the lower-order 24 bits of the local MAC address.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Registers SYMACRU and SYMACRL are used to specify the local MAC address for Ethernet ports 0 and 1. Rewrite registers SYMACRU and SYMACRL before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.40 SYNFP Local IP Address Register (SYIPADDRR)

Address(es): EPTPC0.SYIPADDRR 000C 481Ch, EPTPC1.SYIPADDRR 000C 4C1Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the local IP address.	R/W

The SYIPADDRR register is used to specify the local IP address for Ethernet ports 0 and 1.
Rewrite the SYIPADDRR register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.41 SYNFP Specification Version Setting Register (SYSPVRR)

Address(es): EPTPC0.SYSPVRR 000C 4840h, EPTPC1.SYSPVRR 000C 4C40h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TRSP[3:0]				VER[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	VER[3:0]	versionPTP Field Value	These bits are used to set the versionPTP field value of the PTP v2 header. When a message is received, this value is compared with the versionPTP field of the received frame. In generating messages, the value is used for the versionPTP field of the frame for transmission. Set these bits to 0010b (PTP v2).	R/W
b7 to b4	TRSP[3:0]	transportSpecific Field Value	These bits are used to set the transportSpecific field value of the PTP v2 header. When a message is received, this value is compared with the transportSpecific field of the received frame. In generating messages, the value is used for the transportSpecific field of the frame for transmission. Set these bits to 0000b (IEEE 1588).	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SYSPVRR register is used to set the transportSpecific and versionPTP field values of the PTP v2 message header. The value should not be modified while reception or transmission of PTP messages is enabled.

36.2.42 SYNFP Domain Number Setting Register (SYDOMR)

Address(es): EPTPC0.SYDOMR 000C 4844h, EPTPC1.SYDOMR 000C 4C44h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	DNUM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DNUM[7:0]	domainNumber Field Value Setting	These bits are used to set the domainNumber field value of the PTP v2 header. When a message is received, this value is compared with the domainNumber field of the received frame as a condition for PTP reception processing. In generating messages, the value is used for the domainNumber field of the frame for transmission.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SYDOMR register is used to set the domainNumber field value of the PTP v2 message header. The value should not be modified while reception or transmission of PTP messages is enabled.

36.2.43 Announce Message Flag Field Setting Register (ANFR)

Address(es): EPTPC0.ANFR 000C 4850h, EPTPC1.ANFR 000C 4C50h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	FLAG14	FLAG13	—	—	FLAG10	—	FLAG8	—	—	FLAG5	FLAG4	FLAG3	FLAG2	FLAG1	FLAG0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FLAG0	leap61	This bit is used to set the logical value of the leap61 member of timePropertiesDS. 0: leap61 is set to FALSE. 1: leap61 is set to TRUE.	R/W
b1	FLAG1	leap59	This bit is used to set the logical value of the leap59 member of timePropertiesDS. 0: leap59 is set to FALSE. 1: leap59 is set to TRUE.	R/W
b2	FLAG2	currentUtcOffsetValid	This bit is used to set the logical value of the currentUtcOffsetValid member of timePropertiesDS. 0: currentUtcOffsetValid is set to FALSE. 1: currentUtcOffsetValid is set to TRUE.	R/W
b3	FLAG3	ptpTimescale	This bit is used to set the logical value of the ptpTimescale member of timePropertiesDS. 0: ptpTimescale is set to FALSE. 1: ptpTimescale is set to TRUE.	R/W
b4	FLAG4	timeTraceable	This bit is used to set the logical value of the timeTraceable member of timePropertiesDS. 0: timeTraceable is set to FALSE. 1: timeTraceable is set to TRUE.	R/W
b5	FLAG5	frequencyTraceable	This bit is used to set the logical value of the frequencyTraceable member of timePropertiesDS. 0: frequencyTraceable is set to FALSE. 1: frequencyTraceable is set to TRUE.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	FLAG8	alternateMasterFlag	0: alternateMasterFlag is set to FALSE. 1: alternateMasterFlag is set to TRUE.	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	FLAG10	unicastFlag	0: unicastFlag is set to FALSE. 1: unicastFlag is set to TRUE.	R/W
b12, b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	FLAG13	PTP profile Specific 1	0: PTP profile Specific 1 is set to FALSE. 1: PTP profile Specific 1 is set to TRUE.	R/W
b14	FLAG14	PTP profile Specific 2	0: PTP profile Specific 2 is set to FALSE. 1: PTP profile Specific 2 is set to TRUE.	R/W
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ANFR register is used to set the flagField section of the header when the SYNFP module is to generate an Announce message.

The value specified by the ANFR register is only reflected in the SYNFP module after the SYRVLDR.ANUP bit is set to 1.

36.2.44 Sync Message Flag Field Setting Register (SYNFR)

Address(es): EPTPC0.SYNFR 000C 4854h, EPTPC1.SYNFR 000C 4C54h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	FLAG14	FLAG13	—	—	FLAG10	FLAG9	FLAG8	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	FLAG8	alternateMasterFlag	0: alternateMasterFlag is set to FALSE. 1: alternateMasterFlag is set to TRUE.	R/W
b9	FLAG9	twoStepFlag	Set this bit to 0 (FALSE).	R/W
b10	FLAG10	unicastFlag	0: unicastFlag is set to FALSE. 1: unicastFlag is set to TRUE.	R/W
b12, b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	FLAG13	PTP profile Specific 1	0: PTP profile Specific 1 is set to FALSE. 1: PTP profile Specific 1 is set to TRUE.	R/W
b14	FLAG14	PTP profile Specific 2	0: PTP profile Specific 2 is set to FALSE. 1: PTP profile Specific 2 is set to TRUE.	R/W
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SYNFR register is used to set the flagField section of the header when the SYNFP module is to generate a Sync message.

The value specified by the SYNFR register is only reflected in the SYNFP module after the SYRVLDR.STUP bit is set to 1.

36.2.45 Delay_Req Message Flag Field Setting Register (DYRQFR)

Address(es): EPTPC0.DYRQFR 000C 4858h, EPTPC1.DYRQFR 000C 4C58h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	FLAG14	FLAG13	—	—	FLAG10	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10	FLAG10	unicastFlag	0: unicastFlag is set to FALSE. 1: unicastFlag is set to TRULE.	R/W
b12, b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	FLAG13	PTP profile Specific 1	0: PTP profile Specific 1 is set to FALSE. 1: PTP profile Specific 1 is set to TRULE.	R/W
b14	FLAG14	PTP profile Specific 2	0: PTP profile Specific 2 is set to FALSE. 1: PTP profile Specific 2 is set to TRULE.	R/W
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Register DYRQFR is used to set the flagField section of the header when the SYNFP module is to generate a Delay_Req or Pdelay_Req message.
The value specified by DYRQFR is only reflected in the SYNFP module after the SYRVLDR.STUP bit is set to 1.

36.2.46 Delay_Resp Message Flag Field Setting Register (DYRPFR)

Address(es): EPTPC0.DYRPFR 000C 485Ch, EPTPC1.DYRPFR 000C 4C5Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	FLAG14	FLAG13	—	—	FLAG10	FLAG9	FLAG8	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	FLAG8	alternateMasterFlag*1	0: alternateMasterFlag is set to FALSE. 1: alternateMasterFlag is set to TRUE.	R/W
b9	FLAG9	twoStepFlag*2	Set this bit to 0 (FALSE).	R/W
b10	FLAG10	unicastFlag	0: unicastFlag is set to FALSE. 1: unicastFlag is set to TRUE.	R/W
b12, b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	FLAG13	PTP profile Specific 1	0: PTP profile Specific 1 is set to FALSE. 1: PTP profile Specific 1 is set to TRUE.	R/W
b14	FLAG14	PTP profile Specific 2	0: PTP profile Specific 2 is set to FALSE. 1: PTP profile Specific 2 is set to TRUE.	R/W
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit is reserved for Pdelay_Resp messages. Set the bit to 0.
Note 2. This bit is reserved for Delay_Resp messages.

The DYRPFR register is used to set the flagField section of the header when the SYNFP module is to generate a Delay_Resp or PDelay_Resp message.
The value specified by the DYRPFR register is only reflected in the SYNFP module after the SYRVLDR.STUP bit is set to 1.
Do not change the value of the DYRPFR register while processing for the transmission of Delay_Resp or Pdelay_Resp messages is enabled. Furthermore, after disabling this processing for transmission, do not change the value of the DYRPFR register until the SYSR.RESDN flag becomes 1.

36.2.47 SYNFP Local Clock ID Registers (SYCIDRU, SYCIDRL)

Address(es): EPTPC0.SYCIDRU 000C 4860h, EPTPC1.SYCIDRU 000C 4C60h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the higher-order 32 bits of the clock-ID of the local port.	R/W

Address(es): EPTPC0.SYCIDRL 000C 4864h, EPTPC1.SYCIDRL 000C 4C64h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the lower-order 32 bits of the clock-ID of the local port.	R/W

The SYCIDR registers are used to set the clock-ID of the local port.
The registers are used for the clockIdentity section in the sourcePortIdentity field of the header when the SYNFP module is to generate a PTP message.
When a PTP message is received, the value of these registers is compared with the clockIdentity section in the sourcePortIdentity field of the PTP message to determine whether the message is one that was transmitted by you.
Usually, the setting should be the same as the value of portDS.portIdentity.clockIdentity.
Do not change the value of these registers while processing for the reception or transmission of PTP messages is enabled.

36.2.48 SYNFP Local Port Number Register (SYPNUMR)

Address(es): EPTPC0.SYPNUMR 000C 4868h, EPTPC1.SYPNUMR 000C 4C68h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PNUM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	PNUM[15:0]	Local Port Number Setting	These bits hold the setting for the port number of the local port.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SYPNUMR register is used to set the port number of the local port.
This register is used for the portNumber section in the sourcePortIdentity field of the header when the SYNFP module is to generate a PTP message.
When a PTP message is received, the value of this register is compared with the portNumber section in the sourcePortIdentity field of the PTP message to determine whether the message is one that was transmitted by the local device.
Usually, the setting should be the same as the value of portDS.portIdentity.portNumber.
Do not change the value of this register while processing for the reception or transmission of PTP messages is enabled.

36.2.49 SYNFP Register Value Load Directive Register (SYRVLDR)

Address(es): EPTPC0.SYRVLDR 000C 4880h, EPTPC1.SYRVLDR 000C 4C80h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ANUP	STUP	BMUP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BMUP	BMC Update	When this bit is set to 1, the SYNFP module simultaneously reflects values of registers storing the MasterClock identifying information.	W
b1	STUP	State Update	When this bit is set to 1, the SYNFP module simultaneously reflects register values for PTP message reception and transmission.	W
b2	ANUP	Announce Message Generation Information Update	When this bit is set to 1, the Announce message generation block simultaneously reflects register values required for generating Announce messages.	W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SYRVLDR register is used for simultaneously updating multiple register values in the SYNFP module.

BMUP Bit (BMC Update)

When the BMUP bit is set to 1, the SYNFP module simultaneously reflects the values of registers storing the MasterClock identifying information listed below.

- Registers MTCIDU and MTCIDL
- MTPID register

STUP Bit (State Update)

When the STUP bit is set to 1, the SYNFP module simultaneously reflects the values of registers and bits for PTP message reception and transmission listed below.

- SYNFR register
- DYRQFR register
- SYTLIR.DREQ[7:0] bits
- RSTOUTR register
- SYRFL1R register
- SYRFL2R register
- SYTRENR register

ANUP Bit (Announce Message Generation Information Update)

When the ANUP bit is set to 1, the Announce message generation block simultaneously reflects the values of registers and bits required for generating Announce messages listed below.

- ANFR register
- SYTLIR.ANCE[7:0] bits
- GMPR register
- GMCQR register
- Registers GMIDRU and GMIDRL
- CUOTSR register
- SRR register

36.2.50 SYNFP Reception Filter Register 1 (SYRFL1R)

Address(es): EPTPC0.SYRFL1R 000C 4890h, EPTPC1.SYRFL1R 000C 4C90h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	PDFUP[2:0]			—	PDRP[2:0]			—	PDRQ[2:0]			—	DRP[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	DRQ[2:0]			—	FUP[2:0]			—	SYNC[2:0]			—	—	ANCE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANCE[1:0]	Announce Message Processing	0: Messages are not transferred to the PTPEDMAC. 1: Messages are transferred to the PTPEDMAC.	R/W
b1			0: The PRC-TC does not relay messages between ports 0 and 1. 1: The PRC-TC relays messages between ports 0 and 1.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	SYNC[2:0]	Sync Message Processing	0: Messages are not transferred to the PTPEDMAC. 1: Messages are transferred to the PTPEDMAC.	R/W
b5			0: The PRC-TC does not relay messages between ports 0 and 1. 1: The PRC-TC relays messages between ports 0 and 1.	R/W
b6			0: The SYNFP does not process messages. 1: The SYNFP processes messages.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R
b8	FUP[2:0]	Follow_Up Message Processing	0: Messages are not transferred to the PTPEDMAC. 1: Messages are transferred to the PTPEDMAC.	R/W
b9			0: The PRC-TC does not relay messages between ports 0 and 1. 1: The PRC-TC relays messages between ports 0 and 1.	R/W
b10			0: The SYNFP does not process messages. 1: The SYNFP processes messages.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R
b12	DRQ[2:0]	Delay_Req Message Processing	0: Messages are not transferred to the PTPEDMAC. 1: Messages are transferred to the PTPEDMAC.	R/W
b13			0: The PRC-TC does not relay messages between ports 0 and 1. 1: The PRC-TC relays messages between ports 0 and 1.	R/W
b14			0: The SYNFP does not process messages. 1: The SYNFP processes messages.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R
b16	DRP[2:0]	Delay_Resp Message Processing	0: Messages are not transferred to the PTPEDMAC. 1: Messages are transferred to the PTPEDMAC.	R/W
b17			0: The PRC-TC does not relay messages between ports 0 and 1. 1: The PRC-TC relays messages between ports 0 and 1.	R/W
b18			0: The SYNFP does not process messages. 1: The SYNFP processes messages.	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R
b20	PDRQ[2:0]	Pdelay_Req Message Processing	0: Messages are not transferred to the PTPEDMAC. 1: Messages are transferred to the PTPEDMAC.	R/W
b21			0: The PRC-TC does not relay messages between ports 0 and 1. 1: The PRC-TC relays messages between ports 0 and 1.	R/W
b22			0: The SYNFP does not process messages. 1: The SYNFP processes messages.	R/W

Bit	Symbol	Bit Name	Description	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R
b24	PDRP[2:0]	Pdelay_Resp Message Processing	0: Messages are not transferred to the PTPEDMAC. 1: Messages are transferred to the PTPEDMAC.	R/W
b25			0: The PRC-TC does not relay messages between ports 0 and 1. 1: The PRC-TC relays messages between ports 0 and 1.	R/W
b26			0: The SYNFP does not process messages. 1: The SYNFP processes messages.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R
b28	PDFUP[2:0]	Pdelay_Resp_Follow_Up Message Processing	0: Messages are not transferred to the PTPEDMAC. 1: Messages are transferred to the PTPEDMAC.	R/W
b29			0: The PRC-TC does not relay messages between ports 0 and 1. 1: The PRC-TC relays messages between ports 0 and 1.	R/W
b30			0: The SYNFP does not process messages. 1: The SYNFP processes messages.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R

The SYRFL1R register is used to set up filtering for the reception of PTP messages.
Multiple bits corresponding to different types of messages can be set to 1. Setting all bits for a type of message to 0 leads to all messages of the given type being discarded.
The value specified by the SYRFL1R register is only reflected in the SYNFP module after the SYRVLDR.STUP bit is set to 1.

36.2.51 SYNFP Reception Filter Register 2 (SYRFL2R)

Address(es): EPTPC0.SYRFL2R 000C 4894h, EPTPC1.SYRFL2R 000C 4C94h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	ILL[1:0]		—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	SIG[1:0]		—	—	MAN[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MAN[1:0]	Management Message Processing Setting	0: Messages are not transferred to the PTPEDMAC. 1: Messages are transferred to the PTPEDMAC.	R/W
b1			0: The PRC-TC does not relay messages between ports 0 and 1. 1: The PRC-TC relays messages between ports 0 and 1.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	SIG[1:0]	Signaling Message Processing Setting	0: Messages are not transferred to the PTPEDMAC. 1: Messages are transferred to the PTPEDMAC.	R/W
b5			0: The PRC-TC does not relay messages between ports 0 and 1. 1: The PRC-TC relays messages between ports 0 and 1.	R/W
b27 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b28	ILL[1:0]	Illegal Message Processing Setting*1	0: Messages are not transferred to the PTPEDMAC. 1: Messages are transferred to the PTPEDMAC.	R/W
b29			0: The PRC-TC does not relay messages between ports 0 and 1. 1: The PRC-TC relays messages between ports 0 and 1.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. PTP messages other than PTP v2 messages and messages of undefined type are handled as illegal messages.

The SYRFL2R register is used to set up filtering for the reception of PTP messages.
Multiple bits corresponding to different types of messages can be set to 1. Setting all bits for a type of message to 0 leads to all messages of the given type being discarded.
The value specified by the SYRFL2R register is only reflected in the SYNFP module after the SYRVLDR.STUP bit is set to 1.

36.2.52 SYNFP Transmission Enable Register (SYTRENR)

Address(es): EPTPC0.SYTRENR 000C 4898h, EPTPC1.SYTRENR 000C 4C98h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PDRQ	—	—	—	DRQ	—	—	—	SYNC	—	—	—	ANCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANCE	Announce Message Transmission Enable	0: Announce messages are not transmitted. 1: Announce messages are transmitted.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	SYNC	Sync Message Transmission Enable	0: Sync messages are not transmitted. 1: Sync messages are transmitted.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	DRQ	Delay_Req Message Transmission Enable	0: Delay_Req messages are not transmitted. 1: Delay_Req messages are transmitted.	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	PDRQ	Pdelay_Req Message Transmission Enable	0: Pdelay_Req messages are not transmitted. 1: Pdelay_Req messages are transmitted.	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SYTRENR register is used to enable or disable transmission of PTP messages.
The PDRQ and DRQ bits should not be set to 1 at the same time. Operation is not guaranteed when both bits are set to 1.
The value specified by the SYTRENR register is only reflected in the SYNFP module after the SYRVLDR.STUP bit is set to 1.

36.2.53 Master Clock ID Registers (MTCIDU, MTCIDL)

Address(es): EPTPC0.MTCIDU 000C 48A0h, EPTPC1.MTCIDU 000C 4CA0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the higher-order 32 bits of the clock-ID of the master clock.	R/W

Address(es): EPTPC0.MTCIDL 000C 48A4h, EPTPC1.MTCIDL 000C 4CA4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the lower-order 32 bits of the clock-ID of the master clock.	R/W

Registers MTCIDU and MTCIDL are used to set the clock-ID of the master for synchronization.
The value specified by registers MTCIDU and MTCIDL is only reflected in the SYNFP module after the SYRVLDR.BMUP bit is set to 1.

36.2.54 Master Clock Port Number Register (MTPID)

Address(es): EPTPC0.MTPID 000C 48A8h, EPTPC1.MTPID 000C 4CA8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PNUM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	PNUM[15:0]	Master Clock Port Number Setting	These bits hold the setting for the port number of the master clock.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

The MTPID register is used to set the port number of the master for synchronization.
The value specified by the MTPID register is only reflected in the SYNFP module after the SYRVLDR.BMUP bit is set to 1.
In normal usage, set the value of parentDS.parentPortIdentity.portNumber in this register.

36.2.55 SYNFP Transmission Interval Setting Register (SYTLIR)

Address(es): EPTPC0.SYTLIR 000C 48C0h, EPTPC1.SYTLIR 000C 4CC0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DREQ[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC[7:0]								ANCE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ANCE[7:0]	Announce Message Transmission Interval Setting	These bits set the interval for the transmission of Announce messages.	R/W
b15 to b8	SYNC[7:0]	Sync Message Transmission Interval Setting	These bits set the interval for the transmission of Sync messages. The setting is also placed in the logMessageInterval field of transmitted Sync messages.	R/W
b23 to b16	DREQ[7:0]	Delay_Req Transmission Interval Average Value/ Pdelay_Req Transmission Interval Setting	The bits set the average interval for the transmission of Delay_Req messages and the interval for the transmission of Pdelay_Req messages. The setting is also placed in the logMessageInterval field of Delay_Resp messages.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SYTLIR register is used to set the interval for the transmission of messages generated by the SYNFP module. The setting is an integer logarithm in base 2 ($\log_2(x)$) and determines a value x in seconds.
In other words, the interval for transmission is 2^n (s), where n is the setting. The available settings are from -7 (F9h) to +6 (06h).
Example:
If the setting is 06h, then the interval for transmission is $2^6 = 64$ (s).
If the setting is 00h, then the interval for transmission is $2^0 = 1$ (s).
If the setting is FFh, then the interval for transmission is $2^{-1} = 0.5$ (s) = 500 (ms).
If the setting is F9h, then the interval for transmission is $2^{-7} = 0.0078125$ (s) = 7.8125 (ms).

The value set in the ANCE[7:0] bits is only reflected in the SYNFP module after the SYRVLDR.ANUP bit is set to 1.
The value set in the DREQ[7:0] and SYNC[7:0] bits is only reflected in the SYNFP module after the SYRVLDR.STUP bit is set to 1.

36.2.56 SYNFP Received logMessageInterval Value Indication Register (SYRLIR)

Address(es): EPTPC0.SYRLIR 000C 48C4h, EPTPC1.SYRLIR 000C 4CC4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DRESP[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC[7:0]								ANCE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ANCE[7:0]	Announce Message logMessageInterval Field Indication Flag	These bits indicate the logMessageInterval field value of a received Announce message.	R
b15 to b8	SYNC[7:0]	Sync Message logMessageInterval Field Indication Flag	These bits indicate the logMessageInterval field value of a received Sync message.	R
b23 to b16	DRESP[7:0]	Delay_Resp Message logMessageInterval Field Indication Flag	These bits indicate the logMessageInterval field value of a received Delay_Resp message.	R
b31 to b24	—	Reserved	These bits are read as 0.	R

The SYRLIR register indicates the logMessageInterval field values of received PTP messages.

36.2.57 offsetFromMaster Value Registers (OFMRU, OFMRL)

Address(es): EPTPC0.OFMRU 000C 48C8h, EPTPC1.OFMRU 000C 4CC8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits indicate the higher-order 32 bits of the calculated offsetFromMaster value.	R

Address(es): EPTPC0.OFMRL 000C 48CCh, EPTPC1.OFMRL 000C 4CCCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits indicate the lower-order 32 bits of the calculated offsetFromMaster value.	R

Registers OFMRU and OFMRL indicate the calculated offsetFromMaster value.
The value is expressed as two's complements in units of nanoseconds. Note that the numeric representation*1 differs from that of the offsetFromMaster member of the current data set (currentDS).
When read, access to the registers should be in order of OFMRU and then OFMRL.

Note 1. The value of currentDS.offsetFromMaster is multiplied by 2^{16}.
Example: 2.5 (ns) = 00000000_00028000h

36.2.58 meanPathDelay Value Registers (MPDRU, MPDRL)

Address(es): EPTPC0.MPDRU 000C 48D0h, EPTPC1.MPDRU 000C 4CD0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits indicate the higher-order 32 bits of the calculated meanPathDelay value.	R

Address(es): EPTPC0.MPDRL 000C 48D4h, EPTPC1.MPDRL 000C 4CD4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits indicate the lower-order 32 bits of the calculated meanPathDelay value.	R

Registers MPDRU and MPDRL indicate the calculated meanPathDelay value.
The value is expressed as two's complements in units of nanoseconds. Note that the numeric representation*1 differs from that of the meanPathDelay member of the current data set (currentDS).
When read, access to the registers should be in order of MPDRU and then MPDRL.

Note 1. The value of currentDS.meanPathDelay is multiplied by 2^{16}.
Example: 2.5 (ns) = 00000000_00028000h

36.2.59 grandmasterPriority Field Setting Register (GMPR)

Address(es): EPTPC0.GMPR 000C 48E0h, EPTPC1.GMPR 000C 4CE0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	GMPR1[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	GMPR2[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	GMPR2[7:0]	grandmasterPriority2 Field Value Setting	These bits are used to set the value of the grandmasterPriority2 fields of Announce messages.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R
b23 to b16	GMPR1[7:0]	grandmasterPriority1 Field Value Setting	These bits are used to set the value of the grandmasterPriority1 fields of Announce messages.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

The GMPR register is used to specify the grandmasterPriority1 and grandmasterPriority2 field values of Announce messages generated by the SYNFP module.
The value specified by the GMPR register is only reflected in the SYNFP module after the SYRVLDR.ANUP bit is set to 1.

36.2.60 grandmasterClockQuality Field Setting Register (GMCQR)

Address(es): EPTPC0.GMCQR 000C 48E4h, EPTPC1.GMCQR 000C 4CE4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits are used to set the value of the grandmasterClockQuality fields of Announce messages. The correspondence between bits and the grandmasterClockQuality fields is as listed below. b31 to b24: clockClass b23 to b16: clockAccuracy b15 to b0: offsetScaledLogVariance	R/W

The GMCQR register is used to specify the grandmasterClockQuality field value of Announce messages generated by the SYNFP module.

The value specified by the GMCQR register is only reflected in the SYNFP module after the SYRVLDR.ANUP bit is set to 1.

36.2.61 grandmasterIdentity Field Setting Registers (GMIDRU, GMIDRL)

Address(es): EPTPC0.GMIDRU 000C 48E8h, EPTPC1.GMIDRU 000C 4CE8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the higher-order 32 bits of the value of the grandmasterIdentity fields of Announce messages.	R/W

Address(es): EPTPC0.GMIDRL 000C 48ECh, EPTPC1.GMIDRL 000C 4CECh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the lower-order 32 bits of the value of the grandmasterIdentity fields of Announce messages.	R/W

Registers GMIDRU and GMIDRL are used to specify the grandmasterIdentity field value of Announce messages generated by the SYNFP module.
The value specified by registers GMIDRU and GMIDRL is only reflected in the SYNFP module after the SYRVLDR.ANUP bit is set to 1.

36.2.62 currentUtcOffset/timeSource Field Setting Register (CUOTSR)

Address(es): EPTPC0.CUOTSR 000C 48F0h, EPTPC1.CUOTSR 000C 4CF0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CUTO[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TSRC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TSRC[7:0]	timeSource Field Setting	These bits set the value of the timeSource fields of Announce messages.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R
b31 to b16	CUTO[15:0]	currentUtcOffset Field Setting	These bits set the value of the currentUtcOffset fields of Announce messages.	R/W

The CUOTSR register is used to specify the currentUtcOffset and timeSource field values of Announce messages generated by the SYNFP module.
The value specified by the CUOTSR register is only reflected in the SYNFP module after the SYRVLDR.ANUP bit is set to 1.

36.2.63 stepsRemoved Field Setting Register (SRR)

Address(es): EPTPC0.SRR 000C 48F4h, EPTPC1.SRR 000C 4CF4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SRMV[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	SRMV[15:0]	stepsRemoved Field Value Setting	These bits set the value of the stepsRemoved fields of Announce messages.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SRR register is used to specify the stepsRemoved field value of Announce messages generated by the SYNFP module.
The value specified by the SRR register is only reflected in the SYNFP module after the SYRVLDR.ANUP bit is set to 1.

36.2.64 PTP-primary Message Destination MAC Address Setting Registers (PPMACRU, PPMACRL)

Address(es): EPTPC0.PPMACRU 000C 4900h, EPTPC1.PPMACRU 000C 4D00h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	1	1	0	1	1	0	0	0	1	1	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	—	These bits hold the setting for the higher-order 24 bits of the destination MAC address for PTP-primary messages.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

Address(es): EPTPC0.PPMACRL 000C 4904h, EPTPC1.PPMACRL 000C 4D04h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	—	These bits hold the setting for the lower-order 24 bits of the destination MAC address for PTP-primary messages.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

The PPMACR registers are used to specify the destination MAC address for PTP-primary messages. In normal usage, set 01:1B:19:00:00:00 in these registers.

The value is used in the destination MAC address field when generating an Ethernet frame for a PTP-primary message. It is also used as a determining condition for received frames carrying PTP messages.

Set a value in these registers before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.65 PTP-pdelay Message MAC Address Setting Registers (PDMACRU, PDMACRL)

Address(es): EPTPC0.PDMACRU 000C 4908h, EPTPC1.PDMACRU 000C 4D08h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	—	These bits hold the setting for the higher-order 24 bits of the destination MAC address for PTP-pdelay messages.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

Address(es): EPTPC0.PDMACRL 000C 490Ch, EPTPC1.PDMACRL 000C 4D0Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	—	These bits hold the setting for the lower-order 24 bits of the destination MAC address for PTP-pdelay messages.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

The PDMACR registers are used to specify the destination MAC address for PTP-pdelay messages. In normal usage, set 01:80:C2:00:00:0E in these registers.

The value is used in the destination MAC address field when generating frames carrying PTP-pdelay messages in the Ethernet format. It is also used as a determining condition for received frames carrying PTP messages.

Set a value in these registers before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.66 PTP Message Ethertype Setting Register (PETYPER)

Address(es): EPTPC0.PETYPER 000C 4910h, EPTPC1.PETYPER 000C 4D10h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TYPE[15:0]															
Value after reset:	1	0	0	0	1	0	0	0	1	1	1	1	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TYPE[15:0]	PTP Message Ethertype Value Setting	These bits hold the setting for the Ethertype field value for frames in the Ethernet II format.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

The PETYPER register is used to specify the Ethertype field for frames carrying the PTP messages. In normal usage, set 0000 88F7h in this register.

The value is used in the Ethertype field when generating frames carrying PTP messages in the Ethernet II format. It is also used as a determining condition for received frames carrying PTP messages.

Set a value in these registers before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.67 PTP-primary Message Destination IP Address Setting Register (PPIPR)

Address(es): EPTPC0.PPIPR 000C 4920h, EPTPC1.PPIPR 000C 4D20h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the destination IP address for PTP-primary messages.	R/W

The PPIPR register is used to specify the destination IP address for PTP messages. In normal usage, set E000 0181h (224.0.1.129) in this register.

The value is used in the destination IP address field when generating frames carrying PTP-primary messages in the IPv4 format. The lower-order 23 bits are also used in the destination MAC address field for Ethernet frames. The value is also used as a determining condition for received frames carrying PTP messages.

Set a value in this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.68 PTP-pdelay Message Destination IP Address Setting Register (PDIPR)

Address(es): EPTPC0.PDIPR 000C 4924h, EPTPC1.PDIPR 000C 4D24h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the destination IP address for PTP-pdelay messages.	R/W

The PDIPR register is used to specify the destination IP address for PTP-pdelay messages. In normal usage, set E000 006Bh (224.0.0.107) in this register.
The value is used in the destination IP address field when generating frames carrying PTP-pdelay messages in the IPv4 format. The lower-order 23 bits are also used in the destination MAC address field for Ethernet frames. The value is also used as a determining condition for received frames carrying PTP messages.
Set a value in this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.69 PTP Event Message TOS Setting Register (PETOSR)

Address(es): EPTPC0.PETOSR 000C 4928h, EPTPC1.PETOSR 000C 4D28h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	EVTO[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	EVTO[7:0]	PTP Event Message TOS Field Value Setting	These bits hold the setting for the value of the TOS field within the IPv4 headers of PTP event messages.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

The PETOSR register is used to set the TOS (type of service) field value within the IPv4 headers of PTP event messages. Set a value in this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.70 PTP general Message TOS Setting Register (PGTOSR)

Address(es): EPTPC0.PGTOSR 000C 492Ch, EPTPC1.PGTOSR 000C 4D2Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	GETO[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	GETO[7:0]	PTP general Message TOS Field Value Setting	These bits hold the setting for the value of the TOS field within the IPv4 headers of PTP general messages.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

The PGTOSR register is used to set the TOS (type of service) field value within the IPv4 headers of PTP general messages.
Set a value in this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.71 PTP-primary Message TTL Setting Register (PPTTLR)

Address(es): EPTPC0.PPTTLR 000C 4930h, EPTPC1.PPTTLR 000C 4D30h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	PRTL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PRTL[7:0]	PTP-primary Message TTL Field Value Setting	These bits hold the setting for the value of the TTL field within the IPv4 headers of PTP-primary messages.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

The PPTTLR register is used to set the TTL (time to live) field value within the IPv4 headers of PTP-primary messages. Set a value in this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.72 PTP-pdelay Message TTL Setting Register (PDTTLR)

Address(es): EPTPC0.PDTTLR 000C 4934h, EPTPC1.PDTTLR 000C 4D34h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	PDTL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PDTL[7:0]	PTP-pdelay Message TTL Field Value	These bits hold the setting for the value of the TTL field within the IPv4 headers of PTP-pdelay messages.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

The PDTTLR register is used to set the TTL field value within the IPv4 headers of PTP-pdelay messages.
Set a value in this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.73 PTP Event Message UDP Destination Port Number Setting Register (PEUDPR)

Address(es): EPTPC0.PEUDPR 000C 4938h, EPTPC1.PEUDPR 000C 4D38h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EVUPT[15:0]															
Value after reset:	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	EVUPT[15:0]	PTP Event Message Destination Port Number Setting	These bits hold the setting for the value of the destination port number field within the UDP headers of PTP event messages.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

The PEUDPR register is used to set the destination port number field value within the UDP headers of PTP event messages. In normal usage, set 013Fh (319) in this register.
Set a value in this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.74 PTP general Message UDP Destination Port Number Setting Register (PGUDPR)

Address(es): EPTPC0.PGUDPR 000C 493Ch, EPTPC1.PGUDPR 000C 4D3Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GEUPT[15:0]															
Value after reset:	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	GEUPT[15:0]	PTP general Message Destination Port Number	These bits hold the setting for the value of the destination port number field within the UDP headers of PTP general messages.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

The PGUDPR register is used to set the destination port number field value within the UDP headers of PTP general messages. In normal usage, set 0140h (320) in this register.

Set a value in this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.75 Frame Reception Filter Setting Register (FFLTR)

Address(es): EPTPC0.FFLTR 000C 4940h, EPTPC1.FFLTR 000C 4D40h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EXTPRM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ENB	PRT	SEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SEL	Receive MAC Address Select*1	These bits select how filtering is handled when multicast frames other than PTP messages are received. b2 b1 b0 0 0 0: Filtering is disabled (all multicast frames are received). 0 0 1: Filtering is disabled (all multicast frames are received). 0 1 0: Filtering is disabled (all multicast frames are received). 0 1 1: Filtering is disabled (all multicast frames are received). 1 0 0: Do not receive multicast frames. 1 0 1: Do not receive multicast frames. 1 1 0: Only receive multicast frames matching the MAC address setting in registers FMAC0RU and FMAC0RL. 1 1 1: Only receive multicast frames matching the MAC address setting in registers FMAC1RU and FMAC1RL).	R/W
b1	PRT	Frame Reception Enable*1		R/W
b2	ENB	Reception Filter Enable*1		R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	EXTPRM	Extended Promiscuous Mode Setting	0: Normal operation (unicast frames addressed to the EPTPC are received, filtering of PTP frames is applied, multicast filtering is applied, and all broadcast frames are received). 1: Extended promiscuous mode (all frames are received)	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. The setting of these bits is only effective when the EXTPRM bit is 0.

The FFLTR register is used to switch extended promiscuous mode on or off and to select how filtering is handled when multicast frames other than PTP messages are received.

The filter for the reception of multicast frames other than PTP messages is enabled by setting the ENB, PRT, and SEL bits to 110b or 111b. Frames passed by the filter are then transferred by EDMACn.

Relaying of multicast frames other than PTP messages is in accord with the TRNMR register and relaying and reception of PTP messages is in accord with the setting of registers SYRFL1R and SYRFL2R.

Set a value in this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.76 Frame Reception Filter MAC Address 0 Setting Registers (FMAC0RU, FMAC0RL)

Address(es): EPTPC0.FMAC0RU 000C 4960h, EPTPC1.FMAC0RU 000C 4D60h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	—	These bits hold the setting for the higher-order 24 bits of the destination MAC address for received multicast frames.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

Address(es): EPTPC0.FMAC0RL 000C 4964h, EPTPC1.FMAC0RL 000C 4D64h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	—	These bits hold the setting for the lower-order 24 bits of the destination MAC address for received multicast frames.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

Registers FMAC0RU and FMAC0RL are used to specify the MAC address for filtering in the reception of multicast frames other than PTP messages.

Set a value in this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.77 Frame Reception Filter MAC Address 1 Setting Registers (FMAC1RU, FMAC1RL)

Address(es): EPTPC0.FMAC1RU 000C 4968h, EPTPC1.FMAC1RU 000C 4D68h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	—	These bits hold the setting for the higher-order 24 bits of the destination MAC address for received multicast frames.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

Address(es): EPTPC0.FMAC1RL 000C 496Ch, EPTPC1.FMAC1RL 000C 4D6Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	—	These bits hold the setting for the lower-order 24 bits of the destination MAC address for received multicast frames.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

Registers FMAC1RU and FMAC1RL are used to specify the MAC address for filtering in the reception of multicast frames other than PTP messages.

Set a value in this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.78 Asymmetric Delay Setting Registers (DASYMRU, DASYMRL)

Address(es): EPTPC0.DASYMRU 000C 49C0h, EPTPC1.DASYMRU 000C 4DC0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	—	These bits hold the setting for the higher-order 16 bits of the asymmetric delay value. Set them to 0000h in this MCU.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

Address(es): EPTPC0.DASYMRL 000C 49C4h, EPTPC1.DASYMRL 000C 4DC4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	These bits hold the setting for the lower-order 32 bits of the asymmetric delay value. Set them to 0000 0000h in this MCU.	R/W

Registers DASYMRU and DASYMRL are used to set the asymmetric delay value (delayAsymmetry).
Set the registers DASYMRU and DASYMRL to 0000 0000h in this MCU.

36.2.79 Timestamp Latency Setting Register (TSLATR)

Address(es): EPTPC0.TSLATR 000C 49C8h, EPTPC1.TSLATR 000C 4DC8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	INGP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EGP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	EGP[15:0]	Output Port Timestamp Latency Setting	These bits hold the setting for the time stamp latency (ns) for the output ports.	R/W
b31 to b16	INGP[15:0]	Input Port Timestamp Latency Setting	These bits hold the setting for the time stamp latency (ns) for the input ports.	R/W

The TSLATR register is used to set the amount of latency in timestamp acquisition in nanoseconds.
The value should not be modified while reception or transmission of PTP messages is enabled.

EGP[15:0] Bits (Output Port Timestamp Latency Setting)

Set the corresponding fixed value in Table 36.8. The timestamp latency differs with the link transfer rate (100 Mbps or 10 Mbps) and the frequency of the STCA clock (20, 25, 50, or 100 MHz).

Table 36.8 Settings of the EGP[15:0] Bits (ns)

		STCA Clock Frequency			
Link Transfer Rate		**20 MHz**	**25 MHz**	**50 MHz**	**100 MHz**
MII	100 Mbps	590	625	695	730
	10 Mbps	7430	7465	7535	7570
RMII	100 Mbps	770	805	875	910
	10 Mbps	9230	9265	9335	9370

INGP[15:0] Bits (Input Port Timestamp Latency Setting)

Set the corresponding fixed value in Table 36.9. The timestamp latency differs with the link transfer rate (100 Mbps or 10 Mbps) and the frequency of the STCA clock (20, 25, 50, or 100 MHz).

Table 36.9 Settings of the INGP[15:0] Bits (ns)

		STCA Clock Frequency			
Link Transfer Rate		**20 MHz**	**25 MHz**	**50 MHz**	**100 MHz**
MII	100 Mbps	980	945	875	840
	10 Mbps	8180	8145	8075	8015
RMII	100 Mbps	1060	1025	955	920
	10 Mbps	8980	8945	8875	8815

36.2.80 SYNFP Operation Setting Register (SYCONFR)

Address(es): EPTPC0.SYCONFR 000C 49CCh, EPTPC1.SYCONFR 000C 4DCCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	TCMOD	—	—	—	FILDIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SBDIS	—	—	—	—	TCYC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TCYC[7:0]	PTP Message Transmission Interval Setting	These bits are used to set the time from the completion of one transmission to the start of the next in cycles of the transmission clock. A value n in these bits means that a transmission interval of n cycles will be secured. No interval is secured if the setting is 00h. The recommended value is 28h (40 cycles).	R/W
b11 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	SBDIS	Sync Message Transmission Bandwidth Securing Disable	0: Securing of the bandwidth for the transmission of SYNC messages is enabled (transfer by the EDMAC is given lower priority). 1: Securing of the bandwidth for the transmission of SYNC messages is disabled (transfer by the EDMAC is given higher priority).	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	FILDIS	Receive Message domainNumber Filter Disable	0: Filtering conditions for the reception of PTP messages include comparison with the domainNumber field. 1: Filtering conditions for the reception of PTP messages do not include comparison with the domainNumber field.	R/W
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	TCMOD	TC Mode Setting	0: E2E TC 1: P2P TC	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SYCONFR register controls operation of the SYNFP module.
Set a value in this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

TCYC[7:0] Bits (PTP Message Transmission Interval Setting)

The TCYC[7:0] bits are used to secure the waiting time between packets to keep a fixed transmission delay time.
The setting defines the interval from input of the transmission completed signal from the ETHERC to output of the next transmission request as a number of cycles of the transmission clock (which runs at 2.5 MHz if the link transfer rate is 10 Mbps and at 25 MHz if the rate is 100 Mbps).

SBDIS Bit (Sync Message Transmission Bandwidth Securing Disable)

This bit disables securing of bandwidth to increase accuracy of the interval for the transmission of SYNC messages.

TCMOD Bit (TC Mode Setting)

This bit sets the delay mechanism for operation as a transparent clock (TC). The bit setting changes the method of calculating the values used in the correctionField field during operation as a TC.

36.2.81 SYNFP Frame Format Setting Register (SYFORMR)

Address(es): EPTPC0.SYFORMR 000C 49D0h, EPTPC1.SYFORMR 000C 4DD0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FORM1	FORM0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FORM0	Ethernet Frame Format Setting	Set this bit to 0 (Ethernet II frame format).	R/W
b1	FORM1	Ethernet/UDP Encapsulation	0: PTP directly over Ethernet 1: PTP over UDP/IPv4	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SYFORMR register is used to set the format for frame generation by the SYNFP module.
Set a value in this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

36.2.82 Response Message Reception Timeout Register (RSTOUTR)

Address(es): EPTPC0.RSTOUTR 000C 49D4h, EPTPC1.RSTOUTR 000C 4DD4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	Response Message Reception Timeout Time Setting	A response message not being received within n × 1024 (ns), where n is the setting, is judged to represent a timeout.	R/W

The RSTOUTR register is used to set the time for detection of a timeout in the reception of PTP response messages (Delay_Resp and Pdelay_Resp).
If a Delay_Resp or Pdelay_Resp message is not received within the time specified by this register after transmission of a Delay_Req or Pdelay_Req message, the SYSR.DRPTO flag becomes 1.
The value specified by this register is only reflected in the SYNFP module after the SYRVLDR.STUP bit is set to 1.

36.2.83 PTP Reset Register (PTRSTR)

Address(es): EPTPC.PTRSTR 000C 0500h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RESET
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RESET	EPTPC Software Reset	0: The EPTPC is not reset. 1: The EPTPC is reset.*1	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Do not access the EPTPC-related registers other than this register while a software reset is being issued.

The PTRSTR register is used to reset the EPTPC.
It takes 64 cycles of the peripheral module clock (PCLKA) until initialization of the EPTPC is completed. After the RESET bit is set to 1, wait for 64 cycles of the PCLKA before setting its value back to 0.

36.2.84 STCA Clock Select Register (STCSELR)

Address(es): EPTPC.STCSELR 000C 0504h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCLKSEL[2:0]			—	—	—	—	—	SCLKDIV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SCLKDIV[2:0]	PCLKA Clock Frequency Division	b2 b0 0 0 1: 1 0 1 0: 1/2 0 1 1: 1/3 1 0 0: 1/4 1 0 1: 1/5 1 1 0: 1/6 Settings other than above are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b10 to b8	SCLKSEL[2:0]	STCA Clock Select	b10 b8 0 0 0: PCLKA clock divided by 1 to 6 0 1 0: Input clock from the REF50CK0 pin 0 1 1: Input clock from the REF50CK1 pin Settings other than above are prohibited.	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R

The STCSELR register selects the STCA clock signal for use in the EPTPC.
Set a value in this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

SCLKDIV[2:0] Bits (PCLKA Clock Frequency Division)

These bits set the division ratio of PCLKA.
When the setting of the SCLKSEL[2:0] bits is 000b, the frequency-divided PCLKA is used as the STCA clock signal.

SCLKSEL[2:0] Bits (STCA Clock Select)

These bits select the STCA clock signal for use in the EPTPC.

36.3 Operation

The EPTPC is set not to receive (analyze) and transmit (generate) PTP messages after release from the reset state. Accordingly, the EPTPC has no effect on the transmission and reception of frames by the ETHERC and EDMAC at that time.

Settings of the EPTPC registers are required for the ETHERC and EDMAC to be able to use packet filtering by MAC address in the SYNFP modules and for relaying between Ethernet ports by the PRC-TC module. That is, the transmission and reception of PTP messages requires setting of the EPTPC.

Figure 36.4 is a block diagram of the modules involved in transferring and relaying frames.

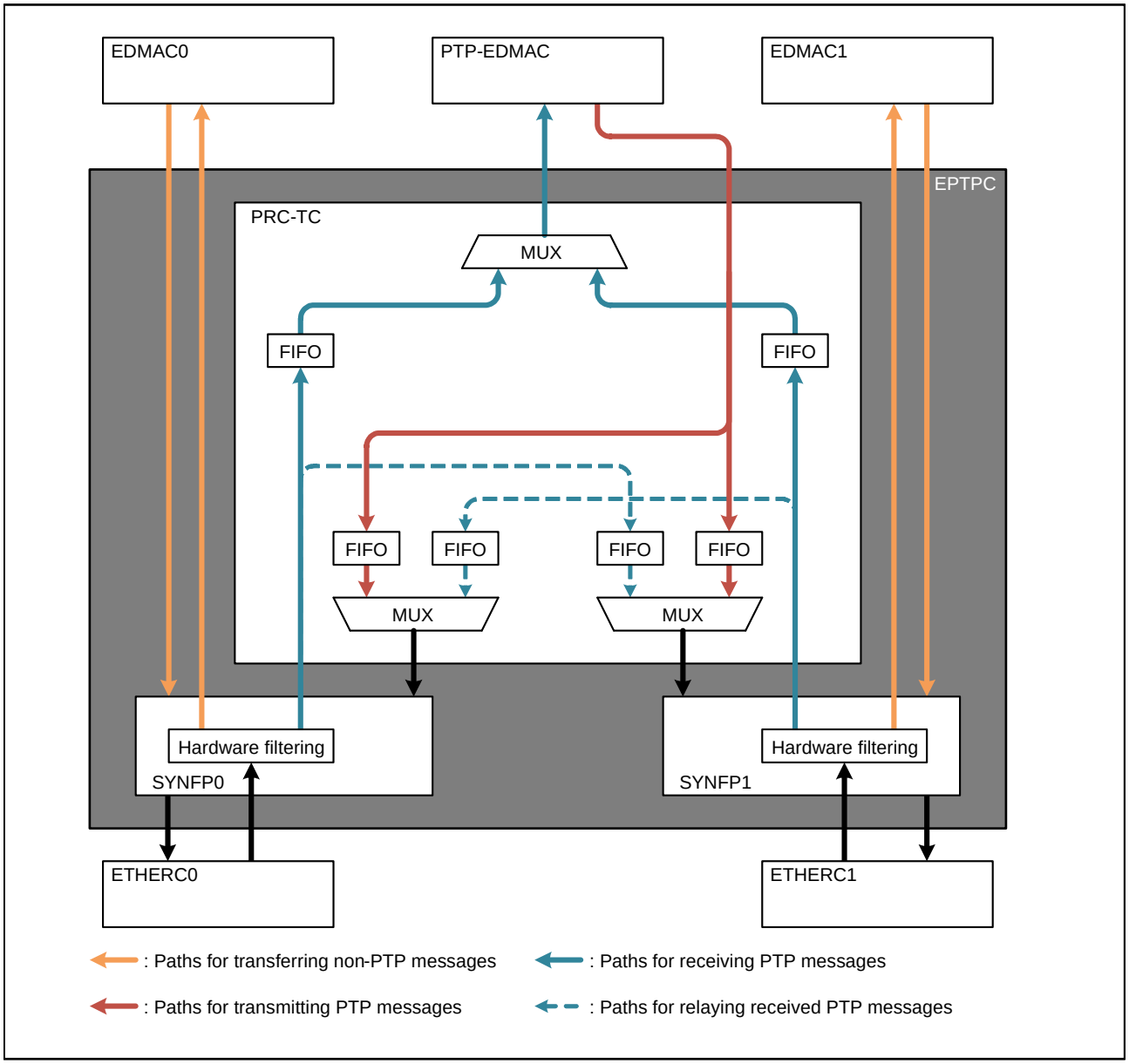


Figure 36.4 Block Diagram of the Modules Involved in Transferring and Relaying Frames

36.3.1 Transmission and Reception and Relaying of Non-PTP Messages

The EPTPC operates in extended promiscuous mode if the setting of the FFLTR.EXTPRM bit is 1 (extended promiscuous mode). In this mode, all frames received by the Ethernet ports are transferred to the EDMAC without filtering.
The EPTPC operates in normal mode if the setting of the FFLTR.EXTPRM bit is 0 (normal operation). In this mode, the SYNFP module applies its hardware filtering function to filter frames received by the Ethernet ports.
Frames for the other node can also be relayed to the Ethernet port on the other side in accord with the settings of the TRNMR.FWD0 and FWD1 bits.
The EPTPC and EDMAC transfer received unicast frames if they are for the given node. When a received frame is for the other node, the EPTPC is able to relay received frames to the other port if relaying between Ethernet ports is enabled.

Operation when multicast frames are received can be selected from the following: frames are transferred to the EDMAC, frames are not transferred to the EDMAC, or frames are transferred to the EDMAC only when the address matches the specified MAC address. The EPTPC also relays multicast frames to the other Ethernet port if relaying between Ethernet ports is enabled.
The EPTPC transfers received broadcast frames to the EDMAC for the receiving Ethernet port. The EPTPC also relays broadcast frames to the other Ethernet port if relaying between Ethernet ports is enabled.
The PRC-TC module, which handles the relaying of frames received by one Ethernet port to the other Ethernet port, has a relaying FIFO buffer for use in this process. Either of two methods for reading out from the relaying FIFO is selectable: store and forward or cut-through. With the cut-through method, reading out from the relaying FIFO starts when it holds more than one frame or more data than a specified threshold value, or when both of these conditions are satisfied. The TRNCTTDR.THVAL[10:0] bits specify the threshold value for reading out from the relaying FIFO.
With the cut-through method, when a frame is handled as an error frame due to a fault (such as malformed data being read) after a read operation starts, there is no procedure for notifying the ETHERC module of the error. Accordingly, the ETHERC may send such error frames over the line as if they were normal frames. Avoid situations of this kind by using the store-and-forward method with the relaying FIFO. With the store-and-forward method, the PRC-TC module internally discards error frames.

36.3.2 Paths for the Transfer of Non-PTP Messages

Messages received through the Ethernet port are transferred to the EDMAC. The PRC-TC module is capable of relaying messages between the Ethernet ports.

Figure 36.5 is a diagram of paths for the transmission and reception and relaying of non-PTP messages.

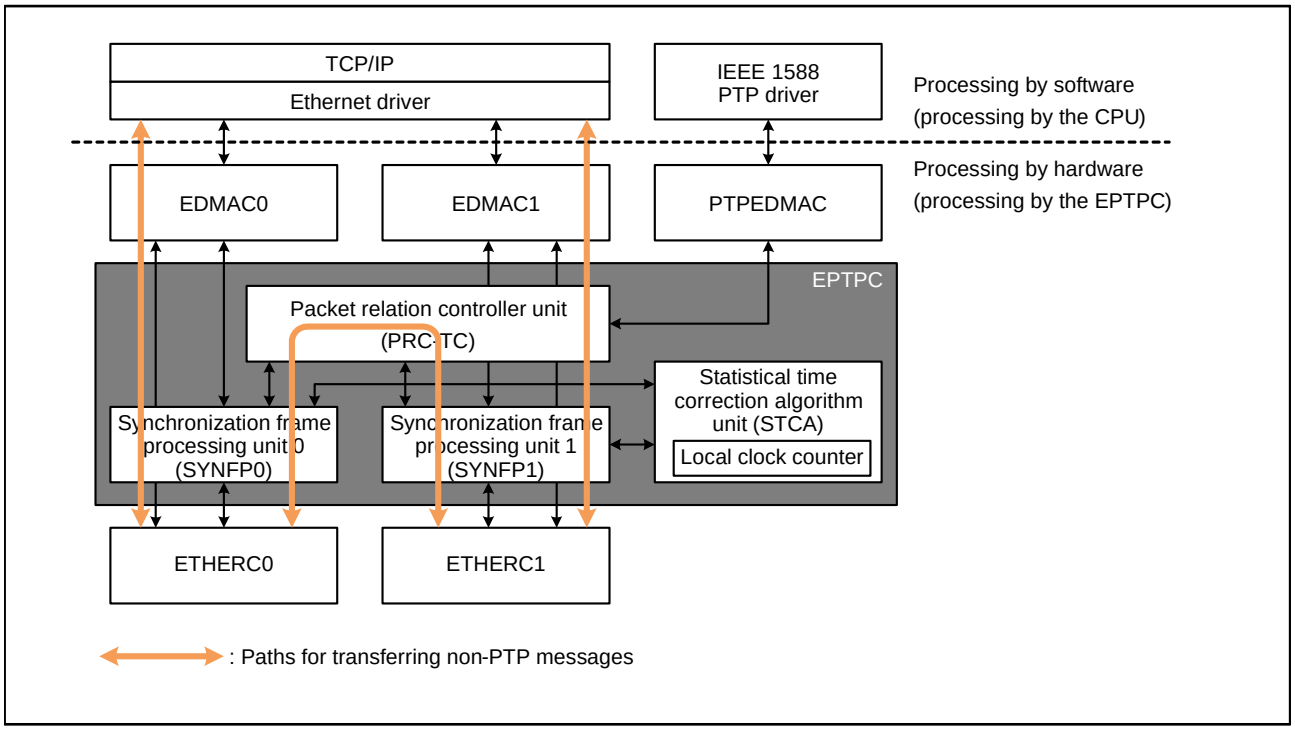


Figure 36.5 Paths for the Transmission and Reception and Relaying of Non-PTP Messages

36.3.3 Transmission and Reception and Relaying of PTP Messages

Hardware of the EPTPC automatically handles analysis of and the extraction of fields from received PTP messages, and generation and transmission of PTP messages. However, software must still handle the transmission of certain PTP messages. Table 36.10 shows the specifications for control over the transmission and reception of the various PTP messages.

Table 36.10 Control over the Transmission and Reception of PTP Messages

Message Type	Message	OC (Ordinary Clock)/BC (Boundary Clock)		TC (Transparent Clock)
		Master	Slave	
Event	Sync	Generation (automatic)	Reception (automatic)	Relaying (automatic)
	Delay_Req	Generation (automatic)	Reception (automatic)	Relaying (automatic)
	Pdelay_Req	Generation and reception (automatic)	Generation and reception (automatic)	Generation and reception (automatic)
	Pdelay_Resp	Generation and reception (automatic)	Generation and reception (automatic)	Generation and reception (automatic)
General	Announce	Generation (automatic)	Reception (software)	Reception (software)
	Follow_Up	—*1	Reception (automatic)	Relaying (automatic)
	Delay_Resp	Packet generation	Reception (automatic)	Relaying (automatic)
	Pdelay_Resp_Follow_Up	—*1	Reception (automatic)	Relaying (automatic)
	Management	Transmission and reception (software)		Transmission and reception (software) Relaying (automatic)
	Signaling	Transmission and reception (software)		Transmission and reception (software) Relaying (automatic)

Note 1. Control is not required as the clock in this case is a one-step clock.

36.3.4 Paths for the Transfer of PTP Messages

Paths for the transfer of PTP messages differ according to whether transfer requires processing by software or is automatically processed by hardware.

36.3.4.1 Paths for the Transfer of PTP Messages Requiring Processing by Software

Paths for the transfer of PTP messages where transfer requires processing by software are shown in Figure 36.6. The figure shows paths for message, clock-type, and process combinations for which “(software)” is indicated in Table 36.10.

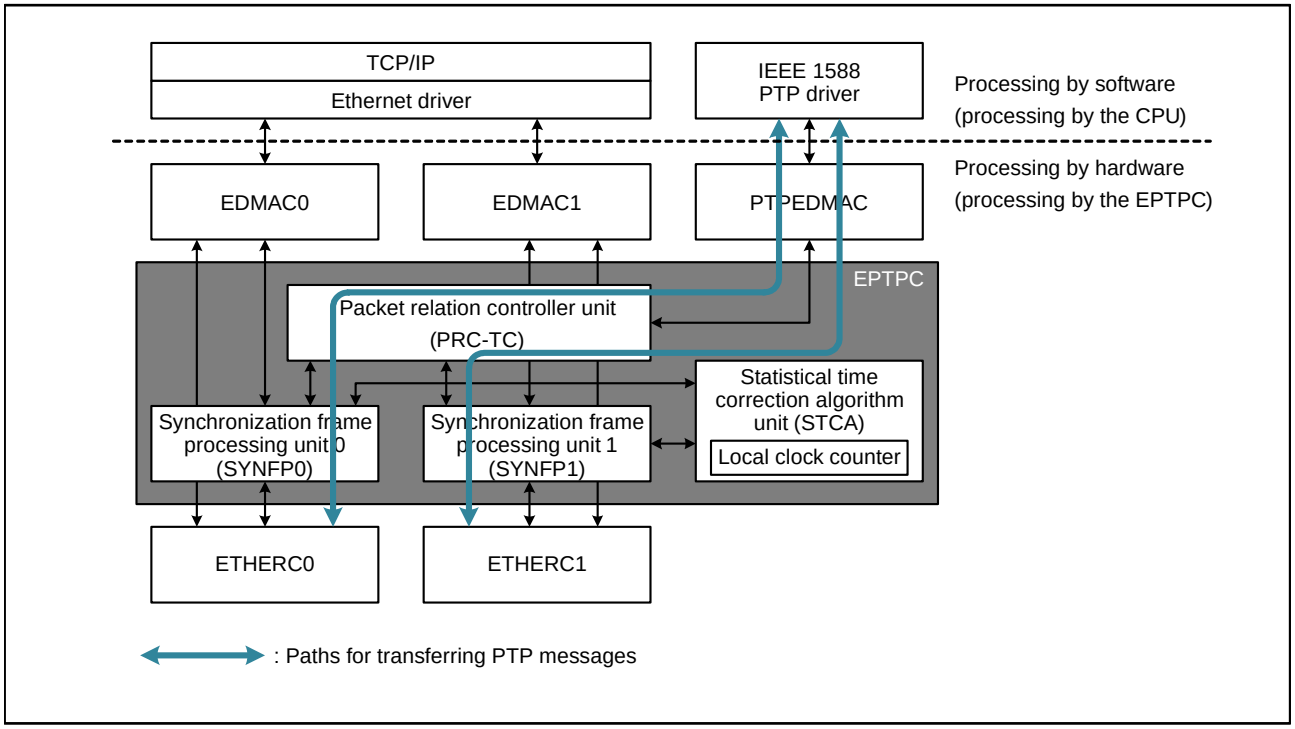


Figure 36.6 Paths for the Transfer of PTP Messages Requiring Processing by Software

36.3.4.2 Paths for the Transfer of PTP Messages Automatically Handled by Hardware

In the case of PTP messages for which hardware automatically handles the processing, the SYNFP modules handle transmission and reception, and the PRC-TC module handles relaying.

(1) Generation of and Response to PTP Messages by Hardware

Paths for transfer in the automatic generation of and response to PTP messages by the SYNFP modules are shown in Figure 36.7. The paths in the figure are used for operations “Generation (automatic)”, “Reception (automatic)”, and “Generation and reception (automatic)” indicated in Table 36.10.

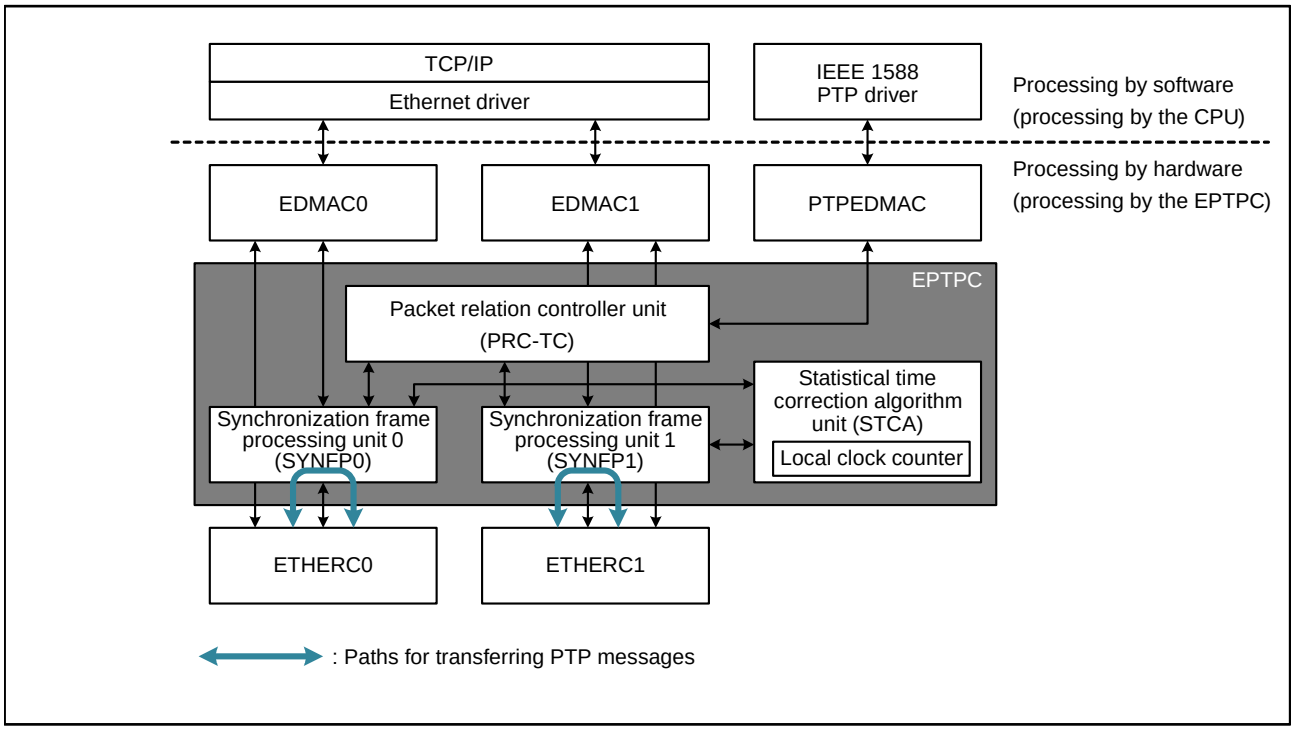


Figure 36.7 Paths for the Generation of and Response to PTP Messages by Hardware

(2) Relaying of PTP Messages by Hardware

Figure 36.8 shows paths for the relaying of PTP messages by the PRC-TC module. The paths in the figure are used for the “Relaying” operation indicated in Table 36.10.

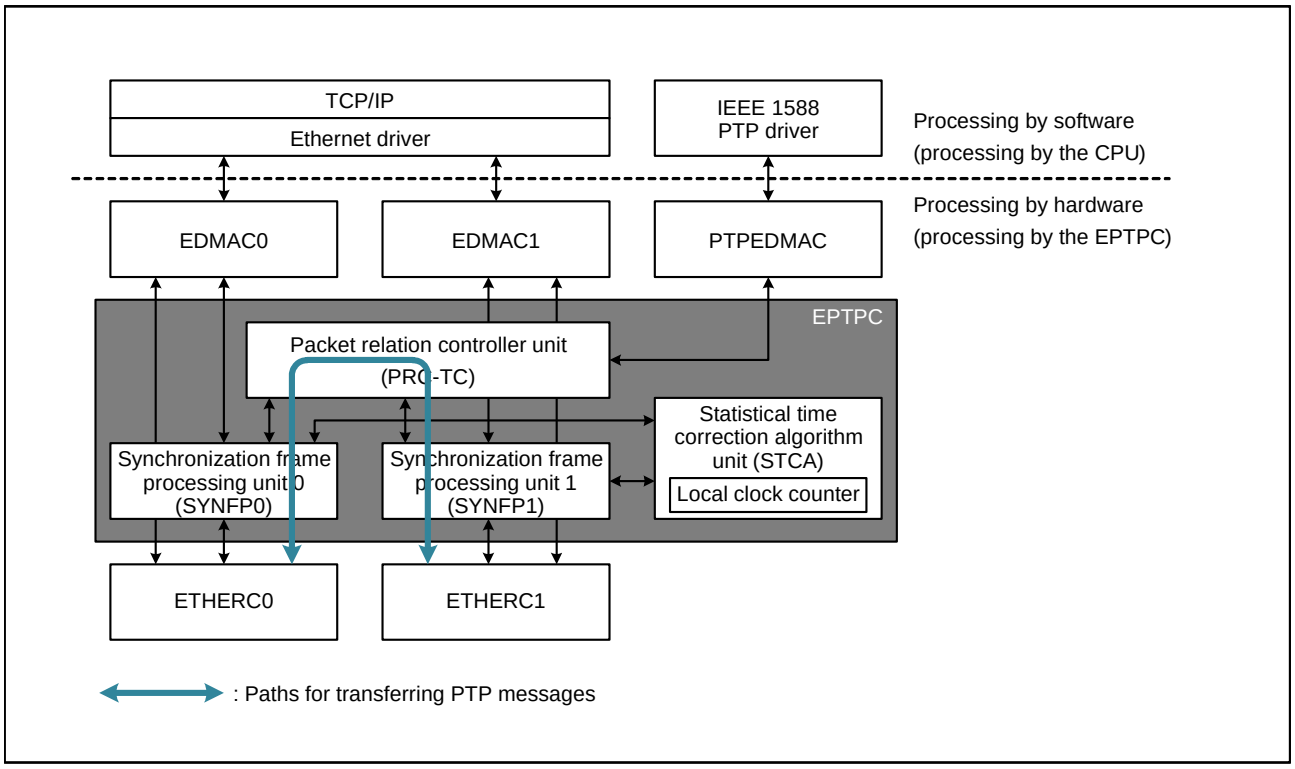


Figure 36.8 Paths for the Relaying of PTP Messages by Hardware

36.3.5 Clock Devices

The EPTPC can operate as the clock device defined in IEEE 1588.

36.3.5.1 End-to-End (E2E)

(1) Master

PTP messages are transmitted and received as listed below in operation as an end-to-end (E2E) master.

Table 36.11 List of PTP Message Processing (by an E2E Master)

Message Type	Message	Details of Processing
Event	Sync	Sync messages are transmitted at the fixed interval specified by the SYTLIR.SYNC[7:0] bits.
	Delay_Req	When this message is received, a Delay_Resp message is transmitted in response.
	Pdelay_Req	—
	Pdelay_Resp	—
General	Announce	Announce messages are transmitted at the fixed interval specified by the SYTLIR.ANCE[7:0] bits.
	Follow_Up	—
	Delay_Resp	This is transmitted as the response to a received Delay_Req messages.
	Pdelay_Resp_Follow_Up	—
	Management	Management messages are transmitted and received by software via the PTPEDMAC.
	Signaling	Signaling messages are transmitted and received by software via the PTPEDMAC.

(2) Slave

PTP messages are transmitted and received as listed below in operation as an E2E slave, and the calculated offsetFromMaster is used to correct the local time information.

Table 36.12 List of PTP Message Processing (by an E2E Slave)

Message Type	Message	Details of Processing
Event	Sync	The offsetFromMaster value is calculated when this message is received if twoStepFlag in flagField was FALSE (one-step clock).
	Delay_Req	Delay_Req messages are transmitted at random intervals from 0 to the time specified by the SYTLIR.DREQ[7:0] bits × 2.
	Pdelay_Req	—
	Pdelay_Resp	—
General	Announce	Announce messages are transmitted by software via the PTPEDMAC.
	Follow_Up	The offsetFromMaster value is calculated when this message is received if twoStepFlag in flagField of the most recently received Sync message was TRUE (two-step clock).
	Delay_Resp	The meanPathDelay value is calculated when this message is received.
	Pdelay_Resp_Follow_Up	—
	Management	Management messages are transmitted and received by software via the PTPEDMAC.
	Signaling	Signaling messages are transmitted and received by software via the PTPEDMAC.

36.3.5.2 Peer-to-Peer (P2P)

(1) Master

PTP messages are transmitted and received as listed below in operation as a peer-to-peer (P2P) master.

Table 36.13 List of PTP Message Processing (by a P2P Master)

Packet Type	Message	Details of Processing
Event	Sync	Timestamps for transmission are transmitted at the fixed interval specified by the SYTLIR.SYNC[7:0] bits.
	Delay_Req	—
	Pdelay_Req	Pdelay_Req messages are transmitted at the fixed interval specified by the SYTLIR.DREQ[7:0] bits. When this message is received, a Pdelay_Resp message is transmitted in response.
	Pdelay_Resp	This is transmitted as the response to a received Pdelay_Req messages. The meanPathDelay value is calculated when this message is received if twoStepFlag in flagField was FALSE (one-step clock).
General	Announce	Announce messages are transmitted at the fixed interval specified by the SYTLIR.ANCE[7:0] bits.
	Follow_Up	—
	Delay_Resp	—
	Pdelay_Resp_Follow_Up	The meanPathDelay value is calculated when this message is received if twoStepFlag in flagField of the most recently received Pdelay_Resp message was TRUE (two-step clock).
	Management	Management messages are transmitted by software via the PTPEDMAC.
	Signaling	Signaling messages are transmitted by software via the PTPEDMAC.

(2) Slave

PTP messages are transmitted and received as listed below in operation as a P2P slave, and the calculated offsetFromMaster is used to correct the local time information.

Table 36.14 List of PTP Message Processing (by a P2P Slave)

Packet Type	Message	Details of Processing
Event	Sync	The offsetFromMaster value is calculated when this message is received if twoStepFlag in flagField was FALSE (one-step clock).
	Delay_Req	—
	Pdelay_Req	Pdelay_Req messages are transmitted at the fixed interval specified by the SYTLIR.DREQ[7:0] bits. When this message is received, a Pdelay_Resp message is transmitted in response.
	Pdelay_Resp	This is transmitted as the response to a received Pdelay_Req messages. The meanPathDelay value is calculated when this message is received if twoStepFlag in flagField was FALSE (one-step clock).
General	Announce	Announce messages are transmitted by software via the PTPEDMAC.
	Follow_Up	The offsetFromMaster value is calculated when this message is received if twoStepFlag in flagField of the most recently received Sync message was TRUE (two-step clock).
	Delay_Resp	—
	Pdelay_Resp_Follow_Up	The meanPathDelay value is calculated when this message is received if twoStepFlag in flagField of the most recently received Pdelay_Resp message was TRUE (two-step clock).
	Management	Management messages are transmitted and received by software via the PTPEDMAC.
	Signaling	Signaling messages are transmitted and received by software via the PTPEDMAC.

36.3.5.3 Ordinary Clock (OC)

PTP messages are transmitted and received through one Ethernet port in operation as an ordinary clock.
An ordinary clock operates as the grand master clock or as a slave clock in the master-slave hierarchy.
For operation as an E2E master, E2E slave, P2P master, or P2P slave, see section 36.3.7, Operation as an E2E Master, section 36.3.8, Operation as an E2E Slave, section 36.3.10, Operation as a P2P Master, and section 36.3.11, Operation as a P2P Slave.

36.3.5.4 Boundary Clock (BC)

PTP messages are transmitted and received through both ports in operation as a boundary clock.
One port operates as a slave in synchronization with the root master clock and the other operates as the master that delivers time information synchronized with the master clock.
Both ports can also operate as masters.
For operation as an E2E master, E2E slave, P2P master, or P2P slave, see section 36.3.7, Operation as an E2E Master, section 36.3.8, Operation as an E2E Slave, section 36.3.10, Operation as a P2P Master, and section 36.3.11, Operation as a P2P Slave.

36.3.5.5 Transparent Clock (TC)

(1) E2E TC

In operation as an E2E transparent clock, received PTP-primary and PTP-pdelay messages are relayed.

Table 36.15 List of Packet Processing (by an E2E TC)

Message Type	Message	Details of Processing
Event	Sync	The packet residence time in the clock device is added when these messages are relayed.
	Delay_Req	
	Pdelay_Req	
	Pdelay_Resp	
General	Announce	These messages are relayed.
	Follow_Up	
	Delay_Resp	
	Pdelay_Resp_Follow_Up	
	Management	These messages are relayed. Messages are also transmitted and received by software via the PTPEDMAC.
	Signaling	

(2) P2P TC

In operation as a P2P transparent clock, PTP-primary messages other than Delay_Req and Delay_Resp are relayed.

Table 36.16 List of Packet Processing (by a P2P TC)

Message Type	Message	Details of Processing
Event	Sync	The packet residence time in the clock device is added when this message is relayed.
	Delay_Req	—
	Pdelay_Req	Pdelay_Req messages are transmitted at the fixed interval specified by the SYTLIR.DREQ[7:0] bits. When this message is received, a Pdelay_Resp message is transmitted in response.
	Pdelay_Resp	This is transmitted as the response to a received Pdelay_Req messages. The meanPathDelay value is calculated when this message is received if twoStepFlag in flagField was FALSE (one-step clock).
General	Announce	These messages are relayed.
	Follow_Up	
	Delay_Resp	—
	Pdelay_Resp_Follow_Up	The meanPathDelay value is calculated when this message is received if twoStepFlag in flagField of the most recently received Pdelay_Resp message was TRUE (two-step clock).
	Management	These messages are relayed. Messages are also transmitted and received by software via the PTPEDMAC.
	Signaling	

36.3.6 EPTPC Initialization

Transmitting and receiving PTP messages requires settings in various registers of the EPTPC.
Table 36.17 is a list of the registers for which settings are required. Set the registers corresponding to the Ethernet port used. Also set the registers listed in Table 36.18 if the UDP and IPv4 are used for the frame format of the PTP messages.

Table 36.17 List of the Registers for which Settings are Required

Register Name	Settings	Description
STCFR	Example: 0x00000002	The value for 50 MHz is given as an example. Three other settings are also available.
PRMACRU0, PRMACRL0	As desired	This setting is not required If Ethernet port 0 is not used.
PRMACRU1, PRMACRL1	As desired	This setting is not required If Ethernet port 1 is not used.
SYCONFR	Example: 0x00000002	The setting differs with the type of PTP clock operation.
SYMACRU, SYMACRL	As desired	
SYSPVRR	0x0000002	transportSpecific and version fields
SYDOMR	As desired	
SYCIDRU, SYCIDRL	As desired	Set the same value for Ethernet ports 0 and 1.
SYPNUMR	0x00000001 or 0x00000002	If the PTP clock operates as an OC, the setting is 0x00000001. If it operates as a BC or TC, set 0x00000001 for one port and 0x00000002 for the other.
PPMACRU, PPMACRL	01:1B:19:00:00:00	MAC address for PTP-primary messages
PDMACRU, PDMACRL	01:80:C2:00:00:0E	MAC address for PTP-pdelay messages
DASYMRU, DASYMRL	0x00000000	
TSLATR	As desired	Depends on the link transfer rate and STCA clock frequency.
SYFORMR	As desired	Two settings are available.
PETYPER	0x000088F7	Ethertype for PTP messages

Table 36.18 List of the Registers for which Settings are Required (Additional Settings Required when UDP or IPv4 is to be Used)

Register Name	Settings	Description
SYIPADDRR	As desired	Local IP address
PRIPR	0xE0000181	IP address for PTP-primary messages
PETOSR	As desired	Set the highest allowable traffic class selector codepoint as the value for the differentiated service (DS) field.
PGTOSR	As desired	
PPTTLR	As desired	TTL field value for PTP-primary messages
PEUDPR	0x0000013F	UDP port number for event messages
PGUDPR	0x00000140	UDP port number for general messages
PDIPR	0xE000006B	IP address for PTP-pdelay messages
PDTTLR	0x00000001	TTL field value for PTP-pdelay messages

In operation as an OC or BC, set registers as shown below in order to transfer received Announce, Management, and Signaling messages to the PTPEDMAC.

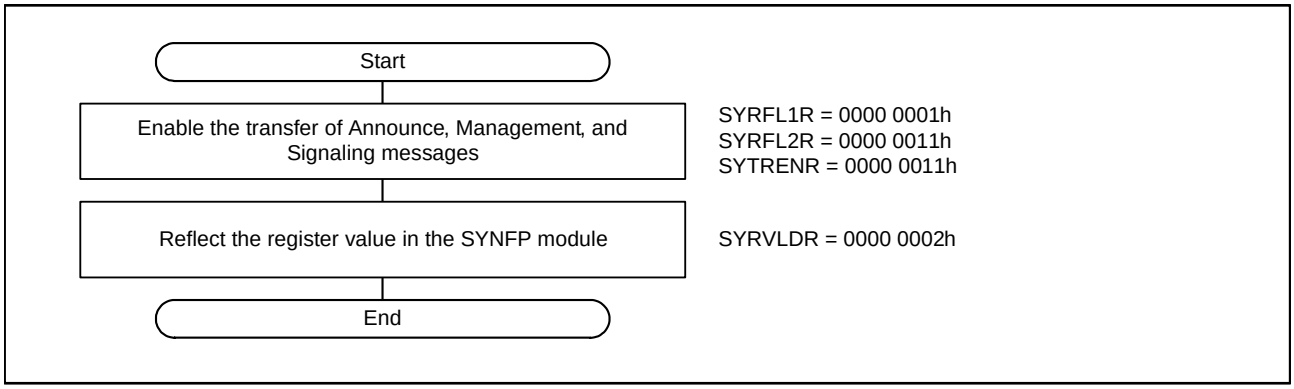


Figure 36.9 Common Settings for PTP Devices

36.3.7 Operation as an E2E Master

36.3.7.1 Preparatory Setting

Table 36.19 lists the registers for use in operation as an E2E master.
If the EPTPC operates as an OC or as a BC with both ports serving as masters, set the initial value of the time information in advance. Refer to section 36.2.18, Local Clock Counter Initial Value Registers (LCIVRU, LCIVRM, LCIVRL) for the initial value of the time information. Reflecting the values set in the registers requires setting the SYRVLDR.STUP or ANUP bit to 1.

Table 36.19 Registers for Use in Operation as an E2E Master

Register Name	SYRVLDR Register Bits Used for Loading Direction	Settings	Description
SYNFR	STUP	0x00000000	flagField for Sync messages
SYTLIR	STUP ANUP	Example: 0x00000001	Delay_Resp: 1 s Sync: 1 s Announce: 2 s
ANFR	ANUP	0x00000000	flagField for Announce messages
GMPR	ANUP	As desired	
GMCQR	ANUP	As desired	
GMIDR	ANUP	As desired	
CUOTSR	ANUP	As desired	timeSource: Internal Oscillator
SRR	ANUP	As desired	If the EPTPC operates as an OC or as a BC with both ports serving as masters, set this register to 0x00000000. If the EPTPC operates as a BC in the combination of a slave and master, set this register to the StepsRemoved field value of Announce messages received by the slave plus one.
SYRFL1R	STUP	0x00004001	This enables the processing of Delay_Req messages by the SYNFP module.
SYRFL2R	STUP	0x00000011	This enables the transfer of Signaling and Management messages to the PTPEDMAC.
SYTRENR	STUP	0x00000011	This enables the transmission of Sync and Announce messages.

36.3.7.2 Procedure for Starting Operations

Figure 36.10 shows the procedure for settings to start operation as an E2E master.

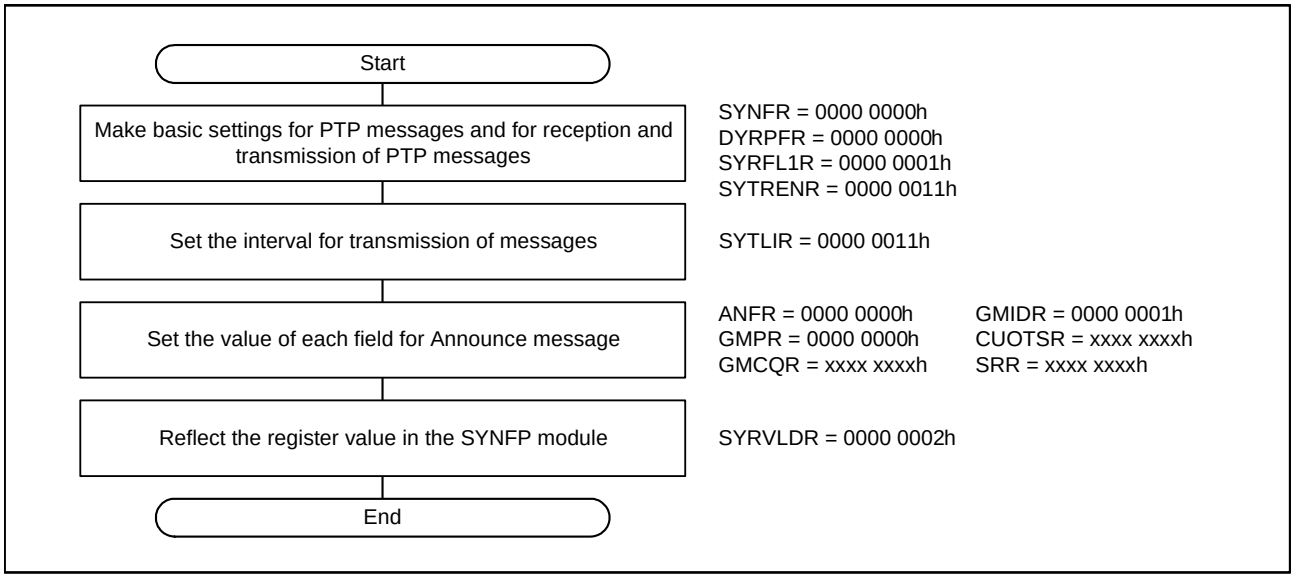


Figure 36.10 Procedure for Starting Operation as an E2E Master

36.3.7.3 Procedure for Changing the Settings

Increases in the frequency of receiving Delay_Req messages due to network conditions may lead to an overflow of the FIFO buffer which is receiving the Delay_Req messages. In such cases, change the value of the logMessageInterval field of Delay_Resp messages so that the slave sending the Delay_Req messages lengthens the interval between the messages. Figure 36.11 shows the procedure for changing the value of the logMessageInterval field.

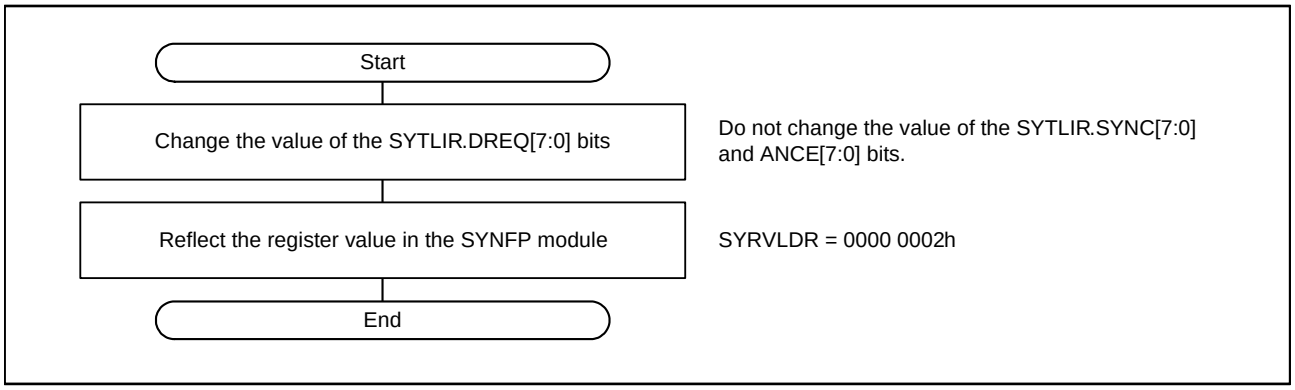


Figure 36.11 Procedure for Changing the Value of the logMessageInterval Field for Delay_Resp Messages

36.3.7.4 Procedure for Stopping Operations

Figure 36.12 shows the procedure for stopping operation as an E2E master. To confirm that the operation as the E2E master is completely stopped, read the SYSR.GENDN flag and RESDN flag to check that generating messages and sending responses are completely stopped.

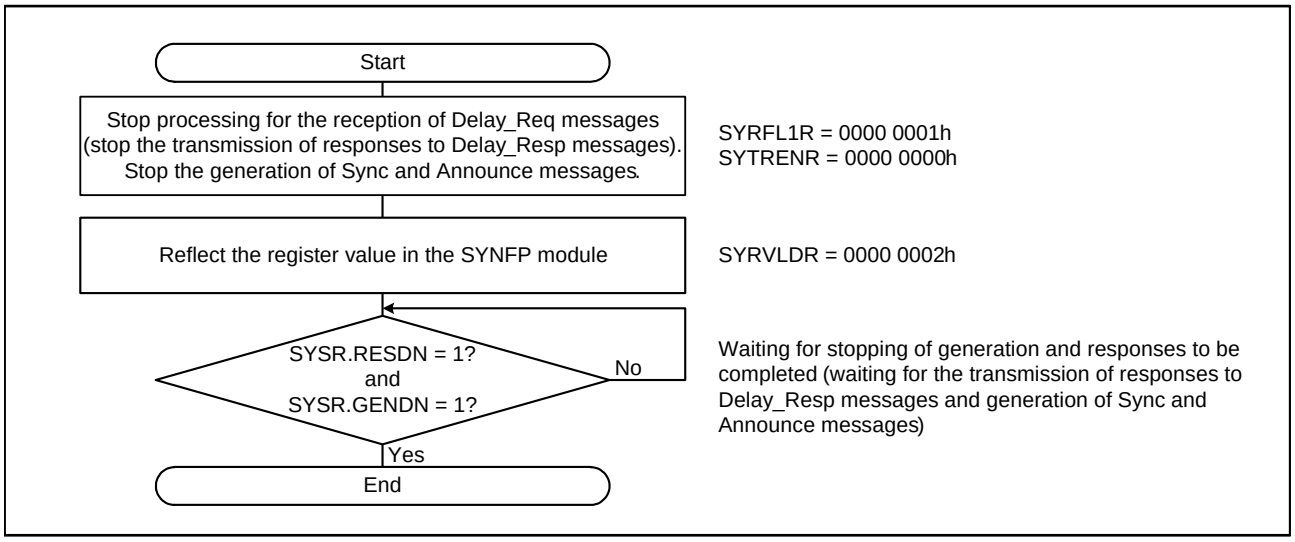


Figure 36.12 Procedure for Stopping Operation as an E2E Master

36.3.8 Operation as an E2E Slave

36.3.8.1 Preparatory Setting

Table 36.20 lists the registers for use in operation as an E2E slave.
Reflecting the value set in the register in SYNFP operations requires setting the SYRVLDR.STUP, ANUP, or BMUP bit to 1.

Table 36.20 Registers for Use in Operation as an E2E Slave

Register Name	SYRVLDR Register Bits Used for Loading Direction	Settings	Description
MTCID	BMUP	As desired	This is the clockIdentity value of the master clock that provides synchronization.
MTPID	BMUP	As desired	This is the portNumber value of the master clock that provides synchronization.
SYTLIR	ANUP BMUP	0x00000000	Delay_Resp: 1 s*1
RSTOUTR	STUP	As desired	
SYNTOR	—	As desired	
SYRFL1R	STUP	0x00040441	This enables the reception of Delay_Resp, Follow_Up, and Sync messages and the transfer of Announce messages to the PTPEDMAC.
SYRFL2R	STUP	0x00000011	This enables the transfer of Signaling and Management messages to the PTPEDMAC.
SYTRENR	STUP	0x00000100	This enables the generation of Delay_Req messages.

Note 1. In the reception of Delay_Resp messages by an E2E slave, the SYTLIR.DREQ[7:0] bits must be adjusted if the value of the SYRLIR.DRESP[7:0] flags is to be altered. The SYTLIR.DREQ[7:0] bits specify a value in the range from -7 to 6. Set the SYTLIR.DREQ[7:0] bits to -7 if the value indicated by the SYRLIR.DRESP[7:0] flags is less than or equal to -8 and to 6 if the value indicated by the SYRLIR.DRESP[7:0] flags is greater than or equal to 7.

36.3.8.2 Procedure for Starting Operations

Figure 36.13 shows the procedure for settings to start operation as an E2E slave.

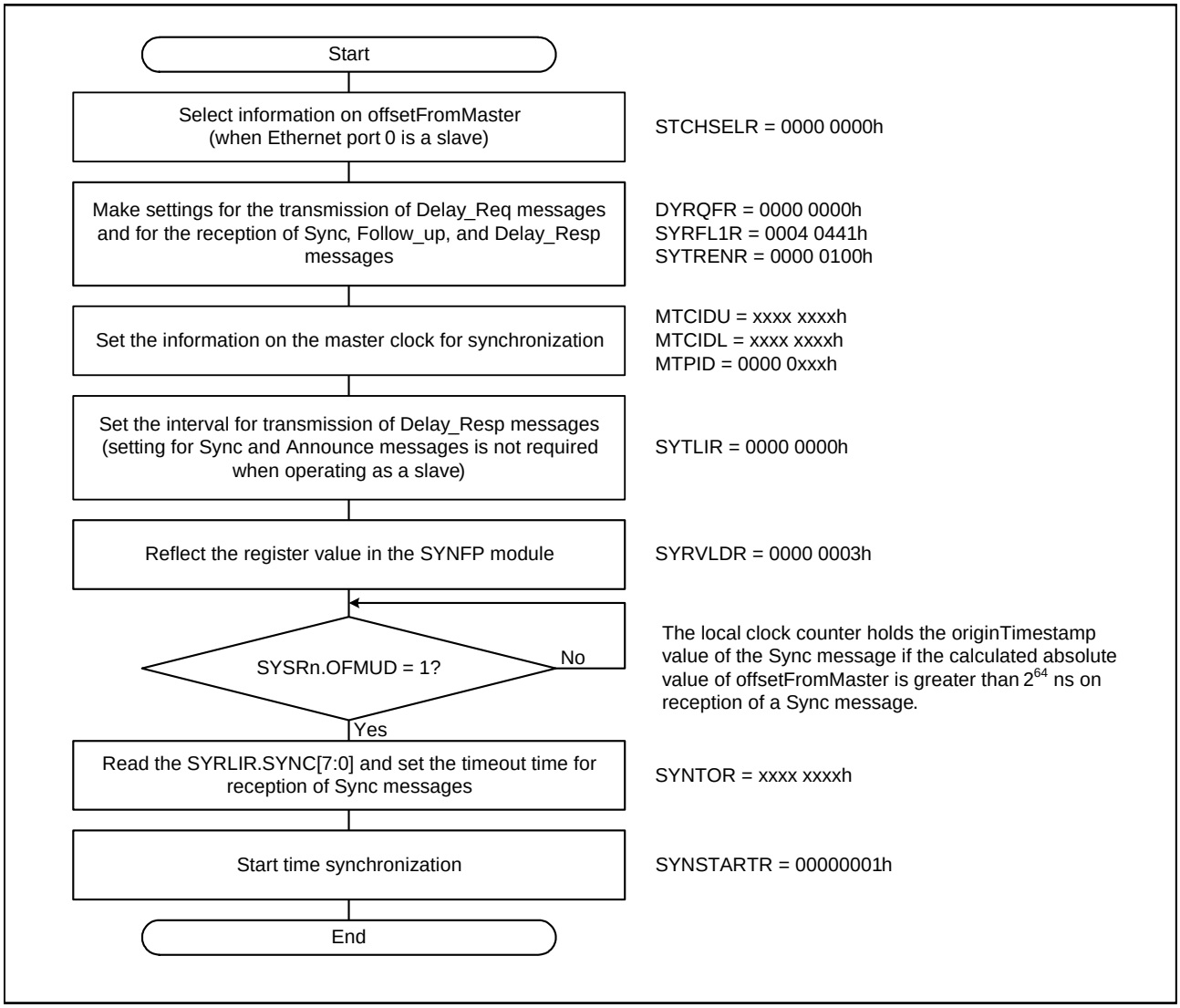


Figure 36.13 Procedure for Starting Operation as an E2E Slave

36.3.8.3 Procedure for Changing the Settings

IEEE 1588 stipulates that the average interval for the transmission of Delay_Req messages must be adjusted in response to changes in the value of the logMessageInterval field of received Delay_Resp messages. The EPTPC sets the SYSR.INTCHG flag to 1 if the logMessageInterval value of a received message differs from that of the previous message.

When this happens, set the SYTLIR.DREQ[7:0] bits to the value of the SYRLIR.DRESP[7:0] bits.

The SYTLIR.DREQ[7:0] bits specify a value in the range from -7 to 6. Set the SYTLIR.DREQ[7:0] bits to -7 if the value indicated by the SYRLIR.DRESP[7:0] flags is less than or equal to -8 and to 6 if the value indicated by the SYRLIR.DRESP[7:0] bits is greater than or equal to 7.

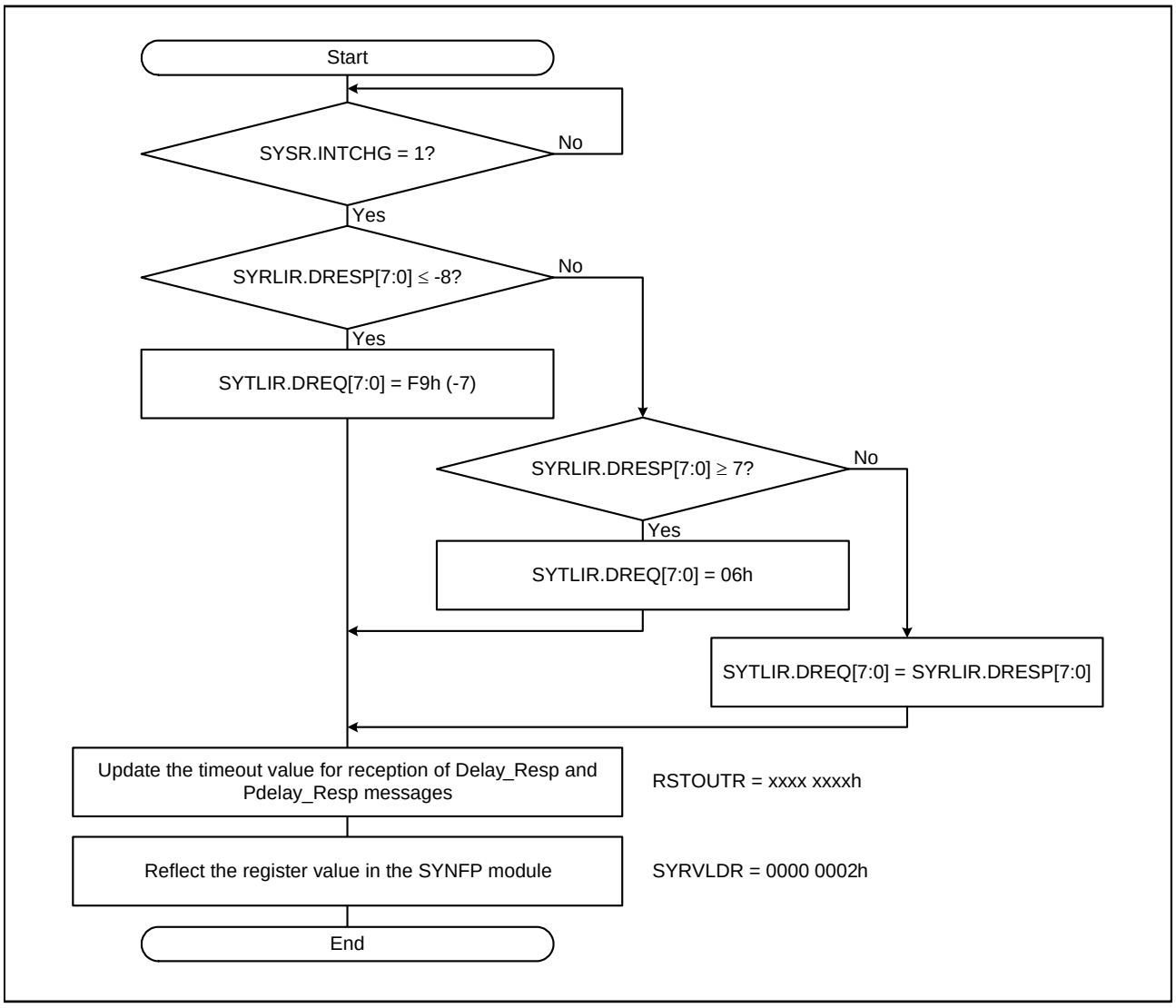


Figure 36.14 Procedure for Changing the Interval for Transmission of Delay_Req Messages

36.3.8.4 Procedure for Stopping Operations

Figure 36.15 shows the procedure for stopping operation as an E2E slave. To confirm that operation as an E2E slave is completely stopped, read the SYSR.GENDN flag to check that generation is completely stopped.

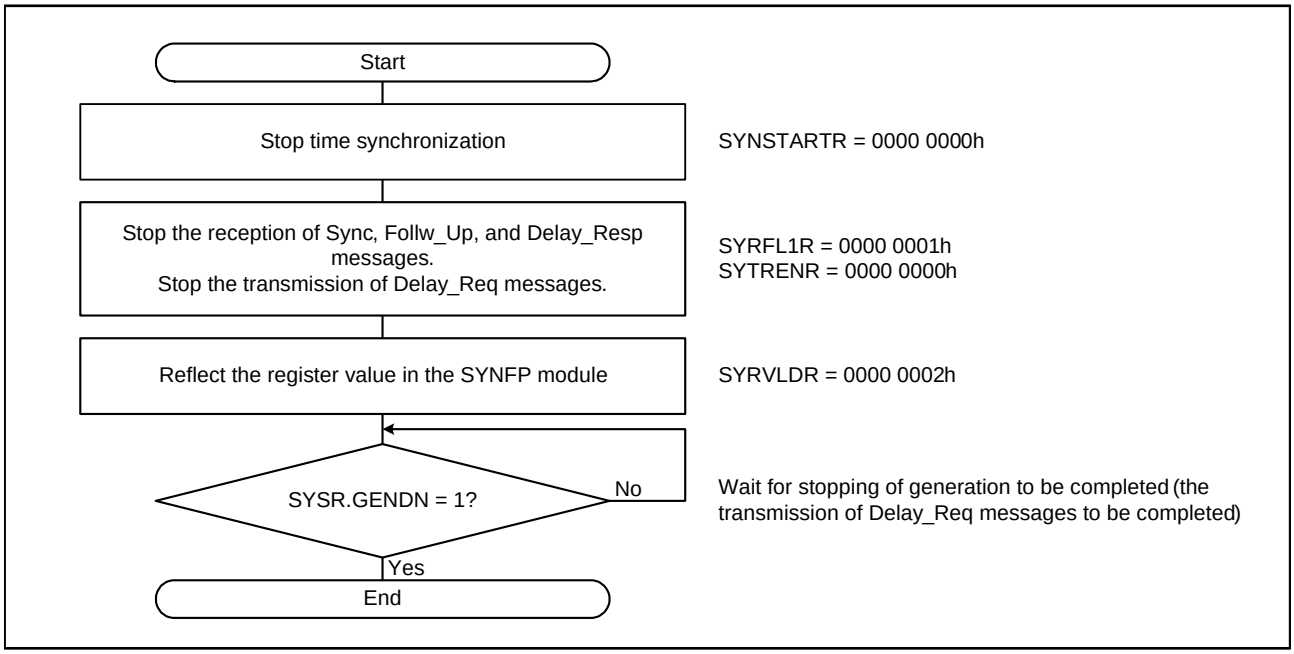


Figure 36.15 Procedure for Stopping Operation as an E2E Slave

36.3.9 P2P Operation (Common to Master and Slave)

If the EPTPC is to be operated as a P2P, the SYNFP module handles the processing of PTP-pdelay messages regardless of whether operation is to be as a master or slave. The interval for Pdelay_Req transmission and the parameters for monitoring of Pdelay_Resp messages must be set at the same time. Registers for use in operation as a P2P are listed in Table 36.21.

Table 36.21 Registers for Use in Operation as a P2P

Register Name	SYRVLDR Register Bits Used for Loading Direction	Settings	Description
MTCID	BMUP	As desired	This is the clockIdentity value of the synchronized master clock.
MTPID	BMUP	As desired	This is the portNumber value of the synchronized master clock.
SYTLIR	ANUP STUP	0x00000000	Announce: — Sync: — Pdelay_Req: 1 s
RSTOUTR	STUP	As desired	
SYRFL1R	STUP	0x44400001	This enables the reception of Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages and the transfer of Announce messages to the PTPEDMAC.
SYRFL2R	STUP	0x00000011	This enables the transfer of Signaling and Management messages to the PTPEDMAC.
SYTRENR	STUP	0x00001000	This enables the generation of Pdelay_Req messages.

36.3.9.1 Procedure for Starting Operations

Figure 36.16 shows the procedure for starting operation as a P2P (sending and receiving PTP-pdelay messages).

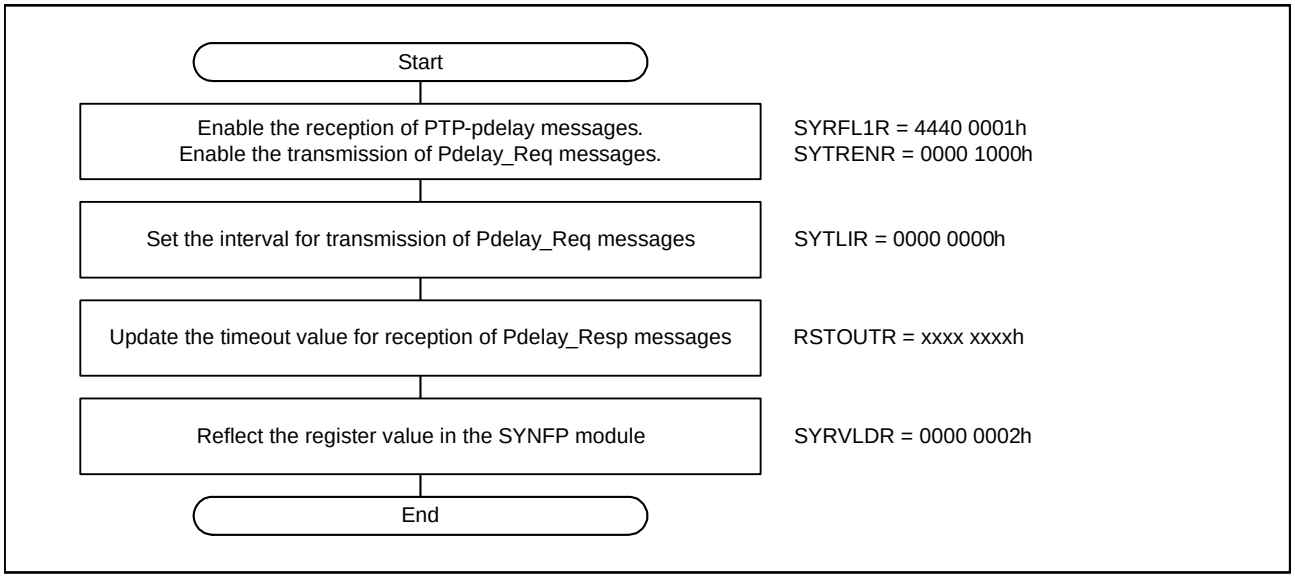


Figure 36.16 Procedure for Starting Operation as a P2P

36.3.9.2 Procedure for Stopping Operations

Figure 36.17 shows the procedure for stopping operation as a P2P (sending and receiving PTP-pdelay messages).

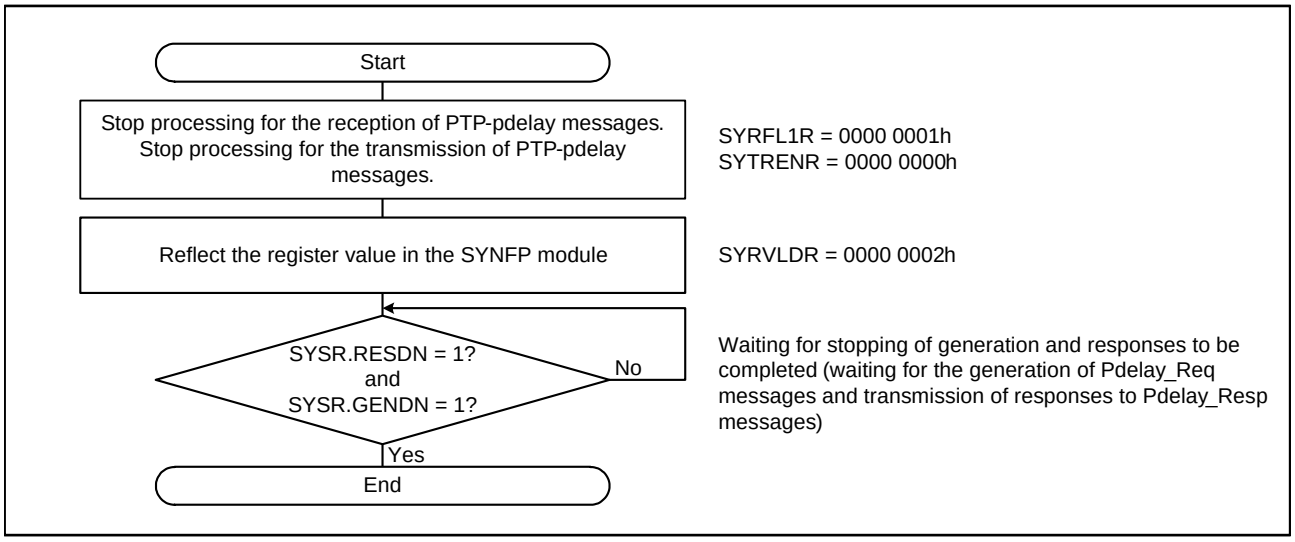


Figure 36.17 Procedure for Stopping Operation as a P2P

36.3.10 Operation as a P2P Master

When the EPTPC operates as OC or BC that uses both ports as the master, set the initial value of time information in advance as required. Refer to section 36.2.18, Local Clock Counter Initial Value Registers (LCIVRU, LCIVRM, LCIVRL) for the initial value of time information. Registers for use in operation as a P2P master are listed in Table 36.22.

Table 36.22 Registers for Use in Operation as a P2P Master

Register Name	SYRVLDR Register Bits Used for Loading Direction	Settings	Description
SYCONFR	—	0x00000028	
ANFR	ANUP	0x00000000	flagField for Announce messages
SYNFR	STUP	0x00000000	flagField for Sync messages
SYTLIR	ANUP STUP	0x00000001	Announce: 2 s Sync: 1 s Pdelay_Req: 1 s
GMPR	ANUP	As desired	Grandmaster Priority1 and Priority2
GMCQR	ANUP	As desired	Grandmaster Quality
GMIDR	ANUP	As desired	Grandmaster Identity
CUOTSR	ANUP	As desired	currentUtcOffset, timeSource
SRR	ANUP	As desired	StepsRemoved
RSTOUTR	STUP	As desired	
SYRFL1R	STUP	0x44400000	This enables the reception of Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages.
SYRFL2R	STUP	0x00000011	This enables the transfer of Signaling and Management messages to the PTPEDMAC.
SYTRENR	STUP	0x00001011	This enables the transmission of Pdelay_Req, Sync, and Announce messages.

36.3.10.1 Procedure for Starting Operations

When transmission of Sync and Announce messages is started during operation as a P2P (sending and receiving PTP-pdelay messages), the EPTPC operates as a P2P master. Figure 36.18 shows the procedure for starting operation as a P2P master.

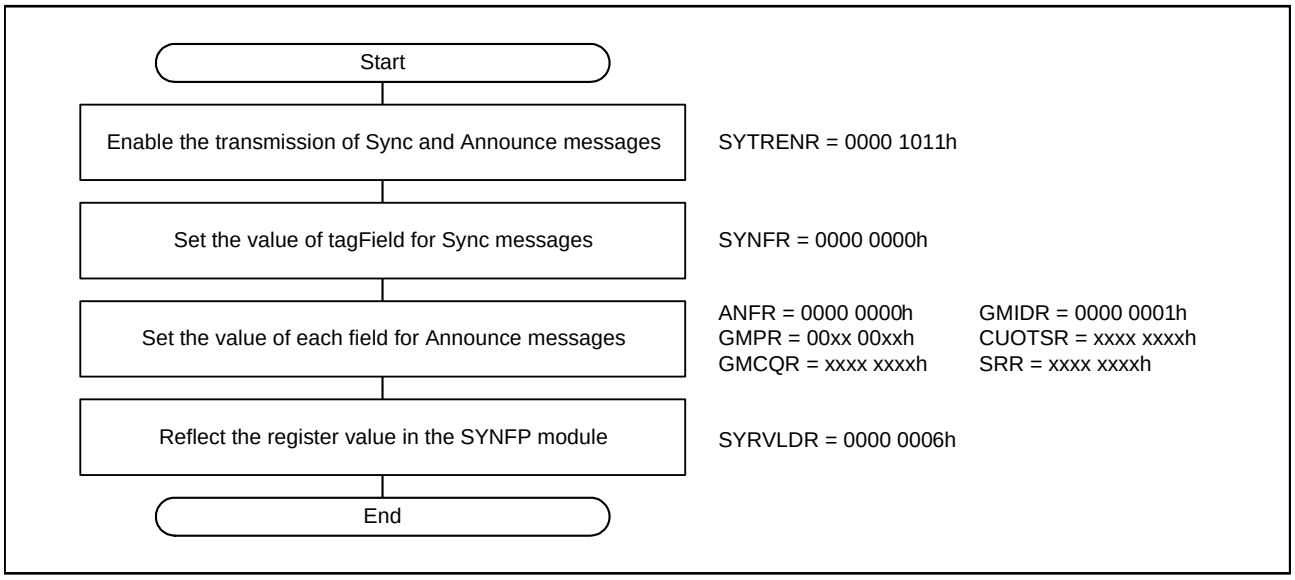


Figure 36.18 Procedure for Starting Operation as a P2P Master

36.3.10.2 Procedure for Stopping Operations

Figure 36.19 shows the procedure for stopping the transmission of Sync and Announce messages to stop operation as a P2P master.

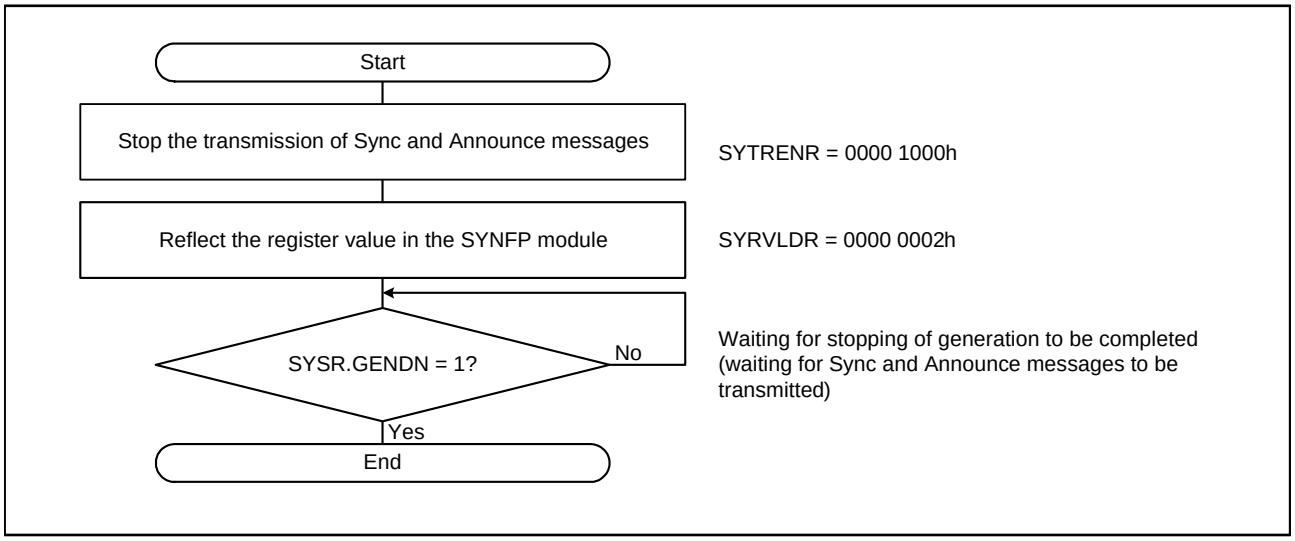


Figure 36.19 Procedure for Stopping Operation as a P2P Master

36.3.11 Operation as a P2P Slave

Setting a SYNFP module to receive Sync messages and Follow_Up messages during P2P operation obtains operation as a P2P slave. Information on the master clock for synchronization must be specified.
Table 36.23 lists the registers for use in operation as a P2P slave.

Table 36.23 Registers for Use in Operation as a P2P Slave

Register Name	SYRVLDR Register Bits Used for Loading Direction	Settings	Description
MTCID	BMUP	As desired	This is the clockIdentity value of the synchronized master clock.
MTPID	BMUP	As desired	This is the portNumber value of the synchronized master clock.
RSTOUTR	STUP	As desired	
SYRFL1R	STUP	0x44400441	This enables the reception of Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up, Follow_Up, and Sync messages and the transfer of Announce messages to the PTPEDMAC.

36.3.11.1 Procedure for Starting Operations

Figure 36.20 shows the procedure for making the additional settings for shifting to operation as a slave during operation as a P2P (sending and receiving PTP-pdelay messages).

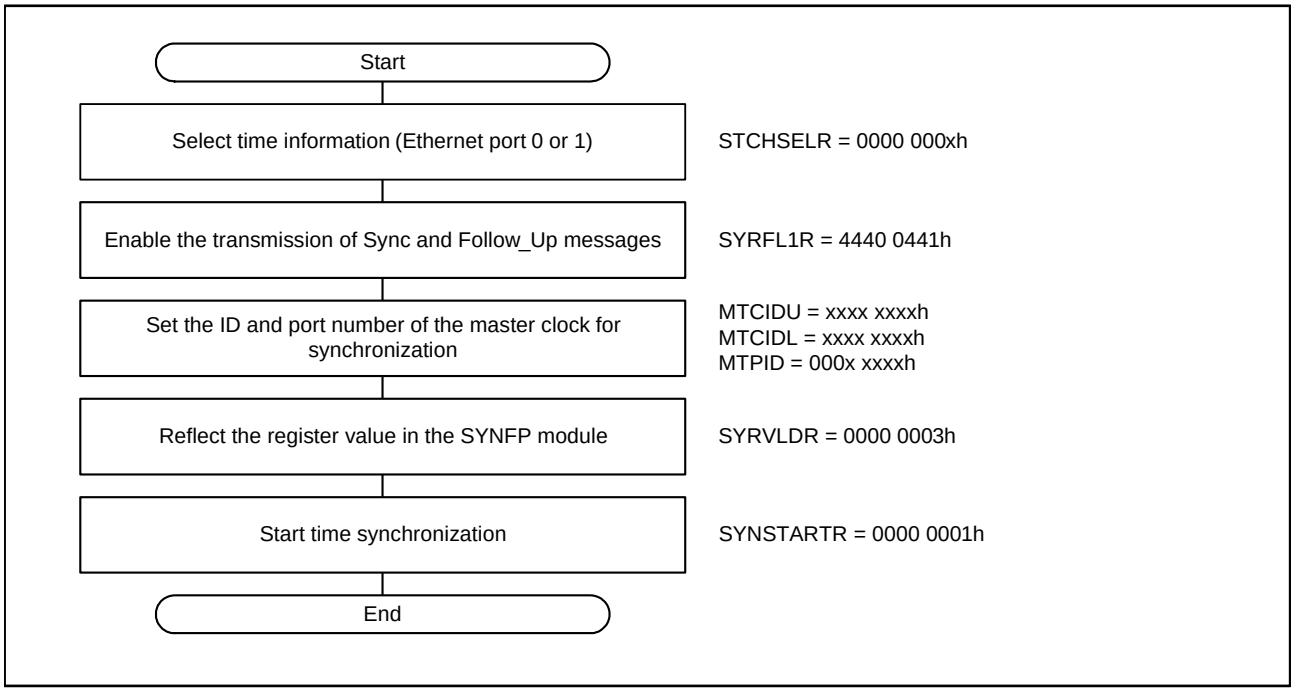


Figure 36.20 Procedure for Starting Operation as a P2P Slave

36.3.11.2 Procedure for Stopping Operations

Figure 36.21 shows the procedure for stopping the reception of Sync and Follow_Up messages to stop operation as a P2P slave.

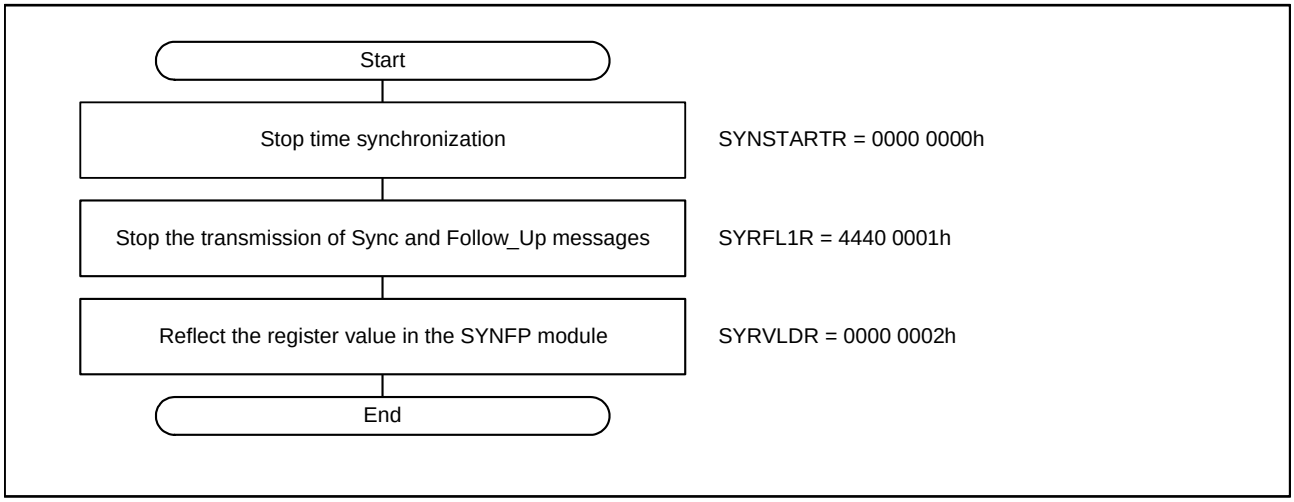


Figure 36.21 Procedure for Stopping Operation as a P2P Slave

36.3.12 Operation as an E2E TC

36.3.12.1 Preparatory Setting

Table 36.24 lists the registers for use in operation as an E2E TC.
Make the settings for both the SYNFP0 and SYNFP1 modules.

Table 36.24 Registers for Use in Operation as an E2E TC

Register Name	SYRVLDR Register Bits Used for Loading Direction	Settings	Description
SYCONFR	—	Example: 0x00000028	Set the TCMOD bit to 0 as a preparatory setting for the PTP device.
SYRFL1R	STUP	0x22222222	This enables the relaying of messages by the PRC-TC.
SYRFL2R	STUP	0x20000033	This enables the transfer of Signaling and Management messages to the PTPEDMAC and the relaying of these messages by the PRC-TC.
SYTRENR	STUP	0x00000000	No generation of messages

36.3.12.2 Procedure for Starting Operations

Figure 36.22 shows the procedure for starting operation as an E2E TC.

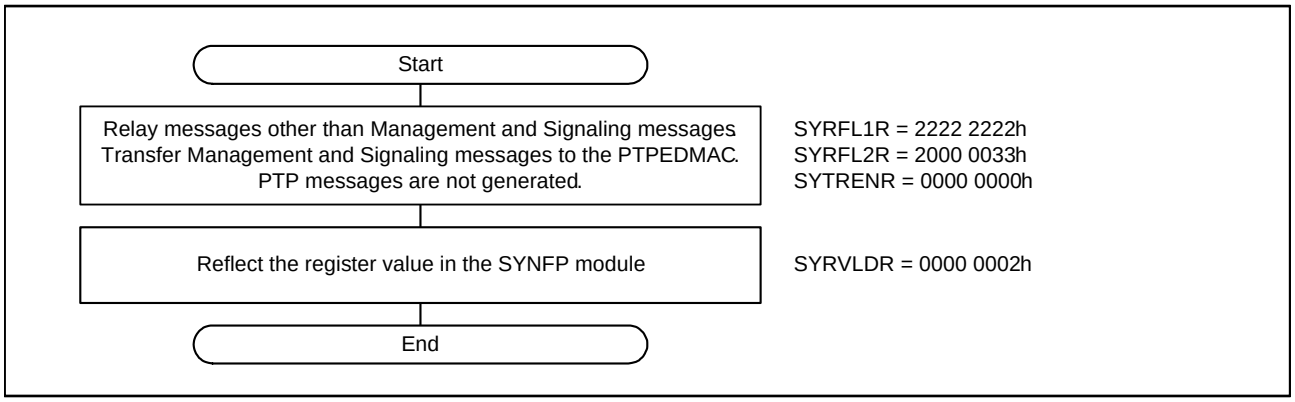


Figure 36.22 Procedure for Starting Operation as an E2E TC

36.3.13 Operation as a P2P TC

Table 36.25 lists the registers for use in operation a P2P TC.
Make the settings for both the SYNFP0 and SYNFP1 modules.

Table 36.25 Registers for Use in Operation as a P2P TC

Register Name	SYRVLDR Register Bits Used for Loading Direction	Settings	Description
SYCONFR	—	Example: 0x00100028	Set the TCMOD bit to 1 as a preparatory setting for the PTP device.
RSTOUTR	STUP	As desired	
SYTLIR	STUP	Example: 0x00000000	Pdelay_Req: 1 s
SYRFL1R	STUP	0x44400222	This enables the discarding of Delay_Req and Delay_Resp messages and the relaying of Sync, Follow_Up, and Announce messages by the PRC-TC.
SYRFL2R	STUP	0x20000033	This enables the transfer of Signaling and Management messages to the PTPEDMAC and the relaying of these messages by the PRC-TC
SYTRENR	STUP	0x00001000	This enables the generation of Pdelay_Req messages.

36.3.13.1 Procedure for Starting Operations

Figure 36.23 shows the procedure for starting operation as a P2P TC.

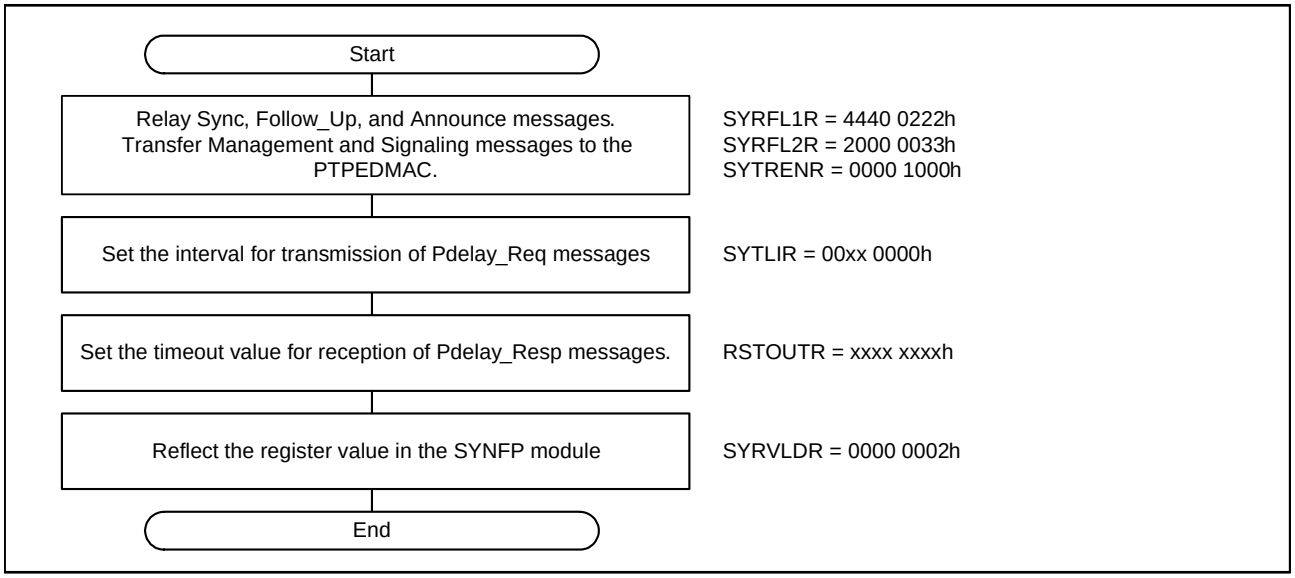


Figure 36.23 Procedure for Starting Operation as a P2P TC

36.3.14 Monitoring of Received Messages

36.3.14.1 Reception of Announce Messages

The EPTPC does not detect timeouts in the reception of Announce messages. To detect timeouts in the reception of these messages, monitor the reception of Announce messages by software.

36.3.14.2 Reception of Sync Messages

The STSR.SYNTOUT flag becomes 1 when a timeout in the reception of a Sync message occurs during the correction of time synchronization.
The SYSR.OFMUD flag becomes 1 when a Sync message is received, regardless of whether time synchronization is being corrected. Accordingly, the reception of Sync messages is detectable by reference to the SYSR.OFMUD flag even if the correction of time synchronization stops due to a timeout in the reception of a Sync message.

36.3.14.3 Reception of Delay_Resp and Pdelay_Resp Messages

The SYSR.DRPTO flag becomes 1 when a timeout in the reception of a Delay_Resp message occurs after the transmission of a Delay_Req message during operation as an E2E slave, or when a timeout in the reception of a Pdelay_Resp message occurs after the transmission of a Pdelay_Req message during operation as a P2P.
The SYSR.MPDUD flag becomes 1 when a Delay_Resp or Pdelay_Resp message is received, so the reception of these messages is still detectable in the case of a timeout in reception.

36.3.15 Correcting Time Synchronization

A slave detects differences in the clock gradient relative to the master clock. The offsetFromMaster values calculated by using the standard IEEE 1588 algorithm are used to calculate the clock gradient, so the result includes elements of network fluctuation besides frequency differences. The EPTPC has a “worst 10” function to eliminate fluctuations due to network load, etc. With these functions, the time is corrected from the calculated gradient difference values and results of correction are obtained as shown in Figure 36.25.

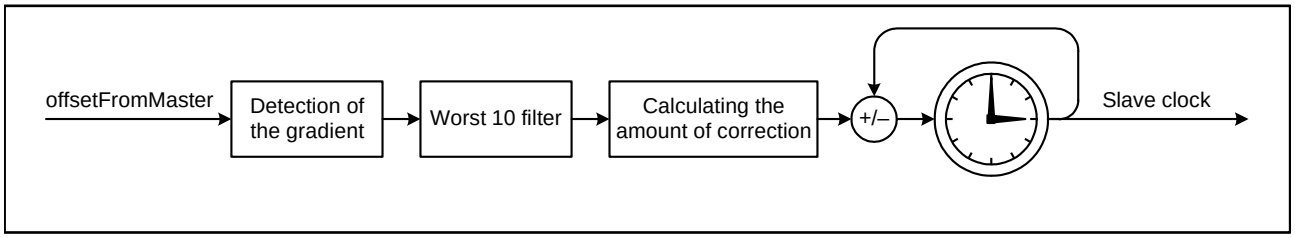


Figure 36.24 Configuration of the Time Correction Circuit

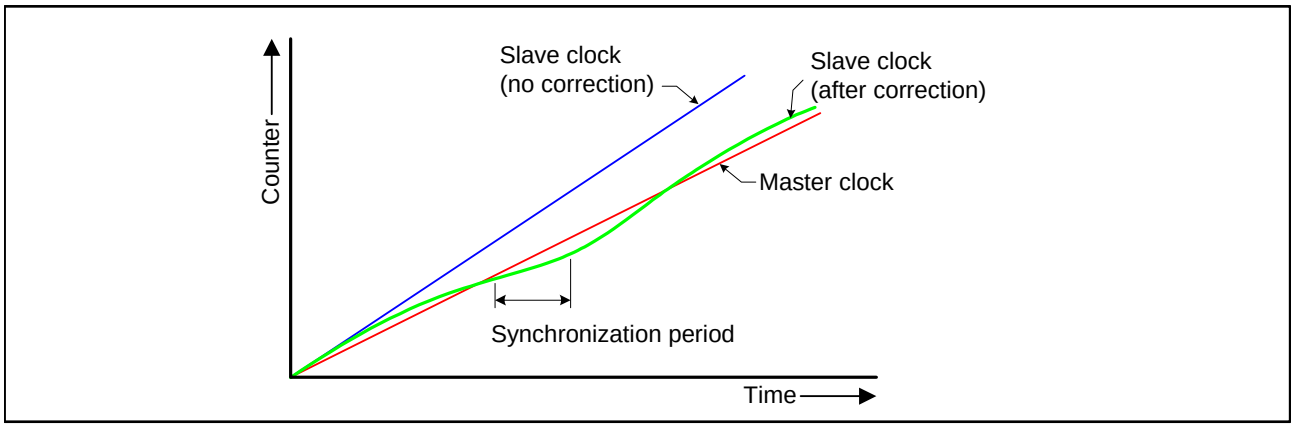


Figure 36.25 Overview of Time Correction

36.3.15.1 Judging Synchronization and Loss of Synchronization

The absolute value of offsetFromMaster reaching or exceeding the value specified in the SYNTDARU or SYNTDARL register is considered a loss of synchronization. The absolute value of offsetFromMaster being less than the absolute values of the synchronization detection threshold registers (SYNTDBRU and SYNTDBRL) is considered synchronization.

The values of the STSR.SYNCOUT and SYNC flags become 1 when synchronization is lost and obtained, respectively. Hysteresis can be obtained by setting the respective threshold registers to appropriate different values. Furthermore, the STMR.DVTH[3:0] and SYTH[3:0] bits can be used to set the consecutive number of times detection must occur for the judgment of loss of synchronization and of synchronization.

Set registers SYNTDARU and SYNTDARL to low values and the number of times detection is required to judge loss of synchronization to one in systems where control must be aborted if synchronization is lost due to fluctuations in network conditions. Set registers SYNTDARU and SYNTDARL and the number of times detection is required to large values in systems where the above condition does not apply.

Figure 36.26 shows an example of a situation where synchronization is being lost and regained. In this example, the number of consecutive times detection is required is three for both synchronization and loss of synchronization.

Note: The setting of the STSR.SYNCOUT flag is 1 when time synchronization starts, even though the condition for judging loss of synchronization has not been satisfied at this stage. For this reason, detection of loss of synchronization must be ignored immediately after time synchronization starts.

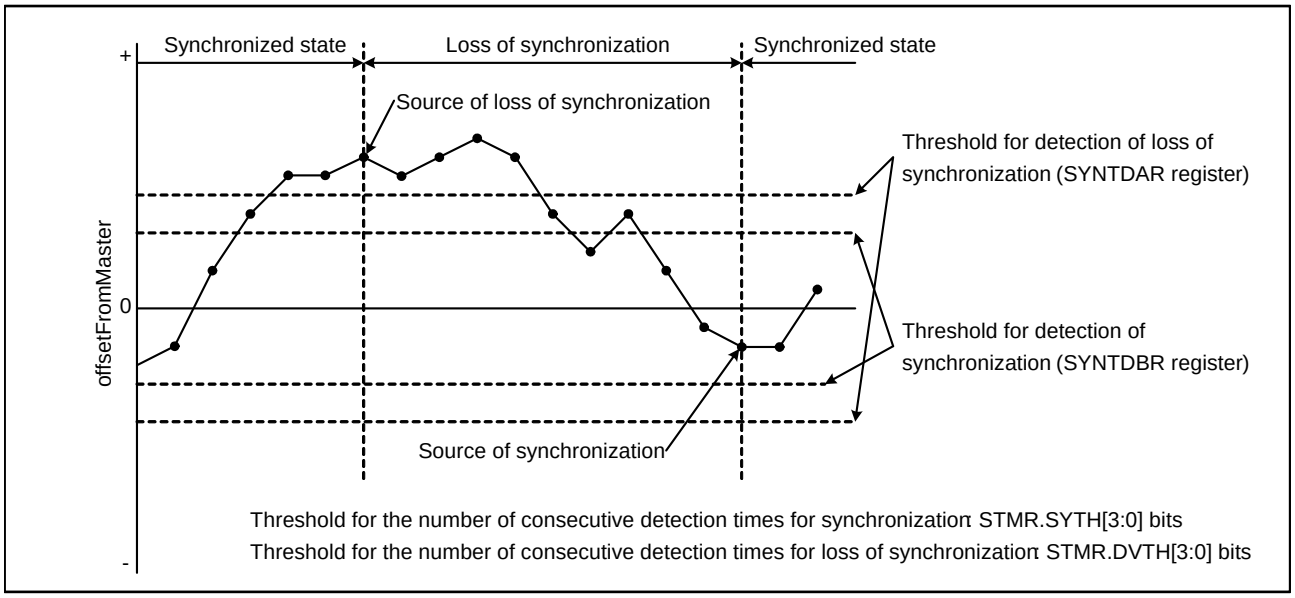


Figure 36.26 Example of a Situation where Synchronization is being Lost and Regained (the number of consecutive detection times is specified as three by the STMR.DVTH[3:0] and SYTH[3:0] bits)

36.3.15.2 Worst 10 Function

The worst 10 function is for imposing limits with regard to exceedingly large and small values among the calculated values for differences in clock gradient.

Values for difference in clock gradient are collected by observing the state of transfer over a specified interval and threshold values to impose limits are extracted from the observed values. With regard to differences in gradient, not only errors in the clock but also fluctuations in network conditions must be considered, and differences in both the positive and negative directions as shown in Figure 36.27 are collected

From among the collected values for positive and negative gradient difference, the largest values for gradient, from the first to the tenth place (worst to tenth worst), are acquired and the tenth worst is used as a threshold value. Fluctuations in the time kept by a slave clock can be suppressed by continually overwriting the tenth worst value for gradient difference with new values large enough to exceed the threshold. Furthermore, periodic collections of gradient values can also be made for updating the threshold values during operations or for the method of setting threshold values from previously measured results

However, note that while fluctuations in the time kept by a slave clock can be suppressed by the effective filtering of values for gradient difference (collecting the worst 10 values and using the tenth worst), following of the time kept by a master clock also becomes slower.

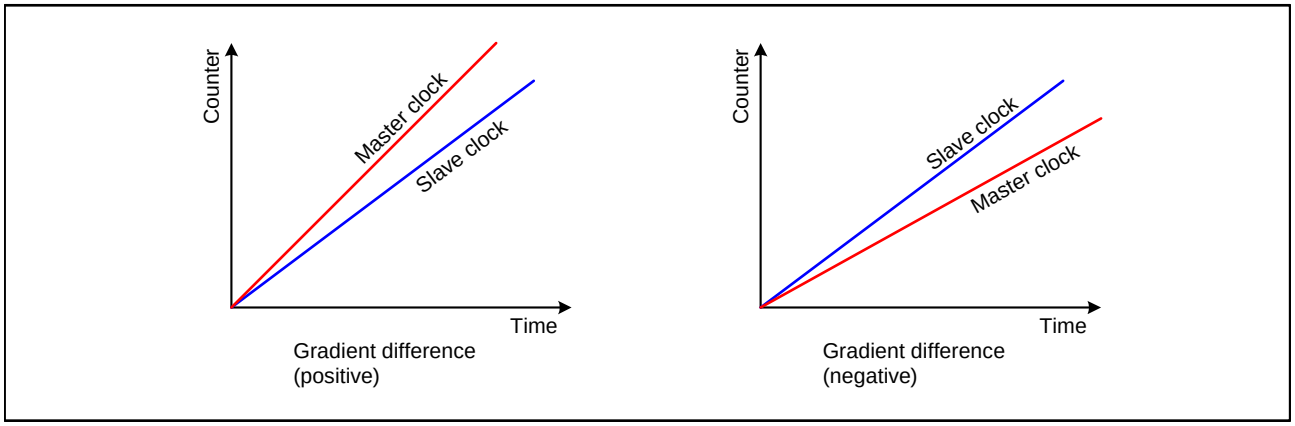


Figure 36.27 Overview of Gradient Differences

36.3.15.3 Collecting Differences in Clock Gradient and Extracting the Worst Ten Values

During slave operation, the EPTPC is capable of calculating the offsetFromMaster values from received messages and then calculating gradient differences between the local clock (acting as a slave clock) and master clock from those values. Specifically, the worst ten values are extracted from the sets of collected values for gradient difference. Either automatic filtering by the hardware or software-triggered filtering can be designated for acquisition of the sets of worst-10 values.

Figure 36.28 gives an overview of the collection of values for gradient difference.

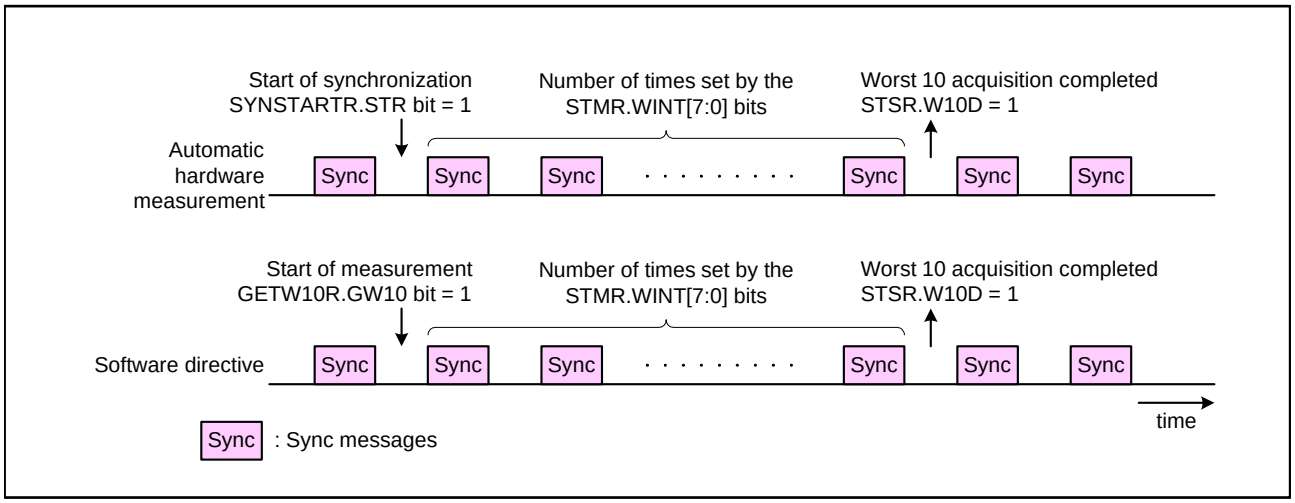


Figure 36.28 Overview of the Collection of Values for Gradient Differences

(1) Collecting Gradient Differences and Extracting the Worst Ten Values by Hardware

The EPTPC automatically collects the values of gradient difference by hardware if the STMR.W10S bit is 0.

When the SYNSTARTR.STR bit is set to 1 (starting slave time synchronization), the EPTPC collects gradient difference values for the number of times set by the STMR.WINT[7:0] bits. When the collection of gradient difference values is finished, the tenth largest values on the positive and negative gradient difference sides are stored as the tenth worst values in registers PW10VRU, PW10VRM, and PW10VRL and registers MW10RU, MW10RM, and MW10RL. When acquisition of the worst 10 values is complete, the STSR.W10D flag becomes 1.

Filtering of gradient difference values by using the stored tenth worst values then proceeds automatically.

Note that if the number of times set by STMR.WINT[7:0] bits are fewer than ten times, the double of the best of the collected values on the positive side is stored in registers PW10VRU, PW10VRM, and PW10VRL, and the half of the best of the collected values on the negative side is stored in registers MW10RU, MW10RM, and MW10RL.

(2) Collecting Gradient Differences and Extracting the Worst Ten Values by Software

The EPTPC collects values of gradient differences by software if the STMR.W10S bit is 1.
When the GETW10R.GW10 bit is set to 1 after time synchronization has started, the EPTPC collects gradient difference values for the number of times set by the STMR.WINT[7:0] bits. When the collection of gradient difference values is finished, the tenth largest values on the positive and negative gradient difference sides are stored as the tenth worst values in registers PW10VRU, PW10VRM, and PW10VRL and registers MW10RU, MW10RM, and MW10RL. When acquisition of the worst 10 values is complete, the STSR.W10D flag becomes 1.
Since filtering of gradient difference values proceeds with the values set in registers PLIMITRU, PLIMITRM, and PLIMITRL and registers MLIMITRU, MLIMITRM, and MLIMITRL as the upper and lower limits for filtering, the values stored in registers PW10VRU, PW10VRM, and PW10VRL and registers MW10RU, MW10RM, and MW10RL must be written to the PLIMITRU, PLIMITRM, and PLIMITRL and registers MLIMITRU, MLIMITRM, and MLIMITRL, respectively.
Note that if the number of times set by the STMR.WINT[7:0] bits is fewer than ten or times, the double of the best of the collected values on the positive is stored in registers PW10VRU, PW10VRM, and PW10VRL, and the half of the best of the collected values on the negative side is stored in registers MW10RU, MW10RM, and MW10RL.
The flowchart in Figure 36.29 shows an example of the procedure for software-triggered acquisition of the worst 10 values.

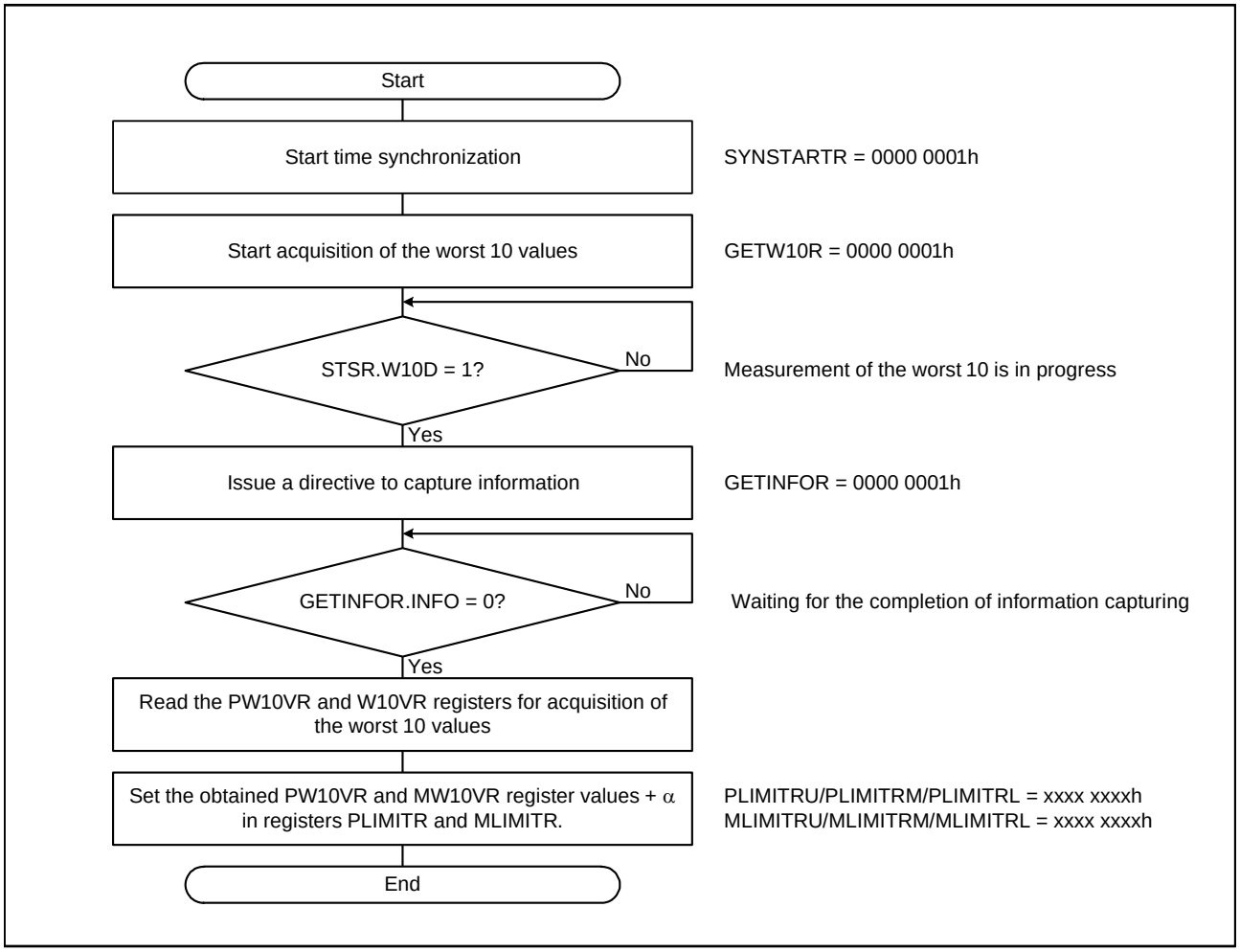


Figure 36.29 Example of the Procedure for Software-Triggered Acquisition of the Worst 10 Values

36.3.16 Local Clock Counter

The local clock counter retains the synchronized time information. The counter starts counting from 0 after the ETHERC is released from the module stop state or the EPTPC is released from the software reset state. The local clock counter can then be set to any desired value. The procedure for setting the initial value in the local clock counter is shown in Figure 36.30.

The time information kept by the local clock counter is also readable. Figure 36.31 shows the procedure for reading the time information kept by the local clock counter.

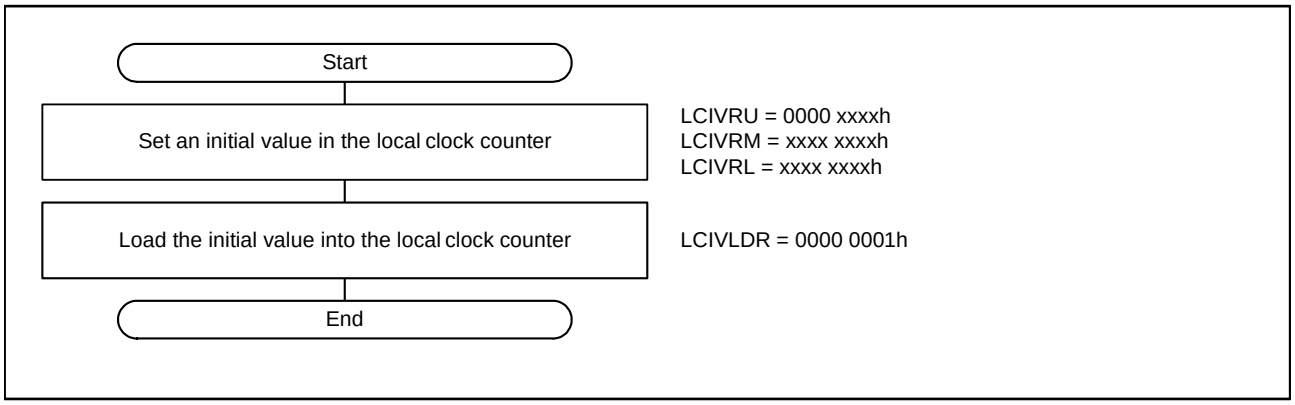


Figure 36.30 Procedure for Setting a New Initial Value in the Local Clock Counter

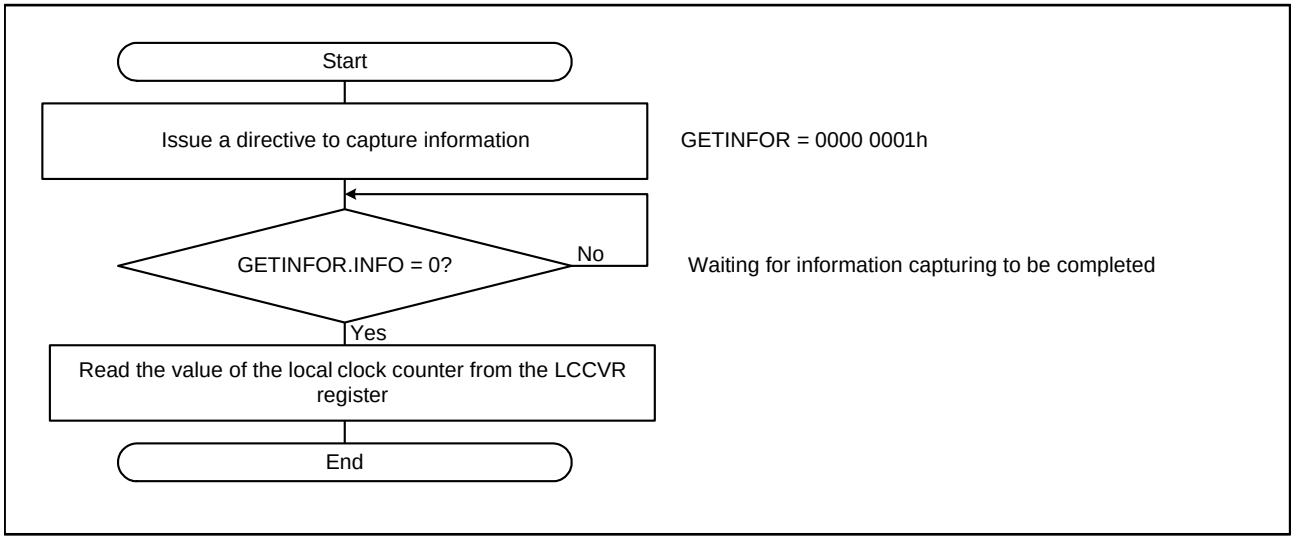


Figure 36.31 Procedure for Reading the Time Kept by the Local Clock Counter

36.3.17 Pulse Output Timer

The STCA module of the EPTPC incorporates six timers (pulse output timers 0 to 5) which operate independently of each other.

The pulse output timers produce periodic pulses, and the rising or falling edges of these pulses can be used as interrupt requests or output to the ELC as event signals. The time at which a pulse output timer starts operating (t_{start}), and the period (t_C) and pulse width (t_W) of the output pulses, can be specified.

The timing of pulse output timer operation is shown in Figure 36.32, and limits on the settings are listed in Table 36.26.

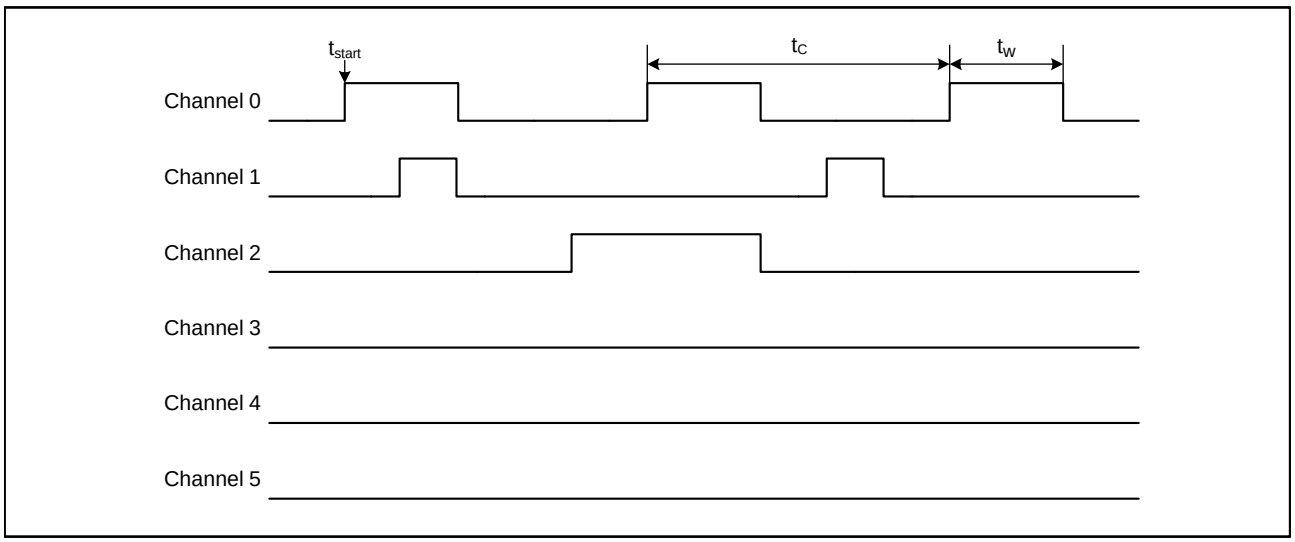


Figure 36.32 Time at which a Pulse Output Timer Starts Operating

Table 36.26 Limits on the Values that can be Specified for a Pulse Output Timer

Item	Description
Cycle (t_C)	From four cycles of the STCA clock to 1 s
Resolution of the cycle	Set in units of nanoseconds. However, the timing of rising edges is rounded by the period of the system clock (50 ns, 40 ns, 20 ns, or 10 ns).
Pulse width (t_W)	From two cycles of the STCA clock to 500 ms
Resolution of the pulse width	Set in units of nanoseconds. However, the timing of falling edges is rounded by the period of the system clock (50 ns, 40 ns, 20 ns, or 10 ns).

36.3.17.1 Procedure for Setting a Pulse Output Timer

Figure 36.33 shows the procedure for setting a pulse output timer.
Note that a timer does not produce periodic pulses if the time set in registers TMSTTRUm and TMSTTRLm (m = 0 to 5) has already passed. Set the time for a pulse output timer to start to a later time than the time when the timer is being set.

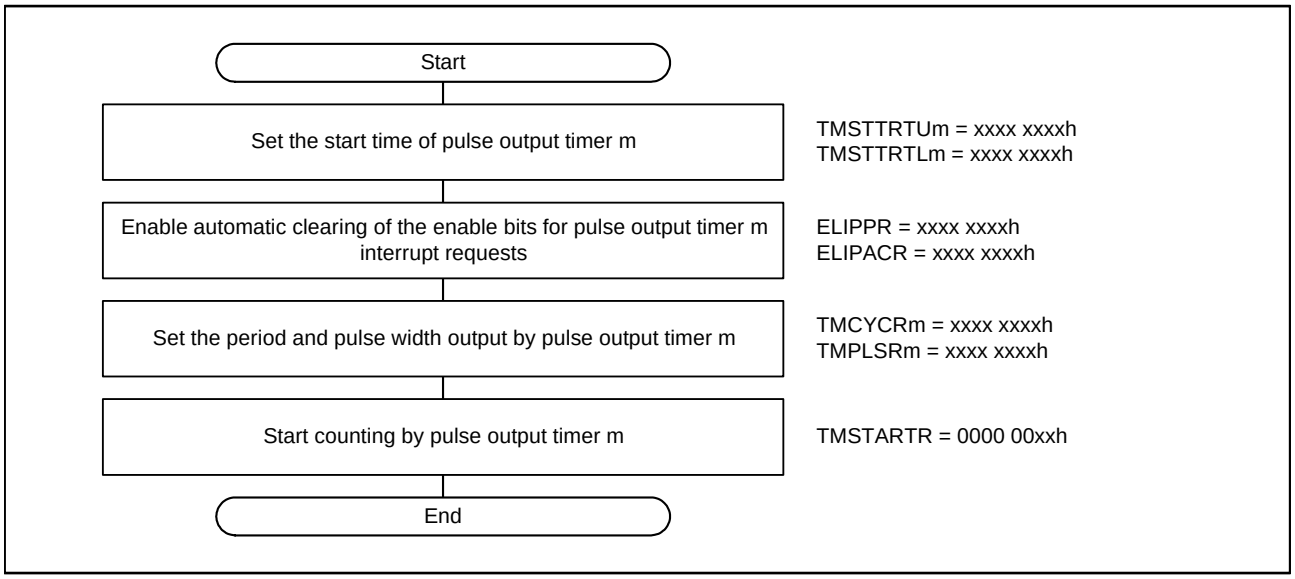


Figure 36.33 Procedure for Setting a Pulse Output Timer

36.3.17.2 Output of periodic pulses as interrupt requests or event signals

MINT interrupt requests, IPLS interrupt requests, or event output signals for the ELC can be generated by detecting rising or falling edges of the periodic pulses from the pulse output timer. The edge for detection and pulse output timer to use can be selected, and automatic clearing of enable bits for the IPLS interrupt or event output can be set. Make the required settings before setting the TMSTARTR.ENm bit to 1 (starting pulse output timer m).

(1) MINT Interrupt Request

Rising edges of the periodic pulses from the pulse output timers can be detected to generate MINT interrupt requests. Falling edges cannot be used in this way. The pulse output timers for use in generating MINT interrupt requests are selected by the MITSELR.MINTENm bits.
Automatic clearing of the enable bits for MINT interrupt requests is not available.

(2) IPLS Interrupt Request

Rising or falling edges of the periodic pulses from the pulse output timers can be detected to generate IPLS interrupt requests. The pulse output timers for use in generating IPLS interrupt requests are selected by the IPTSELR.IPTSELm bits.
Setting the ELIPACR.PLSP or PLSN bit enables automatic clearing of the enable bits for IPLS interrupt requests.

(3) Output of Event Signals to the ELC

Rising or falling edges of the periodic pulses from the pulse output timers can be detected to output event signals to the ELC. The pulse output timers for use in outputting event signals and the edge are selected by the ELIPPR.CYCPm or CYCNm bits.
Setting the ELIPACR.CYCPm or CYCNm bits enables automatic clearing of the event output enable bits.

36.3.18 Priority Control in Transmission

36.3.18.1 Arbitration

In cases of contention between a message for transmission from the PTPEDMAC and a request for relaying from another channel, the PRC-TC module gives priority to the request for relaying from the other channel.
Cases of contention between multiple requests for the transmission of messages by the SYNFP module are also arbitrated in the order of priority shown in Table 36.27.

Table 36.27 Priority in Arbitration for Messages for Transmission

Messages for Transmission	Priority Order	Remark
Sync	1 Highest priority	
Delay_Req, Pdelay_Req	2	There is no device type that simultaneously transmits Delay_Req and Pdelay_Req messages.
Delay_Resp, Pdelay_Resp	3	There is no device type that simultaneously transmits Delay_Resp and Pdelay_Resp messages.
Announce	4	
Relay messages from another channel	5	The PRC-TC module determines the order of priority for these two items.
Messages for transmission from the PTPEDMAC	5	
Messages for transmission from the EDMAC	6	

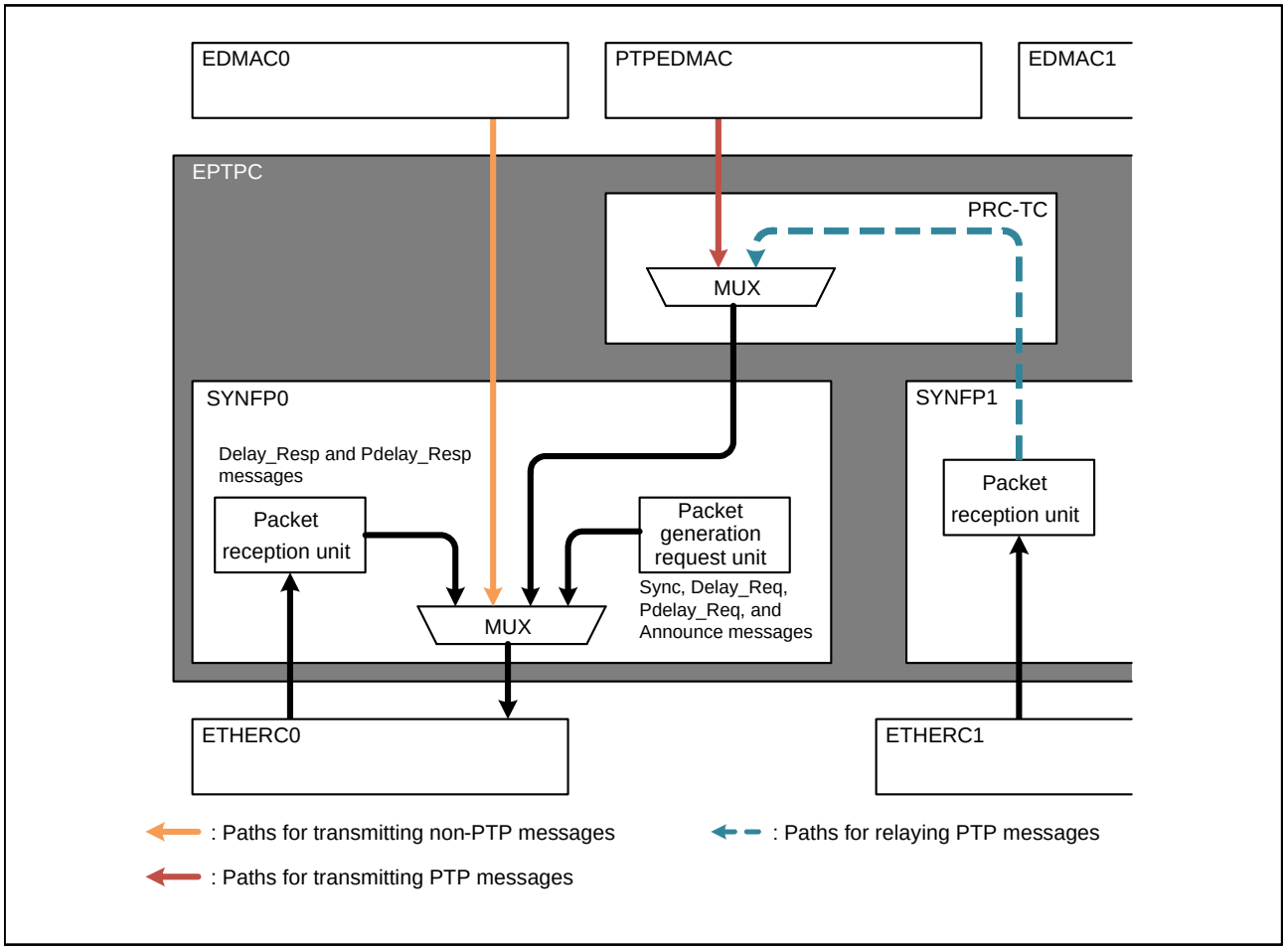


Figure 36.34 Arbitration in Message Transmission

36.3.18.2 Securing of Bandwidth for the Transmission of Sync Messages

The EPTPC secures bandwidth for the transmission of Sync messages, and is capable of handling transmission at very precise intervals.

If the transmission of a Sync message at a fixed interval proceeds at the same time as transmission by the PTPEDMAC or the relaying of messages by another channel is in progress, because transmission of the Sync message proceeds when the other processing is complete, the interval for transmission will no longer be fixed.

By securing bandwidth for the transmission of Sync messages, the transmission of messages from EDMAC0, EDMAC1, and the PTPEDMAC is limited, and the transmission of Sync messages can be handled without fluctuations. Securing of bandwidth for the transmission of Sync messages can be disabled by setting the SYCONFR.SBDIS bit to 1. Figure 36.35 gives a schematic view of securing bandwidth for the transmission of Sync messages.

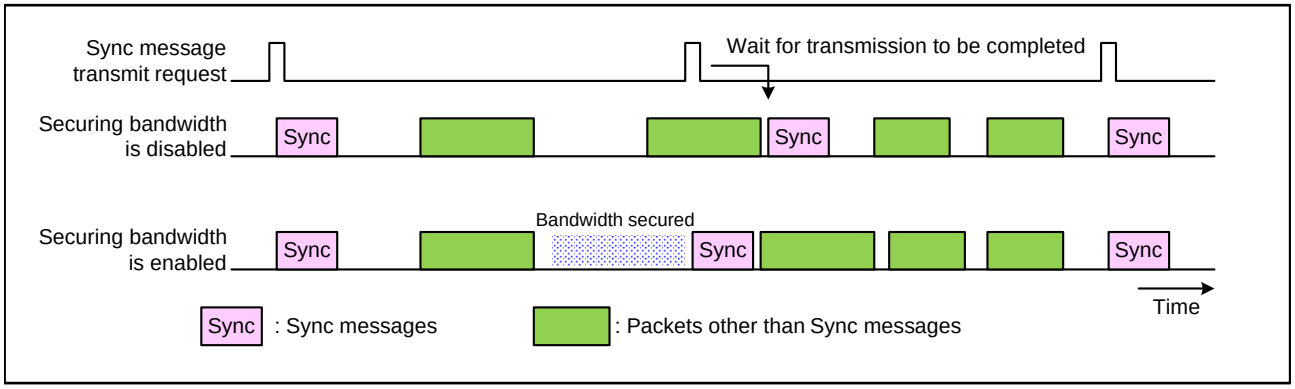


Figure 36.35 Securing of Bandwidth for the Transmission of Sync Messages

36.3.18.3 Securing of Transmission Interval

In the transmission of messages by the ETHERC, if there is a fixed delay from the time of a request for transmission to the time of transmission on the MII of Ethernet port 0 or 1, time stamp values of PTP messages can be used for accurately obtaining the size of the delay during operation as a slave. However, in the case of continuous transfer and so on, in situations where processing of messages to wait for interpacket gap times is required, delay times may fluctuate. Specifying an interval for frame transmission in SYCONFR.TCYC[7:0] bits to control the interval between the completion of transmission and the next request for transmission makes the ETHERC able to secure the reliability of time stamp values, thus avoiding the effects of interpacket gap times and obtaining a fixed delay for transmission.

36.4 Interrupts

The EPTPC has the MINT interrupt request and IPLS interrupt request.
Figure 36.36 shows the relation between the MINT interrupt request and IPLS interrupt request. Figure 36.37 shows the details on interrupt requests of the pulse output timer.

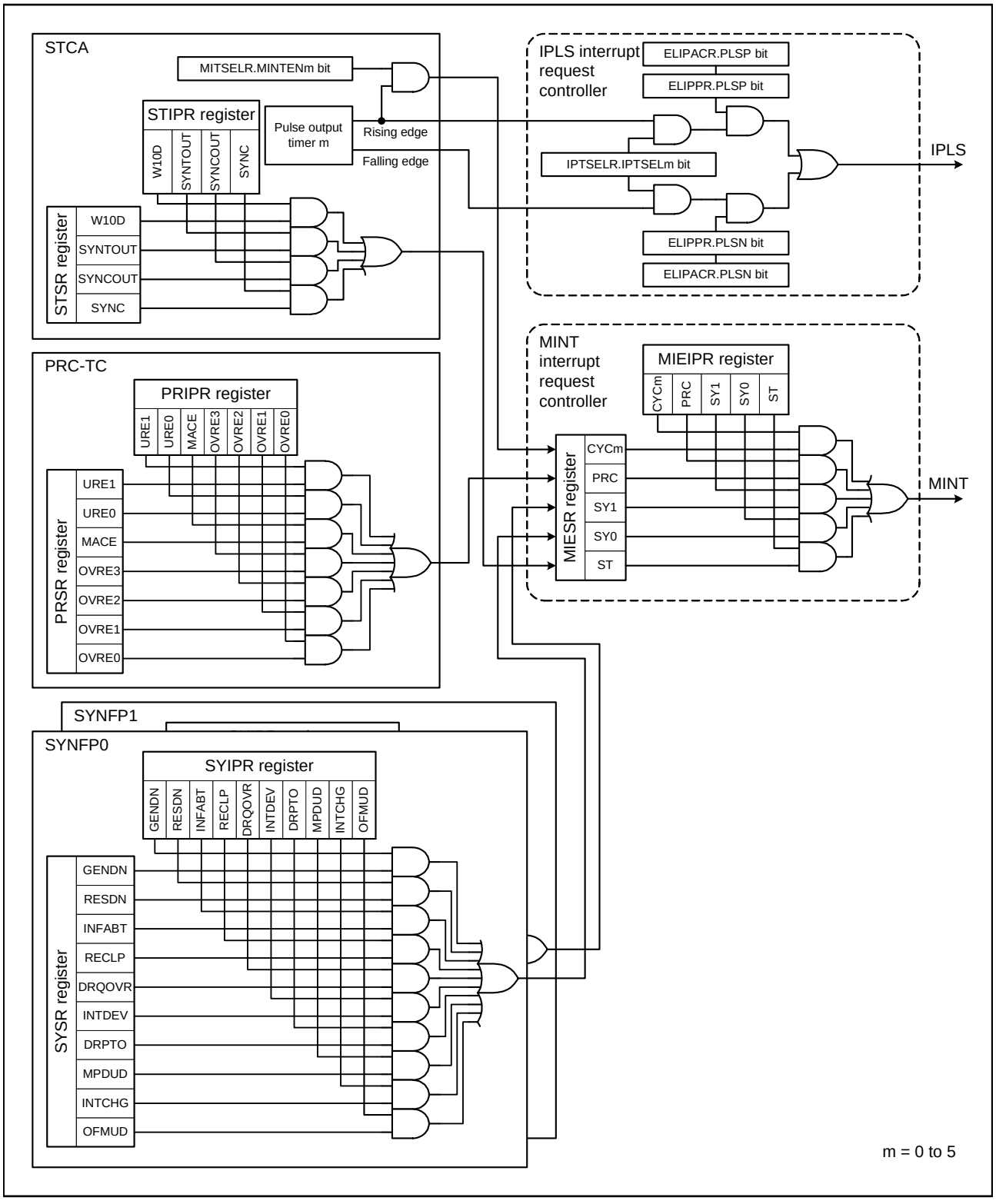


Figure 36.36 MINT Interrupt Request and IPLS Interrupt Request

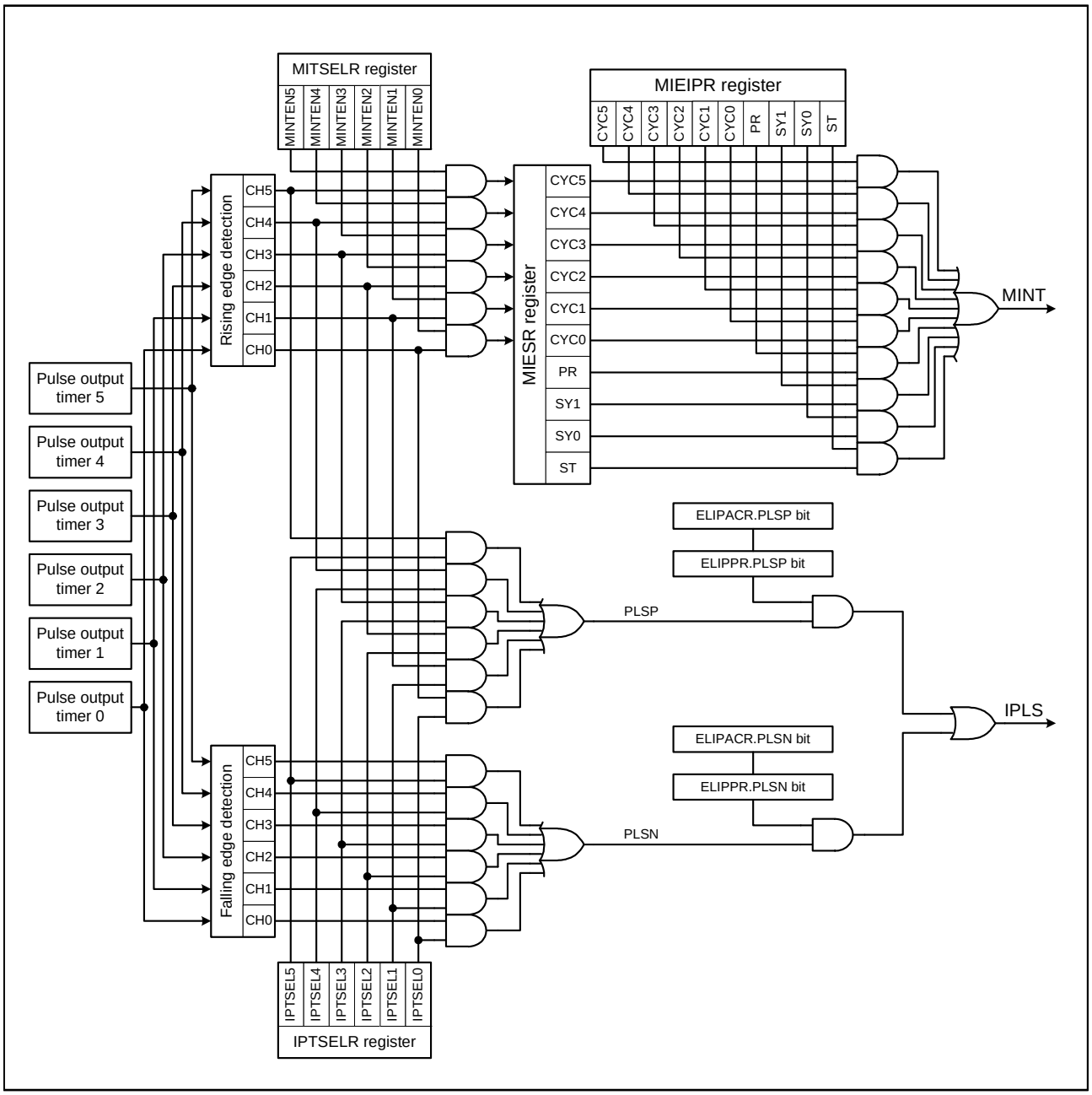


Figure 36.37 Details on Interrupt Requests of the Pulse Output Timer

36.5 Event Link (Output)

The EPTPC can output an event to the ELC by detecting the rising edge or falling edge of the pulse that the pulse output timer outputs.

Figure 36.37 shows the relation between the pulse output timer and the ELC.

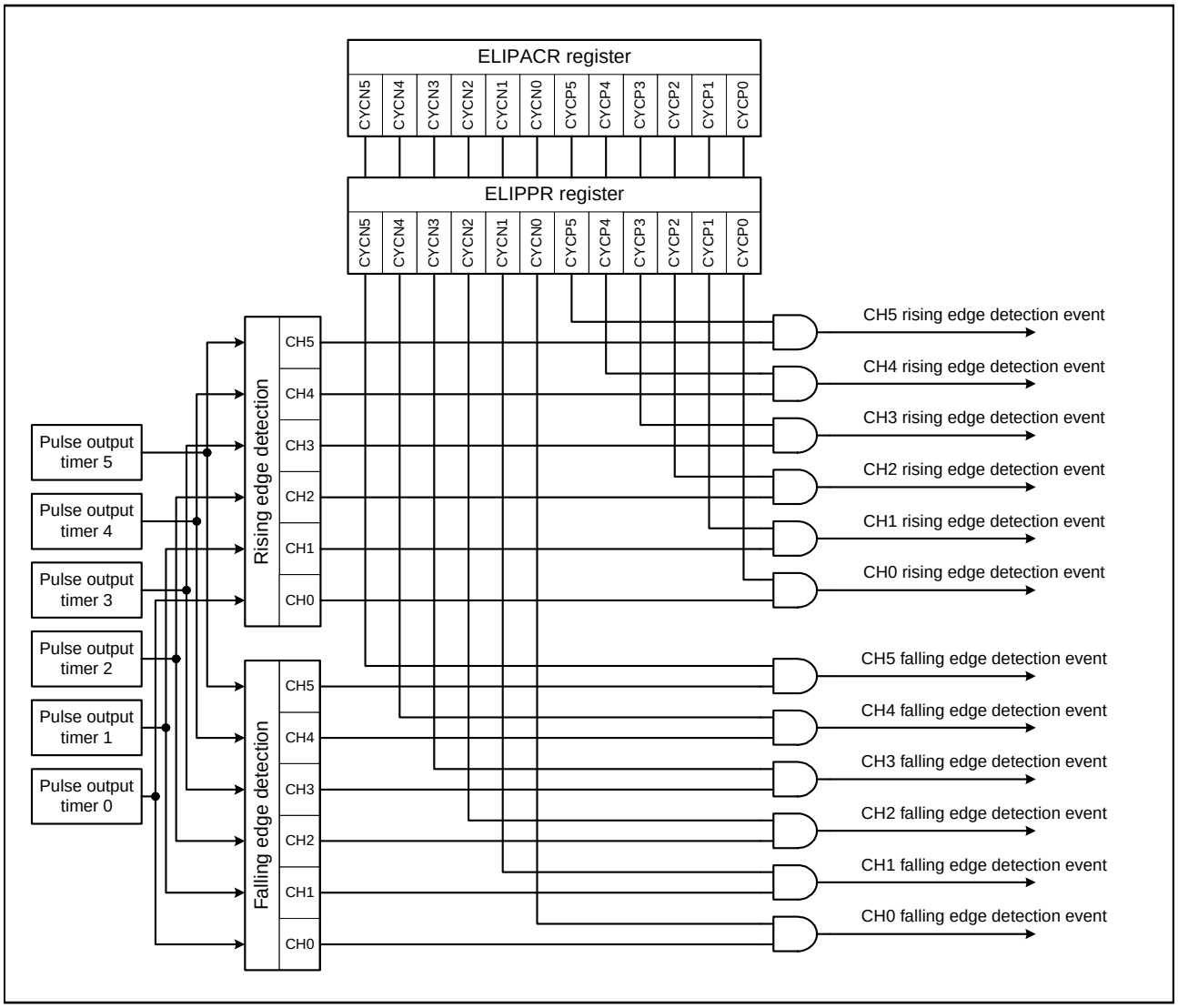


Figure 36.38 Relation between the Pulse Output Timer and ELC

36.6 Usage Notes

36.6.1 Restrictions on Access to Registers

Even if the EPTPC does not handle time synchronization, multicast filtering and so on can still be applied in the internal operation of the module. However, if Ethernet port 0 or Ethernet port 1 is stopped, some registers in the EPTPC module become inaccessible because supply of the clock signal from the corresponding EDMAC stops.
Table 36.28 summarizes the restrictions on access to registers.

Table 36.28 Restrictions on Register Access

Usage of Ethernet Ports		Allocation of Register Addresses for Access					
Ethernet Port 0 (EDMAC0)	**Ethernet Port 1 (EDMAC1)**	**000C 0500h to 000C 05FFh**	**000C 4000h to 000C 403Fh**	**000C 4040h to 000C 43FFh (STCA)**	**000C 4400h to 000C 47FFh (PRC-TC)**	**000C 4800h to 000C 4BFFh (SYNFP0)**	**000C 4C00h to 000C 4FFFh (SYNFP1)**
In use	In use	Accessible	Accessible	Accessible	Accessible	Accessible	Accessible
In use	Not in use	Accessible	Accessible	Accessible	Accessible	Accessible	Access prohibited
Not in use	In use	Accessible	Accessible	Accessible	Accessible	Access prohibited	Accessible
Not in use	Not in use	Accessible	Access prohibited	Access prohibited	Access prohibited	Access prohibited	Access prohibited

Access to an access-prohibited register can lead to a bus timeout error. If a bus timeout error occurs, use the PTRSTR.RESET bit to reset the EPTPC by software. Refer to section 16.7.1.2, Timeout for details on the bus timeout error.

36.6.2 Wait Cycles for Register Access

Access to registers in the EPTPC involves the arbitration of different clock signals, specifically the peripheral module clock signal (PCLKA), the STCA clock signal, and the MII clock signals such as TX_CLK. Accordingly, numbers of wait cycles for register access differ with the combination of the frequency settings of the peripheral module clock signal, of the STCA clock signal, and of the MII clock signals.

Table 36.29 gives examples of numbers of wait cycles for different combinations of clock frequency. Add 1 to 2 cycles to these values to obtain the number of access cycles.

Table 36.29 Wait Cycles for Register Access (when the frequency of the STCA clock is 20 MHz)

	STCA Clock = 20 MHz							
	Peripheral Module Clock PCLKA = 120 MHz				Peripheral Module Clock PCLKA = 20 MHz			
	MII Clock 25 MHz (100 Mbps)		MII Clock 2.5 MHz (10 Mbps)		MII Clock 25 MHz (100 Mbps)		MII Clock 2.5 MHz (10 Mbps)	
Address Range	Read	Write	Read	Write	Read	Write	Read	Write
000C 0500h to 000C 05FFh	2	2	2	2	2	2	2	2
000C 4000h to 000C 403Fh	4	4	4	4	4	4	4	4
000C 4040h to 000C 43FFh (STCA)	7	27 to 41*1	7	27 to 41*1	7	15 to 17*1	7	15 to 17*1
000C 4400h to 000C 47FFh (PRC-TC)	8	8	8	8	8	8	8	8
000C 4800h to 000C 4BFFh (SYNFP0)	8	23 to 33*2	8	111 to 209*2	8	15 to 17*2	8	31 to 49*2
000C 4C00h to 000C 4FFFh (SYNFP1)	8	23 to 33*2	8	111 to 209*2	8	15 to 17*2	8	31 to 49*2

Note 1. The number of wait cycles in access to the STCA-related registers (Wstca) can be calculated to be in the following range from the periods of the peripheral module clock ($t_{c(PCLKA)}$) and STCA clock ($t_{c(STCA)}$).

The minimum value of Wstca = Int ($t_{c(STCA)}/t_{c(PCLKA)}$) × 2 + 15 ($t_{c(PCLKA)} \leq t_{c(STCA)}$)
= 15 ($t_{c(PCLKA)} > t_{c(STCA)}$)

The maximum value of Wstca = Int ($t_{c(STCA)}/t_{c(PCLKA)}$) × 4 + 17 ($t_{c(PCLKA)} \leq t_{c(STCA)}$)
= 17 ($t_{c(PCLKA)} > t_{c(STCA)}$)

- Int(A) is the calculation of the largest integer not greater than A.
- This calculation presumes that the CPU clock and peripheral module clock have the same periods.

For example, if the frequency of the peripheral module clock is 120 MHz and that of the STCA clock is 1/6 that of the peripheral module clock (= 20 MHz),
the minimum value of Wstca = Int (50 [ns]/8.3 [ns]) × 2 + 15 = 27, and
the maximum value of Wstca = Int (50 [ns]/8.3 [ns]) × 4 + 17 = 41.
If REF50CK0 or REF50CK1 is used as the STCA clock, the frequency of the STCA clock is 25 MHz.

Note 2. The number of wait cycles in access to the SYNFP-related registers (Wsynf) can be calculated to be in the following range from the periods of the peripheral module clock ($t_{c(PCLKA)}$) and MII clock ($t_{c(MII)}$).

The minimum value of Wsynf = Int ($t_{c(MII)}/t_{c(PCLKA)}$) × 2 + 15 ($t_{c(PCLKA)} \leq t_{c(MII)}$)
= 15 ($t_{c(PCLKA)} > t_{c(MII)}$)

The maximum value of Wsynf = Int ($t_{c(MII)}/t_{c(PCLKA)}$) × 4 + 17 ($t_{c(PCLKA)} \leq t_{c(MII)}$)
= 17 ($t_{c(PCLKA)} > t_{c(MII)}$)

- Int(A) is the calculation of the largest integer not greater than A.
- This calculation presumes that the CPU clock and peripheral module clock have the same periods.

For example, if the frequency of the peripheral module clock is 120 MHz and the transmission rate is 10 Mbps (so the MII clock is running at 2.5 MHz),
the minimum value of Wsynf = Int (400 [ns]/8.3 [ns]) × 2 + 15 = 111, and
the maximum value of Wsynf = Int (400 [ns]/8.3 [ns]) × 4 + 17 = 209.

37. DMA Controller for the Ethernet Controller (EDMACa)

37.1 Overview

This MCU has three channels for the ethernet controller direct memory access controller (EDMAC): two channels for the ethernet controller (ETHERC) and one channel for the PTP controller (EPTPC). EDMAC0 and EDMAC1 control data transmission and reception for ETHERC0 and ETHERC1, respectively. The PTPEDMAC controls data transmission and reception for ETHERC0 and ETHERC1 according to the EPTPC settings. In this chapter, channel numbers are indicated as n (n = 0, 1).

The EDMAC controls most of the transmit/receive buffer management for communications. This reduces the load on the CPU and allows efficient data transmission and reception. The data transfers are controlled according to the information (referred to as descriptors) on the memory.

Table 37.1 lists the EDMAC specifications. Figure 37.1 shows the EDMAC configuration, and Figure 37.2 shows the configuration of descriptors and transmit/receive buffers on the memory.

Table 37.1 EDMAC Specifications

Item	Description
Data transmission and reception	• Controls data transmission and reception according to descriptors • Supports single buffer frame transmission and reception (1 buffer per frame), and multi-buffer frame transmission and reception (multiple buffers per frame)
Functions	• Minimizes system bus occupation time using block transfer (32-byte units) • Writes back the transmit/receive frame state to descriptors • Inserts padding in receive data
Low power consumption function	The EDMAC can be set to the module-stop state to reduce power consumption.

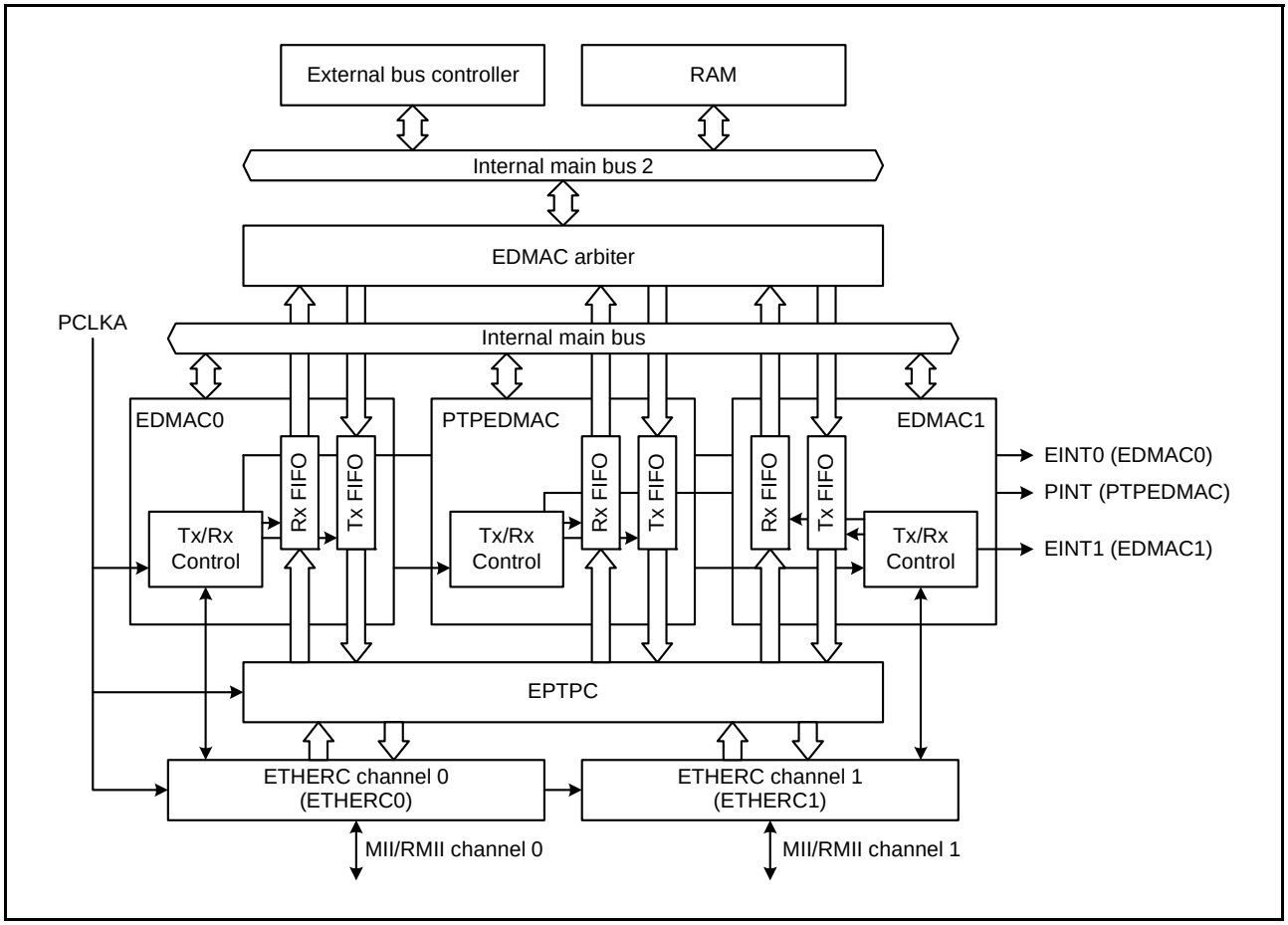


Figure 37.1 EDMAC Configuration

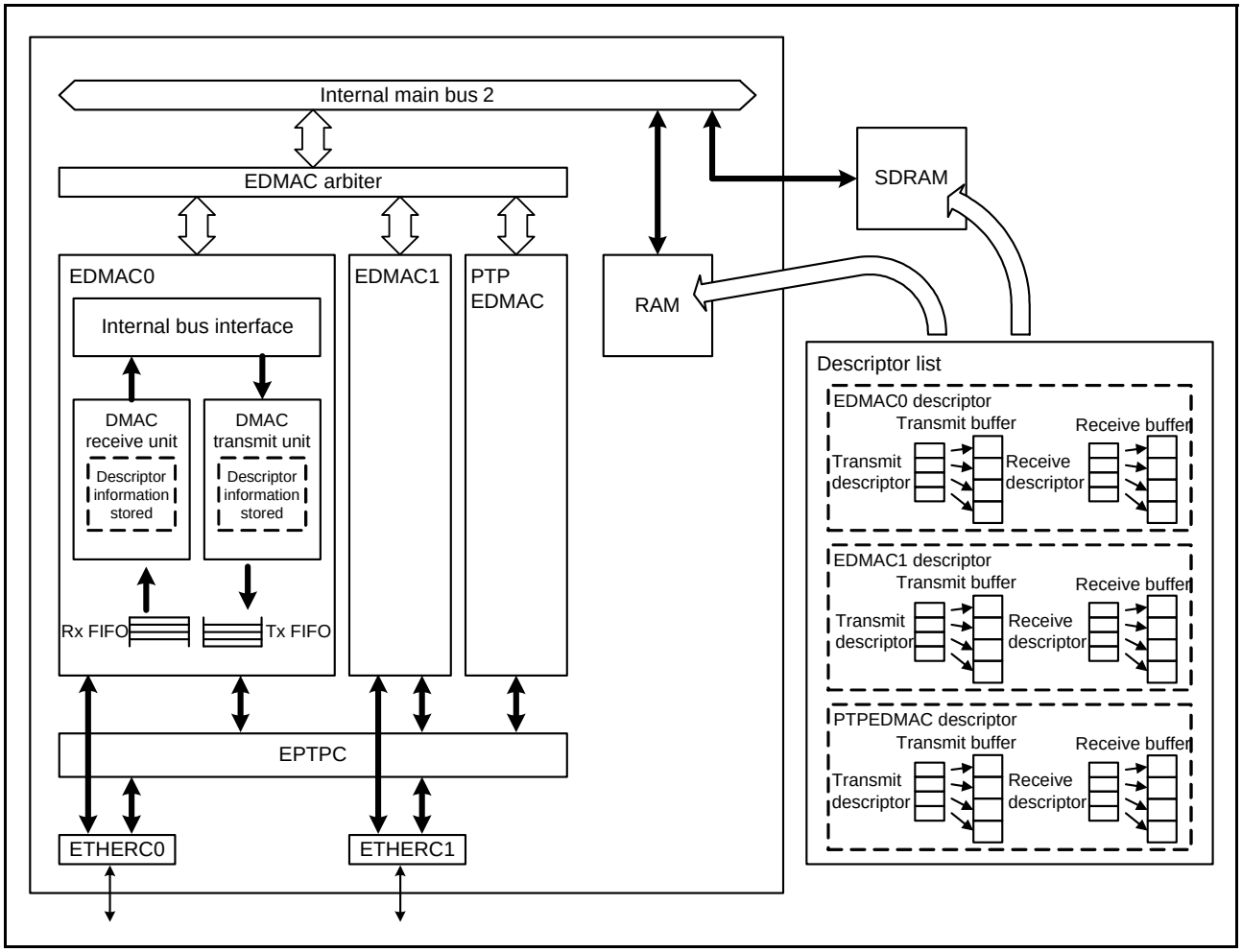


Figure 37.2 Configuration of Descriptors and Transmit/Receive Buffers on the Memory

37.2 Register Descriptions

37.2.1 EDMAC Mode Register (EDMR)

Address(es): EDMAC0.EDMR 000C 0000h, EDMAC1.EDMR 000C 0200h, PTPEDMAC.EDMR 000C 0400h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	DE	DL[1:0]		—	—	—	SWR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SWR	Software Reset	When 1 is written, the corresponding channels of the EDMAC and ETHERC are reset. Note that for the PTPEDMAC, the ETHERC are not reset. Registers TDLAR, RDLAR, RMFCR, TFUCR, and RFOCR are not reset with this bit. The read value is 0.	R/W
b3 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W
b5, b4	DL[1:0]	Transmit/Receive Descriptor Length	b5 b4 0 0: 16 bytes 0 1: 32 bytes 1 0: 64 bytes 1 1: 16 bytes	R/W
b6	DE	Big Endian Mode/Little Endian Mode *1	0: Big endian mode 1: Little endian mode	R/W
b31 to b7	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note 1. This setting applies to data for the transmit/receive buffer. It does not apply to transmit/receive descriptors and registers.

The EDMR register controls EDMAC operation.

Set the EDMR register during the initialization process after a reset. When rewriting this register outside of the initialization process, set the SWR bit to 1 to reset the EDMAC and ETHERC, and then set this register again. If the ETHERC and EDMAC are reset during data transmission or reception, abnormal data may be sent on the line. Do not rewrite this register while the ETHERC transmit or receive function is enabled. It takes 64 cycles of the peripheral module clock (PCLKA) to initialize the ETHERC and EDMAC. Complete the initialization process before accessing registers in the ETHERC and EDMAC.

37.2.2 EDMAC Transmit Request Register (EDTRR)

Address(es): EDMAC0.EDTRR 000C 0008h, EDMAC1.EDTRR 000C 0208h, PTPEDMAC.EDTRR 000C 0408h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TR	Transmit Request	When 1 is written, the EDMAC reads the corresponding descriptor and transmits frames where the TD0.TACT bit is 1. The TR bit becomes 0 after all the valid frames are transmitted. Writing 0 to this bit has no effect.	R/W
b31 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

The EDTRR register controls EDMAC transmission.
After the EDMAC has transmitted one frame, it reads the next descriptor. When the TD0.TACT bit in the descriptor is 1, the EDMAC continues transmission. When the TD0.TACT bit is 0, the EDMAC sets the TR bit to 0 and stops transmission.

37.2.3 EDMAC Receive Request Register (EDRRR)

Address(es): EDMAC0.EDRRR 000C 0010h, EDMAC1.EDRRR 000C 0210h, PTPEDMAC.EDRRR 000C 0410h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RR	Receive Request	0: Receive function is disabled. *1 1: Receive descriptor is read, and the receive function is enabled.	R/W
b31 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note 1. If the receive function is disabled during frame reception, write-back to the receive descriptor is not performed successfully. Subsequent pointers for reading a receive descriptor become abnormal and the EDMAC cannot operate normally. In this case, to enable the EDMAC receive function again, execute a software reset by setting the EDMR.SWR bit to 1.
To disable the EDMAC receive function without resetting the EDMAC, set the ETHERCn.ECMR.RE bit to 0. Next, after the EDMAC has completed reception and write-back to the receive descriptor has been confirmed, set the RR bit to 0.

The EDRRR register controls EDMAC reception.

When the RR bit becomes 1, the EDMAC reads the receive descriptor. When the RD0.RACT bit is 1, the EDMAC waits for a receive request from the ETHERC.

When the EDMAC has received data for the receive buffer size, it reads the next descriptor and waits to receive a frame. If the RD0.RACT bit is 0, the EDMAC sets the RR bit to 0 and stops reception.

37.2.4 Transmit Descriptor List Start Address Register (TDLAR)

Address(es): EDMAC0.TDLAR 000C 0018h, EDMAC1.TDLAR 000C 0218h, PTPEDMAC.TDLAR 000C 0418h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	The start address of the transmit descriptor list is set. Set the start address according to the descriptor length selected by the EDMR.DL[1:0] bits. 16-byte boundary: Lower 4 bits = 0000b 32-byte boundary: Lower 5 bits = 00000b 64-byte boundary: Lower 6 bits = 000000b	R/W

The TDLAR register sets the start address of the transmit descriptor list.
Align each descriptor on the corresponding boundary to the descriptor length selected by the EDMR.DL[1:0] bits.
Do not rewrite the TDLAR register during transmission. Rewrite the TDLAR register while the EDTRR.TR bit is 0.

37.2.5 Receive Descriptor List Start Address Register (RDLAR)

Address(es): EDMAC0.RDLAR 000C 0020h, EDMAC1.RDLAR 000C 0220h, PTPEDMAC.RDLAR 000C 0420h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	—	The start address of the receive descriptor list is set. Set the start address according to the descriptor length selected by the EDMR.DL[1:0] bits. 16-byte boundary: Lower 4 bits = 0000b 32-byte boundary: Lower 5 bits = 00000b 64-byte boundary: Lower 6 bits = 000000b	R/W

The RDLAR register sets the start address of the receive descriptor list.
Align each descriptor on the corresponding boundary to the descriptor length selected by the EDMR.DL[1:0] bits.
Do not rewrite the RDLAR register during reception. Rewrite the RDLAR register while the EDRRR.RR bit is 0.

37.2.6 ETHERC/EDMAC Status Register (EDMACn.EESR)

Address(es): EDMAC0.EESR 000C 0028h, EDMAC1.EESR 000C 0228h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	TWB	—	—	—	TABT	RABT	RFCOF	—	ECI	TC	TDE	TFUF	FR	RDE	RFOF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CND	DLC	CD	TRO	RMAF	—	—	RRF	RTLF	RTSF	PRE	CERF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CERF	CRC Error Flag	0: CRC error has not been detected. 1: CRC error has been detected.	R/W
b1	PRE	PHY-LSI Receive Error Flag	0: PHY-LSI receive error has not been detected. 1: PHY-LSI receive error has been detected.	R/W
b2	RTSF	Frame-Too-Short Error Flag	0: Frame-too-short error has not been detected. 1: Frame-too-short error has been detected.	R/W
b3	RTLF	Frame-Too-Long Error Flag	0: Frame-too-long error has not been detected. 1: Frame-too-long error has been detected.	R/W
b4	RRF	Alignment Error Flag	0: Alignment error has not been detected. 1: Alignment error has been detected.	R/W
b6, b5	—	Reserved	The read value is 0. The write value should be 0.	R/W
b7	RMAF	Multicast Address Frame Receive Flag	0: Multicast address frame has not been received. 1: Multicast address frame has been received.	R/W
b8	TRO	Transmit Retry Over Flag	0: Transmit retry-over condition has not been detected. 1: Transmit retry-over condition has been detected.	R/W
b9	CD	Late Collision Detect Flag	0: Late collision has not been detected. 1: Late collision has been detected during frame transmission.	R/W
b10	DLC	Loss of Carrier Detect Flag	0: Loss of carrier has not been detected. 1: Loss of carrier has been detected during frame transmission.	R/W
b11	CND	Carrier Not Detect Flag	0: A carrier has been detected when transmission starts. 1: A carrier has not been detected during preamble transmission.	R/W
b15 to b12	—	Reserved	The read value is 0. The write value should be 0.	R/W
b16	RFOF	Receive FIFO Overflow Flag	0: Overflow has not occurred. 1: Overflow has occurred.	R/W
b17	RDE	Receive Descriptor Empty Flag	0: The EDMAC detects that the receive descriptor valid bit (RD0.RACT) is 1. 1: The EDMAC detects that the receive descriptor valid bit (RD0.RACT) is 0.	R/W
b18	FR	Frame Receive Flag	0: Frame has not been received. 1: Frame has been received. Update of the receive descriptor is complete.	R/W
b19	TFUF	Transmit FIFO Underflow Flag	0: Underflow has not occurred. 1: Underflow has occurred.	R/W
b20	TDE	Transmit Descriptor Empty Flag	0: The EDMAC detects that the transmit descriptor valid bit (TD0.TACT) is 1. 1: The EDMAC detects that the transmit descriptor valid bit (TD0.TACT) is 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b21	TC	Frame Transfer Complete Flag	0: Transfer have not been completed, or no transfer has been requested. 1: All frames indicated by the transmit descriptor have been completely transferred to the transmit FIFO.	R/W
b22	ECI	ETHERC Status Register Source Flag	0: ETHERC status interrupt source has not been detected. 1: ETHERC status interrupt source has been detected.	R *1
b23	—	Reserved	The read value is 0. The write value should be 0.	R/W
b24	RFCOF	Receive Frame Counter Overflow Flag	0: Receive frame counter has not overflowed. 1: Receive frame counter has overflowed.	R/W
b25	RABT	Receive Abort Detect Flag	0: Frame reception has not been aborted or no reception has been requested. 1: Frame reception has been aborted.	R/W
b26	TABT	Transmit Abort Detect Flag	0: Frame transmission has not been aborted or no transmission has been requested. 1: Frame transmission has been aborted.	R/W
b29 to b27	—	Reserved	The read value is 0. The write value should be 0.	R/W
b30	TWB	Write-Back Complete Flag	0: Write-back has not been completed, or no transmission has been requested. 1: Write-back to the transmit descriptor has been completed.	R/W
b31	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note 1. The ECI flag is read-only. When the source in the ETHERCn.ECSR register is cleared, the ECI flag is also cleared.

The EDMACn.EESR register indicates the ETHERC and EDMAC communication status.
Each flag in the EESR register can be output as an interrupt request signal (EINTn) from the EDMAC. Each flag, excluding the ECI flag, becomes 0 by writing 1. The value of each flag is not changed by writing 0. Each interrupt source can be enabled by the corresponding bit in the EDMACn.EESIPR register.

CERF Flag (CRC Error Flag)

The CERF flag becomes 1 when an error is detected in checking the frame check sequence (FCS) field of the receive frame.

PRE Flag (PHY-LSI Receive Error Flag)

The PRE flag indicates the RX_ER signal output from the PHY-LSI is high.

RTSF Flag (Frame-Too-Short Error Flag)

The RTSF flag indicates that a received frame was less than 64 bytes.

RTLF Flag (Frame-Too-Long Error Flag)

The RTLF flag indicates that a received frame is greater than the upper limit of the receive frame length set in the ETHERCn.RFLR register. The excess data is discarded.

RRF Flag (Alignment Error Flag)

The RRF flag indicates that a frame is not an integral number of octets. The last word that is not an integral number of octets is not transferred.
An error frame shorter than 16 bytes is discarded and the RRF flag does not become 1.

RMAF Flag (Multicast Address Frame Receive Flag)

The RMAF flag indicates that a multicast frame has been received.

TRO Flag (Transmit Retry Over Flag)

This flag indicates that a collision occurred again during the 15th retry of frame transmission.

CD Flag (Late Collision Detect Flag)

The CD flag indicates that a late collision has been detected during frame transmission.

DLC Flag (Loss of Carrier Detect Flag)

The DLC flag indicates that a loss of carrier has been detected during frame transmission.

CND Flag (Carrier Not Detect Flag)

The CND flag becomes 1 when a carrier has not been detected during preamble transmission.

RFOF Flag (Receive FIFO Overflow Flag)

The RFOF flag indicates that the receive FIFO has overflowed during frame reception.

RDE Flag (Receive Descriptor Empty Flag)

The RDE flag indicates that the read receive descriptor is invalid.
When this flag becomes 1, set the RD0.RACT bit in the receive descriptor to 1 and set the EDRRR.RR bit to 1 to resume reception.

FR Flag (Frame Receive Flag)

The FR flag indicates that a frame has been received and the receive descriptor has been updated. The FR flag becomes 1 every time a frame is received.

TFUF Flag (Transmit FIFO Underflow Flag)

The TFUF flag indicates that no data remains in the transmit FIFO during frame transmission. Incomplete data is sent on the line.

TDE Flag (Transmit Descriptor Empty Flag)

The TDE flag indicates that the TD0.TACT bit of the transmit descriptor is 0 while the previous transmit descriptor indicates that the frame is not complete (TD0.TFP[1:0] bits are 10b or 00b) in multi-buffer frame transmission. As a result, an incomplete frame may be sent.
When this flag becomes 1, perform a software reset and then set the EDTRR.TR bit to 1 to resume transmission. Transmission starts from the address stored in the TDLAR register.

TC Flag (Frame Transfer Complete Flag)

The TC flag indicates that all the data specified by the transmit descriptor has been transmitted from the ETHERC. This flag becomes 1 when one frame has been transmitted in single-buffer frame transmission or when the last data of a frame is transmitted in multi-buffer frame transmission and the TD0.TACT bit in the next transmit descriptor is 0. After frame transmission is completed, the EDMAC writes the transfer status back to the descriptor.

ECI Flag (ETHERC Status Register Source Flag)

The ECI flag becomes 1 when an interrupt request is generated by the ETHERCn.ECSR register.

RFCOF Flag (Receive Frame Counter Overflow Flag)

The RFCOF flag indicates that the next frame reception starts while the number of frames stored in the receive FIFO reaches the maximum number of frames (16 frames). Note that the received frame is discarded while the RFCOF flag is 1.

RABT Flag (Receive Abort Detect Flag)

The RABT flag indicates that the ETHERC has aborted frame reception due to a CRC error, PHY-LSI receive error, frame-too-short error, frame-too-long error, or other error.
An error frame shorter than 16 bytes is discarded and the RABT flag does not become 1.

TABT Flag (Transmit Abort Detect Flag)

The TABT flag indicates that the ETHERC has aborted frame transmission due to transmit retry over, loss of carrier, no carrier detection, or other error.

TWB Flag (Write-Back Complete Flag)

The TWB flag indicates the EDMAC has completed writing back to the descriptor after frame transmission. Note that this flag becomes 1 after each frame transmission when the TRIMD.TIM bit is 0. This flag becomes 1 only when the TRIMD.TIS bit is 1.

37.2.7 PTP/EDMAC Status Register (PTPEDMAC.EESR)

Address(es): PTPEDMAC.EESR 000C 0428h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	TWB	—	—	—	TABT	—	RFCOF	—	—	TC	TDE	TFUF	FR	RDE	RFOF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MACE	RPORT	—	—	PVER	TYPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	TYPE[3:0]	PTP v2 Message Type Flag	b3 b0 0 0 0 0: Sync 0 0 0 1: Delay_Req 0 0 1 0: Pdelay_Req 0 0 1 1: Pdelay_Resp 1 0 0 0: Follow_Up 1 0 0 1: Delay_Resp 1 0 1 0: Pdelay_Resp_Follow_Up 1 0 1 1: Announce 1 1 0 0: Signaling 1 1 0 1: Management Settings other than above are reserved.	R/W
b4	PVER	PTP v2 Packet Flag	0: The current packet is not a PTP v2 packet. 1: The current packet is a PTP v2 packet.	R/W
b6, b5	—	Reserved	The read value is 0. The write value should be 0.	R/W
b7	RPORT	Receive Port Flag	0: Port 0 1: Port 1	R/W
8	MACE	MAC Address Mismatch Flag	0: The source MAC address of transmit frame data matches the set value. 1: The source MAC address of transmit frame data does not match the set value.	R/W
b15 to b9	—	Reserved	The read value is 0. The write value should be 0.	R/W
b16	RFOF	Receive FIFO Overflow Flag	0: Overflow has not occurred. 1: Overflow has occurred.	R/W
b17	RDE	Receive Descriptor Empty Flag	0: The EDMAC detects that the receive descriptor valid bit (RD0.RACT) is 1. 1: The EDMAC detects that the receive descriptor valid bit (RD0.RACT) is 0.	R/W
b18	FR	Frame Receive Flag	0: Frame has not been received. 1: Frame has been received. The receive descriptor has been updated.	R/W
b19	TFUF	Transmit FIFO Underflow Flag	0: Underflow has not occurred. 1: Underflow has occurred.	R/W
b20	TDE	Transmit Descriptor Empty Flag	0: The EDMAC detects that the transmit descriptor valid bit (TD0.TACT) is 1. 1: The EDMAC detects that the transmit descriptor valid bit (TD0.TACT) is 0.	R/W
b21	TC	Frame Transfer Complete Flag	0: Transfer has not been completed, or transfer has not been requested. 1: All frames indicated by the transmit descriptor have been completely transferred to the transmit FIFO.	R/W
b23, b22	—	Reserved	The read value is 0. The write value should be 0.	R/W
b24	RFCOF	Receive Frame Counter Overflow Flag	0: Receive frame counter has not overflowed. 1: Receive frame counter has overflowed.	R/W

Bit	Symbol	Bit Name	Description	R/W
b25	—	Reserved	The read value is 0. The write value should be 0.	R/W
b26	TABT	Transmit Abort Detect Flag	0: Frame transmission has not been aborted or transmission has not been requested. 1: Frame transmission has been aborted.	R/W
b29 to b27	—	Reserved	The read value is 0. The write value should be 0.	R/W
b30	TWB	Write-Back Complete Flag	0: Write-back has not been completed, or transmission has not been requested. 1: Write-back to the transmit descriptor has been completed.	R/W
b31	—	Reserved	The read value is 0. The write value should be 0.	R/W

The PTPEDMAC.EESR register indicates the PTPEDMAC communication status.
Each flag in the EESR register can be output as an interrupt request signal (PINT) from the PTPEDMAC. Each flag becomes 0 by writing 1. The value of each flag is not changed by writing 0. Each interrupt source, excluding the TYPE[3:0] flag, can be enabled by setting the corresponding bit in the PTPEDMAC.EESIPR register.

TYPE[3:0] Flags (PTP v2 Message Type Flag)

The TYPE[3:0] flags indicate the type of received PTP message.

PVER Flag (PTP v2 Packet Flag)

The PVER flag indicates whether the received packet is a PTP v2 packet.

RPORT Flag (Receive Port Flag)

The RPORT flag indicates which Ethernet port has been used for receiving PTP messages.

MACE Flag (MAC Address Mismatch Flag)

The MACE flag indicates the source MAC address is different from the set value.

RFOF Flag (Receive FIFO Overflow Flag)

The RFOF flag indicates that the receive FIFO has overflowed during frame reception.

RDE Flag (Receive Descriptor Empty Flag)

The RDE flag indicates that the read receive descriptor is invalid.
When this flag becomes 1, set the RD0.RACT bit in the receive descriptor to 1 and set the EDRRR.RR bit to 1 to resume reception.

FR Flag (Frame Receive Flag)

The FR flag indicates that a frame has been received and the receive descriptor has been updated. The FR flag becomes 1 every time a frame is received.

TFUF Flag (Transmit FIFO Underflow Flag)

The TFUF flag indicates that no data remains in the transmit FIFO during frame transmission. Incomplete data is sent on the line.

TDE Flag (Transmit Descriptor Empty Flag)

The TDE flag indicates that the TD0.TACT bit of the transmit descriptor is 0 while the previous transmit descriptor indicates that the frame is not complete (TD0.TFP[1:0] bits are 10b or 00b) in multi-buffer frame transmission. As a result, an incomplete frame may be sent.
When this flag becomes 1, perform a software reset and then set the EDTRR.TR bit to 1 to resume transmission.

Transmission starts from the address stored in the TDLAR register.

TC Flag (Frame Transfer Complete Flag)

The TC flag indicates that all the data specified by the transmit descriptor has been transmitted from the ETHERC. This flag becomes 1 when one frame has been transmitted in single-buffer frame transmission or when the last data of a frame is transmitted in multi-buffer frame transmission and the TD0.TACT bit in the next transmit descriptor is 0. After frame transmission is completed, the PTPEDMAC writes the transfer status back to the descriptor.

RFCOF Flag (Receive Frame Counter Overflow Flag)

The RFCOF flag indicates that the next frame reception starts while the number of frames stored in the receive FIFO reaches the maximum number of frames (16 frames). Note that the received frame is discarded while the RFCOF flag is 1.

TABT Flag (Transmit Abort Detect Flag)

The TABT flag indicates that the ETHERC has aborted frame transmission due to transmit retry over, loss of carrier, no carrier detection, or other error.

TWB Flag (Write-Back Complete Flag)

The TWB flag indicates the PTPEDMAC has completed writing back to the descriptor after frame transmission. Note that this flag becomes 1 after each frame transmission when the TRIMD.TIM bit is 0. This flag becomes 1 only when the TRIMD.TIS bit is 1.

37.2.8 ETHERC/EDMAC Status Interrupt Enable Register (EDMACn.EESIPR)

Address(es): EDMAC0.EESIPR 000C 0030h, EDMAC1.EESIPR 000C 0230h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	TWBIP	—	—	—	TABTIP	RABTIP	RFCOFIP	—	ECIIP	TCIP	TDEIP	TFUFIP	FRIP	RDEIP	RFOFIP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CNDIP	DLCIP	CDIP	TROIP	RMAFIP	—	—	RRFIP	RTLFIP	RTSFIP	PREIP	CERFIP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CERFIP	CRC Error Interrupt Request Enable	0: CRC error interrupt request is disabled. 1: CRC error interrupt request is enabled.	R/W
b1	PREIP	PHY-LSI Receive Error Interrupt Request Enable	0: PHY-LSI receive error interrupt request is disabled. 1: PHY-LSI receive error interrupt request is enabled.	R/W
b2	RTSFIP	Frame-Too-Short Error Interrupt Request Enable	0: Frame-too-short error interrupt request is disabled. 1: Frame-too-short error interrupt request is enabled.	R/W
b3	RTLFIP	Frame-Too-Long Error Interrupt Request Enable	0: Frame-too-long error interrupt request is disabled. 1: Frame-too-long error interrupt request is enabled.	R/W
b4	RRFIP	Alignment Error Interrupt Request Enable	0: Alignment error interrupt request is disabled. 1: Alignment error interrupt request is enabled.	R/W
b6, b5	—	Reserved	The read value is 0. The write value should be 0.	R/W
b7	RMAFIP	Multicast Address Frame Receive Interrupt Request Enable	0: Multicast address frame receive interrupt request is disabled. 1: Multicast address frame receive interrupt request is enabled.	R/W
b8	TROIP	Transmit Retry Over Interrupt Request Enable	0: Transmit retry over interrupt request is disabled. 1: Transmit retry over interrupt request is enabled.	R/W
b9	CDIP	Late Collision Detect Interrupt Request Enable	0: Late collision detect interrupt request is disabled. 1: Late collision detect interrupt request is enabled.	R/W
b10	DLCIP	Loss of Carrier Detect Interrupt Request Enable	0: Loss of carrier detect interrupt request is disabled. 1: Loss of carrier detect interrupt request is enabled.	R/W
b11	CNDIP	Carrier Not Detect Interrupt Request Enable	0: Carrier not detect interrupt request is disabled. 1: Carrier not detect interrupt request is enabled.	R/W
b15 to b12	—	Reserved	The read value is 0. The write value should be 0.	R/W
b16	RFOFIP	Receive FIFO Overflow Interrupt Request Enable	0: Overflow interrupt request is disabled. 1: Overflow interrupt request is enabled.	R/W
b17	RDEIP	Receive Descriptor Empty Interrupt Request Enable	0: Receive descriptor empty interrupt request is disabled. 1: Receive descriptor empty interrupt request is enabled.	R/W
b18	FRIP	Frame Receive Interrupt Request Enable	0: Frame reception interrupt request is disabled. 1: Frame reception interrupt request is enabled.	R/W
b19	TFUFIP	Transmit FIFO Underflow Interrupt Request Enable	0: Underflow interrupt request is disabled. 1: Underflow interrupt request is enabled.	R/W
b20	TDEIP	Transmit Descriptor Empty Interrupt Request Enable	0: Transmit descriptor empty interrupt request is disabled. 1: Transmit descriptor empty interrupt request is enabled.	R/W
b21	TCIP	Frame Transfer Complete Interrupt Request Enable	0: Frame transmission complete interrupt request is disabled. 1: Frame transmission complete interrupt request is enabled.	R/W

Bit	Symbol	Bit Name	Description	R/W
b22	ECIIP	ETHERC Status Register Source Interrupt Request Enable	0: ETHERC status interrupt request is disabled. 1: ETHERC status interrupt request is enabled.	R/W
b23	—	Reserved	The read value is 0. The write value should be 0.	R/W
b24	RFCOFIP	Receive Frame Counter Overflow Interrupt Request Enable	0: Receive frame counter overflow interrupt request is disabled. 1: Receive frame counter overflow interrupt request is enabled.	R/W
b25	RABTIP	Receive Abort Detect Interrupt Request Enable	0: Receive abort detect interrupt request is disabled. 1: Receive abort detect interrupt request is enabled.	R/W
b26	TABTIP	Transmit Abort Detect Interrupt Request Enable	0: Transmit abort detect interrupt request is disabled. 1: Transmit abort detect interrupt request is enabled.	R/W
b29 to b27	—	Reserved	The read value is 0. The write value should be 0.	R/W
b30	TWBIP	Write-Back Complete Interrupt Request Enable	0: Write-back complete interrupt request is disabled. 1: Write-back complete interrupt request is enabled.	R/W
b31	—	Reserved	The read value is 0. The write value should be 0.	R/W

The EDMACn.EESIPR register enables interrupt requests corresponding to bits in the EDMACn.EESR register. When a bit in this register is 1, the corresponding interrupt request is enabled.

37.2.9 PTP/EDMAC Status Interrupt Enable Register (PTPEDMAC.EESIPR)

Address(es): PTPEDMAC.EESIPR 000C 0430h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	TWBIP	—	—	—	TABTIP	—	RFCOFIP	—	—	TCIP	TDEIP	TFUFIP	FRIP	RDEIP	RFOFIP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MACEIP	RPORTIP	—	—	PVERIP	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	The read value is 0. The write value should be 0.	R/W
b4	PVERIP	PTP v2 Packet Receive Interrupt Request Enable	0: PTP v2 packet receive interrupt request is disabled. 1: PTP v2 packet receive interrupt request is enabled.	R/W
b6, b5	—	Reserved	The read value is 0. The write value should be 0.	R/W
b7	RPORTIP	Receive Port Interrupt Request Enable	0: Interrupt request at a fame reception on port 1 is disabled. 1: Interrupt request at a fame reception on port 1 is enabled.	R/W
b8	MACEIP	MAC Address Mismatch Interrupt Request Enable	0: This bit disables an interrupt request generated when the source MAC address of transmit frame data does not match the set value. 1: This bit enables an interrupt request generated when the source MAC address of transmit frame data does not match the set value.	R/W
b15 to b9	—	Reserved	The read value is 0. The write value should be 0.	R/W
b16	RFOFIP	Receive FIFO Overflow Interrupt Request Enable	0: Overflow interrupt request is disabled. 1: Overflow interrupt request is enabled.	R/W
b17	RDEIP	Receive Descriptor Empty Interrupt Request Enable	0: Receive descriptor empty interrupt request is disabled. 1: Receive descriptor empty interrupt request is enabled.	R/W
b18	FRIP	Frame Receive Interrupt Request Enable	0: Frame receive interrupt request is disabled. 1: Frame receive interrupt request is enabled.	R/W
b19	TFUFIP	Transmit FIFO Underflow Interrupt Request Enable	0: Underflow interrupt request is disabled. 1: Underflow interrupt request is enabled.	R/W
b20	TDEIP	Transmit Descriptor Empty Interrupt Request Enable	0: Transmit descriptor empty interrupt request is disabled. 1: Transmit descriptor empty interrupt request is enabled.	R/W
b21	TCIP	Frame Transfer Complete Interrupt Request Enable	0: Frame transmission complete interrupt request is disabled. 1: Frame transmission complete interrupt request is enabled.	R/W
b23, b22	—	Reserved	The read value is 0. The write value should be 0.	R/W
b24	RFCOFIP	Receive Frame Counter Overflow Interrupt Request Enable	0: Receive frame counter overflow interrupt request is disabled. 1: Receive frame counter overflow interrupt request is enabled.	R/W
b25	—	Reserved	The read value is 0. The write value should be 0.	R/W
b26	TABTIP	Transmit Abort Detect Interrupt Request Enable	0: Transmit abort detect interrupt request is disabled. 1: Transmit abort detect interrupt request is enabled.	R/W
b29 to b27	—	Reserved	The read value is 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b30	TWBIP	Write-Back Complete Interrupt Request Enable	0: Write-back complete interrupt request is disabled. 1: Write-back complete interrupt request is enabled.	R/W
b31	—	Reserved	The read value is 0. The write value should be 0.	R/W

The PTPEDMAC.EESIPR register enables interrupt requests corresponding to bits in the PTPEDMAC.EESR register. When a bit in this register is 1, the corresponding interrupt request is enabled.

37.2.10 ETHERC/EDMAC Transmit/Receive Status Copy Enable Register (EDMACn.TRSCER)

Address(es): EDMAC0.TRSCER 000C 0038h, EDMAC1.TRSCER 000C 0238h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	RMAFCE	—	—	RRFCE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	The read value is 0. The write value should be 0.	R/W
b4	RRFCE	RRF Flag Copy Enable	0: The EESR.RRF flag status is reflected in the RD0.RFE bit of the receive descriptor. 1: The EESR.RRF flag status is not reflected in the RD0.RFE bit of the receive descriptor.	R/W
b6, b5	—	Reserved	The read value is 0. The write value should be 0.	R/W
b7	RMAFCE	RMAF Flag Copy Enable	0: The EESR.RMAF flag status is reflected in the RD0.RFE bit of the receive descriptor. 1: The EESR.RMAF flag status is not reflected in the RD0.RFE bit of the receive descriptor.	R/W
b31 to b8	—	Reserved	The read value is 0. The write value should be 0.	R/W

The EDMACn.TRSCER register selects whether the receive status indicated by flags EDMACn.EESR.RMAF and RRF is reflected in the RFE bit of the receive descriptor as a summary. The bits in the TRSCER register correspond to bits in the EESR register that have the same number. When setting the RMAFCE or RRFCE bit to 0, the corresponding receive status (EESR.RMAF or RRF flag) is reflected in the RFE bit of the receive descriptor. When setting the RMAFCE or RRFCE bit to 1, the corresponding receive status is not reflected in the RFE bit.

37.2.11 Missed-Frame Counter Register (RMFCR)

Address(es): EDMAC0.RMFCR 000C 0040h, EDMAC1.RMFCR 000C 0240h, PTPEDMAC.RMFCR 000C 0440h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MFC[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	MFC[15:0]	Missed-Frame Counter	These bits indicate the number of frames that are discarded and not transferred to the receive buffer during reception.	R/W
b31 to b16	—	Reserved	The read value is 0. The write value should be 0.	R/W

The RMFCR register indicates that the number of frames that could not be stored in the receive FIFO and were therefore discarded during reception. When the receive FIFO overflows, it stops receiving data, and the remaining frames are discarded. At the same time, the RMFCR register value is incremented. When the RMFCR register value reaches FFFFh, incrementing is halted. When any value is written to the RMFCR register, the counter value becomes 0.
For the frame that has not been completely received, after data in the receive FIFO is transferred to the receive buffer, the RACT bit in the receive descriptor 0 (RD0) becomes 0 (descriptor disabled), the RFS9 bit becomes 1 (receive FIFO overflow), and the EDMACn.EESR.RFOF or PTPEDMAC.EESR.RFOF flag becomes 1 (overflow has occurred).

37.2.12 Transmit FIFO Threshold Register (TFTR)

Address(es): EDMAC0.TFTR 000C 0048h, EDMAC1.TFTR 000C 0248h, PTPEDMAC.TFTR 000C 0448h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TFT[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b10 to b0	TFT[10:0]	Transmit FIFO Threshold	000h: Store and forward mode 001h to 00Ch: Setting prohibited 00Dh to 200h: The threshold is the set value multiplied by 4. Example: 00Dh: 52 bytes 040h: 256 bytes 100h: 1024 bytes 200h: 2048 bytes 201h to 7FFh: Setting prohibited	R/W
b31 to b11	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note 1. When starting transmission before one frame data has been completely written, take care to prevent an underflow.
Note 2. To prevent a transmit underflow, using the initial value (store and forward mode) is recommended.

The TFTR register sets the transmit FIFO threshold at which the first transmission starts. The actual threshold is the set value multiplied by 4.

The ETHERC starts transmission when the amount of data in the transmit FIFO exceeds the number of bytes set in the TFTR register, when the transmit FIFO is full, or when one frame data has been completely written. Set the TFTR register while the EDTRR.TR bit is 0.

37.2.13 FIFO Depth Register (FDR)

Address(es): EDMAC0.FDR 000C 0050h, EDMAC1.FDR 000C 0250h, PTPEDMAC.FDR 000C 0450h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TFD[4:0]					—	—	—	RFD[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	RFD[4:0]	Receive FIFO Depth	b4 b0 01111: 4096 bytes Settings other than above are prohibited.	R/W
b7 to b5	—	Reserved	The read value is 0. The write value should be 0.	R/W
b12 to b8	TFD[4:0]	Transmit FIFO Depth	b12 b8 00111: 2048 bytes Settings other than above are prohibited.	R/W
b31 to b13	—	Reserved	The read value is 0. The write value should be 0.	R/W

The FDR register sets the transmit and receive FIFO depths.
Set this register to 0000 070Fh before starting transmission and reception.

37.2.14 Receive Method Control Register (RMCR)

Address(es): EDMAC0.RMCR 000C 0058h, EDMAC1.RMCR 000C 0258h, PTPEDMAC.RMCR 000C 0458h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RNR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RNR	Receive Request Reset	0: EDRRR.RR bit (receive request bit) is set to 0 when one frame has been received. 1: EDRRR.RR bit (receive request bit) is not set to 0 when one frame has been received.	R/W
b31 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

The RMCR register sets how to control the EDRRR.RR bit when receiving a frame.
Since the EDRRR.RR bit becomes 0 when one frame has been received while the RNR bit is 0, the RR bit needs to be set to 1 by software to receive the subsequent frame.
Since the EDRRR.RR bit does not become 0 when one frame has been received while the RNR bit is 1, the EDMAC reads the next receive descriptor and continues receiving frames. However, the EDRRR.RR bit becomes 0 when the RD0.RACT bit in the receive descriptor is 0. It is recommended to set the RNR bit to 1 when receiving data continuously.
Set the RMCR register while the EDRRR.RR bit is 0.

37.2.15 Transmit FIFO Underflow Counter (TFUCR)

Address(es): EDMAC0.TFUCR 000C 0064h, EDMAC1.TFUCR 000C 0264h, PTPEDMAC.TFUCR 000C 0464h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	UNDER[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	UNDER[15:0]	Transmit FIFO Underflow Count	These bits indicate how many times the transmit FIFO has underflowed. The counter stops when the counter value reaches FFFFh.	R/W
b31 to b16	—	Reserved	The read value is 0. The write value should be 0.	R/W

The TFUCR register indicates how many times the transmit FIFO has underflowed.
When writing any value to the TFUCR register, the counter value becomes 0.

37.2.16 Receive FIFO Overflow Counter (RFOCR)

Address(es): EDMAC0.RFOCR 000C 0068h, EDMAC1.RFOCR 000C 0268h, PTPEDMAC.RFOCR 000C 0468h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVER[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	OVER[15:0]	Receive FIFO Overflow Count	These bits indicate how many times the receive FIFO has overflowed. The counter stops when the counter value reaches FFFFh.	R/W
b31 to b16	—	Reserved	The read value is 0. The write value should be 0.	R/W

The RFOCR register indicates how many times the receive FIFO has overflowed.
When writing any value to the RFOCR register, the counter value becomes 0.

37.2.17 Independent Output Signal Setting Register (IOSR)

Address(es): EDMAC0.IOSR 000C 006Ch, EDMAC1.IOSR 000C 026Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ELB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ELB	External Loopback Mode	0: The ETn_EXOUT pin outputs low. 1: The ETn_EXOUT pin outputs high.	R/W
b31 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

The IOSR register selects the output level of the ETHERC external output pin (ETn_EXOUT) in external loopback mode.

The ELB bit value is output to the ETn_EXOUT pin of the MCU as is, which can be used to set loopback mode for the PHY-LSI. To use the loopback function of the PHY-LSI through this register, the PHY-LSI needs to be provided with the pin to be connected to the ETn_EXOUT pin.

37.2.18 Flow Control Start FIFO Threshold Setting Register (FCFTR)

Address(es): EDMAC0.FCFTR 000C 0070h, EDMAC1.FCFTR 000C 0270h, PTPEDMAC.FCFTR 000C 0470h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	RFFO[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RFDO[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RFDO[2:0]	Receive FIFO Data PAUSE Output Threshold	b2 b1 b0 0 0 0: When 224 (256 - 32) bytes of data is stored in the receive FIFO. 0 0 1: When 480 (512 - 32) bytes of data is stored in the receive FIFO. : 1 1 0: When 1760 (1792 - 32) bytes of data is stored in the receive FIFO. 1 1 1: When 2016 (2048 - 32) bytes of data is stored in the receive FIFO.	R/W
b15 to b3	—	Reserved	The read value is 0. The write value should be 0.	R/W
b18 to b16	RFFO[2:0]	Receive FIFO Frame PAUSE Output Threshold	b18 b17 b16 0 0 0: When 2 receive frames have been stored in the receive FIFO. 0 0 1: When 4 receive frames have been stored in the receive FIFO. 0 1 0: When 6 receive frames have been stored in the receive FIFO. : 1 1 0: When 14 receive frames have been stored in the receive FIFO. 1 1 1: When 16 receive frames have been stored in the receive FIFO.	R/W
b31 to b19	—	Reserved	The read value is 0. The write value should be 0.	R/W

The FCFTR register sets the ETHERC flow control (sets the threshold for automatically transmitting a PAUSE frame). The threshold can be set using the data size (RFDO[2:0] bits) and the number of frames (RFFO[2:0] bits) stored in the receive FIFO. The flow control starts when the stored data size or number of stored frames reaches its threshold.

37.2.19 Receive Data Padding Insert Register (RPADIR)

Address(es): EDMAC0.RPADIR 000C 0078h, EDMAC1.RPADIR 000C 0278h, PTPEDMAC.RPADIR 000C 0478h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PADS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	PADR[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PADR[5:0]	Padding Slot	00h: Padding is inserted at the head of received data. 01h: Padding is inserted between the 1st byte and 2nd byte of received data. : 3Eh: Padding is inserted between the 62nd byte and 63rd byte of received data. 3Fh: Padding is inserted between the 63rd byte and 64th byte of received data	R/W
b15 to b6	—	Reserved	The read value is 0. The write value should be 0.	R/W
b17, b16	PADS[1:0]	Padding Size	b17 b16 0 0: No padding is inserted. 0 1: 1 byte is inserted. 1 0: 2 bytes are inserted. 1 1: 3 bytes are inserted.	R/W
b31 to b18	—	Reserved	The read value is 0. The write value should be 0.	R/W

The RPADIR register sets insertion of padding for received data. The padding value is 00h.
Set the EDMR.SWR bit to 1 to reset before rewriting the RPADIR register.

37.2.20 Transmit Interrupt Setting Register (TRIMD)

Address(es): EDMAC0.TRIMD 000C 007Ch, EDMAC1.TRIMD 000C 027Ch, PTPEDMAC.TRIMD 000C 047Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	TIM	—	—	—	TIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TIS	Transmit Interrupt Enable	0: Transmit Interrupt is disabled. 1: Transmit Interrupt is enabled. Set the EESR.TWB flag to 1 in the mode selected by the TIM bit to notify an interrupt.	R/W
b3 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W
b4	TIM	Transmit Interrupt Mode	0: Transmission complete interrupt mode An interrupt occurs when a frame has been transmitted. 1: Write-back complete interrupt mode An interrupt occurs when write-back to the transmit descriptor has been completed while the TWBI bit is 1.	R/W
b31 to b5	—	Reserved	The read value is 0. The write value should be 0.	R/W

The TRIMD register sets transmit interrupt mode and enables/disables the transmit interrupt.
When the condition set in this register is satisfied, the EESR.TWB flag becomes 1, and an interrupt request is output when the EESIPR.TWBIP bit is 1.

37.2.21 Receive Buffer Write Address Register (RBWAR)

Address(es): EDMAC0.RBWAR 000C 00C8h, EDMAC1.RBWAR 000C 02C8h, PTPEDMAC.RBWAR 000C 04C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The RBWAR register indicates the last address that the EDMAC has written data to when writing to the receive buffer. Refer to the address indicated by the RBWAR register to recognize which address in the receive buffer the EDMAC is writing data to. Note that the address that the EDMAC is outputting to the receive buffer may not match the read value of the RBWAR register during data reception.
The RBWAR register is read only. Do not write to this register.

37.2.22 Receive Descriptor Fetch Address Register (RDFAR)

Address(es): EDMAC0.RDFAR 000C 00CCh, EDMAC1.RDFAR 000C 02CCh, PTPEDMAC.RDFAR 000C 04CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The RDFAR register indicates the start address of the last fetched receive descriptor when the EDMAC fetches descriptor information from the receive descriptor.
Refer to the address indicated by the RDFAR register to recognize which receive descriptor information the EDMAC is using for the current processing. Note that the address of the receive descriptor that the EDMAC fetches may not match the read value of the RDFAR register during data reception.
The RDFAR is read only. Do not write to this register.

37.2.23 Transmit Buffer Read Address Register (TBRAR)

Address(es): EDMAC0.TBRAR 000C 00D4h, EDMAC1.TBRAR 000C 02D4h, PTPEDMAC.TBRAR 000C 04D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The TBRAR register indicates the last address that the EDMAC has read data from when reading data from the transmit buffer.
Refer to the address indicated by the TBRAR register to recognize which address in the transmit buffer the EDMAC is reading from. Note that the address that the EDMAC is outputting to the transmit buffer may not match the read value of the TBRAR register.
The TBRAR register is read only. Do not write to this register.

37.2.24 Transmit Descriptor Fetch Address Register (TDFAR)

Address(es): EDMAC0.TDFAR 000C 00D8h, EDMAC1.TDFAR 000C 02D8h, PTPEDMAC.TDFAR 000C 04D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The TDFAR register indicates the start address of the last fetched transmit descriptor when the EDMAC fetches descriptor information from the transmit descriptor.

Refer to the address indicated by the TDFAR register to recognize which transmit descriptor information the EDMAC is using for the current processing. Note that the address of the transmit descriptor that the EDMAC fetches may not match the read value of the TDFAR register.

The TDFAR is read only. Do not write to this register.

37.3 Operation

The EDMAC transfers data according to the information written in the descriptor. Two types of descriptors are provided: transmit descriptor and receive descriptor. A descriptor includes the buffer size, address, and transmit/receive status. The EDMAC transmits or receives data continuously using sequentially arranged descriptors (descriptor list).

37.3.1 Descriptor Lists and Data Buffers

To transfer data using the EDMAC, create the transmit and receive descriptor lists on the memory, set the start address of the transmit descriptor list to the TDLAR register, and set the start address of the receive descriptor list to the RDLAR register. Also, transmit and receive buffers corresponding to the each descriptor are required.
Align the descriptor list on the appropriate address boundary according to the descriptor length set by the EDMR.DL[1:0] bits. The transmit buffer can be aligned on a long word boundary, word boundary, or byte boundary. When the valid transmit buffer size is 16 bytes or less, align it on a 32-byte boundary. Align the receive buffer on a 32-byte boundary. Set different addresses for the transmit and receive descriptors and buffers for EDMAC0, EDMAC1, and the PTPEDMAC.

37.3.1.1 Transmit Descriptor

Figure 37.3 shows the relation between a transmit descriptor and transmit buffer. A transmit descriptor consists of TD0 to TD2. The transmit frame and transmit buffer configuration can be selected from one buffer per frame (single-buffer frame transmission) or multiple buffers per frame (multi-buffer frame transmission) by setting the transmit descriptor.

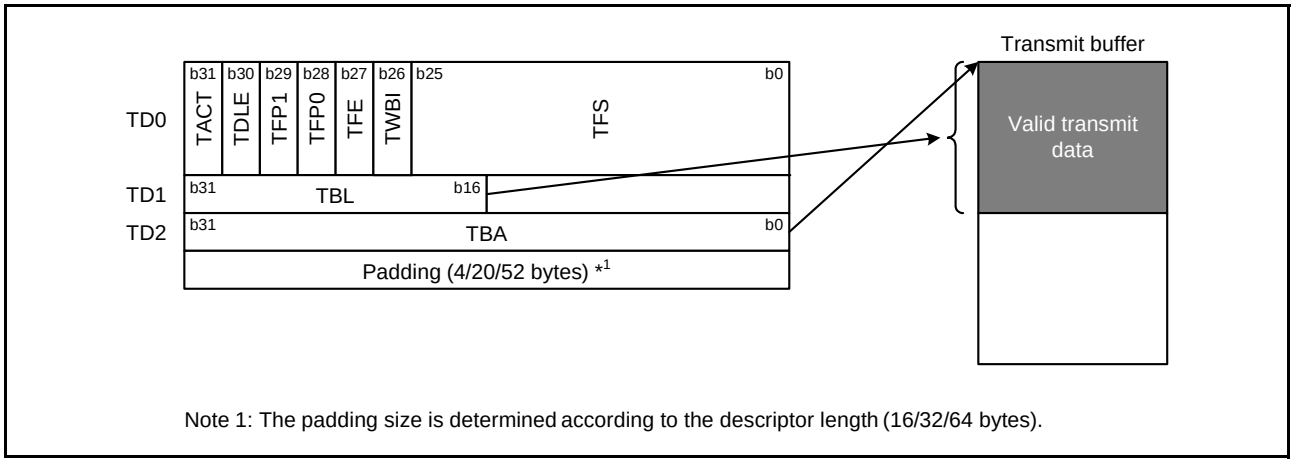


Figure 37.3 Relation between a Transmit Descriptor and Transmit Buffer

(1) Transmit Descriptor 0 (TD0)

Bit	Symbol	Bit Name	Description	R/W
b25 to b0	TFS	Transmit Frame Status	Set all bits to 0 when creating a descriptor. After write-back, the bits indicate the following: [for EDMACn] TFS25 to TFS9: Reserved TFS8: Transmit abort is detected (the value is equivalent to the EESR.TABT flag). TFS7 to TFS4: Reserved TFS3: No carrier is detected (the value is equivalent to the EESR.CND flag). TFS2: Loss of carrier is detected (the value is equivalent to the EESR.DLC flag). TFS1: Late collision during transmission is detected (the value is equivalent to the EESR.CD flag). TFS0: Transmit retry over (the value is equivalent to the EESR.TRO flag) When each bit becomes 1, it indicates that the corresponding error has occurred during frame transmission. When any of the TFS bits becomes 1, the TFE bit also becomes 1. When any bit from TFS3 to TFS0 becomes 1, the TFS8 bit also becomes 1. [for the PTPEDMAC] TFS25 to TFS9: Reserved TFS8: Transmit abort is detected (the value is equivalent to the EESR.TABT flag). TFS7 to TFS1: Reserved TFS0: The transmission source MAC address of the transmit frame data does not match the set value (the value is equivalent to the EESR.MACE flag). When each bit becomes 1, it indicates that the corresponding error has occurred during frame transmission. When any of the TFS bits becomes 1, the TFE bit also becomes 1. When TFS0 becomes 1, TFS8 also becomes 1.	R/W
b26	TWBI	Write-Back Complete Interrupt Enable	0: Interrupt does not occur when write-back to this descriptor has been completed. 1: Interrupt occurs when write-back to this descriptor has been completed.	R/W
b27	TFE	Transmit Frame Error	0: Frame transmission is successfully completed. 1: An error occurs during frame transmission (transmission aborted).	R/W
b29, b28	TFP[1:0]	Transmit Frame Position	b29 b28 0 0: Transmit buffer indicated by this descriptor is the middle of a transmit frame (frame information is incomplete). 0 1: Transmit buffer indicated by this descriptor is the end of a transmit frame (frame information is complete). 1 0: Transmit buffer indicated by this descriptor is the head of a transmit frame (frame information is incomplete.) 1 1: Transmit buffer indicated by this descriptor is all of a transmit frame (one buffer per frame).	R/W
b30	TDLE	Transmit Descriptor List End	When this bit is 1, it indicates that this descriptor is the last descriptor of the descriptor list.	R/W
b31	TACT	Transmit Descriptor Valid	This bit indicates that this descriptor is valid.	R/W

Note: Bits for write-back are underlined.

TD0 indicates the transmit frame settings, and the status after transmission.

TWBI Bit (Write-Back Complete Interrupt Enable)

This bit value is valid when the TRIMD.TIM bit is 1. Set the TRIMD.TIS bit and EESIPR.TWBIP bit to 1 to generate an interrupt request.

TFE Bit (Transmit Frame Error)

When the TFE bit is 1, it indicates that one or more of the TFS bits is 1.

TFP[1:0] Bits (Transmit Frame Position)

The TFP[1:0] bits indicate which part of a transmit frame corresponds to the transmit buffer indicated by this descriptor. The TFP[1:0] and TD1.TBL bit settings must be logically consistent in the previous and next descriptors.

TACT Bit (Transmit Descriptor Valid)

The TACT bit indicates that this descriptor is valid. The TACT bit is set to 1 by software. This bit becomes 0 when the transmit frame has been transferred or when the transmission is aborted.

(2) Transmit Descriptor 1 (TD1)

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	The read value is 0. The write value should be 0.	R/W
b31 to b16	TBL	Transmit Buffer Length	Set a valid byte length of the corresponding transmit buffer. Set a value equal to or greater than 1.	R/W

TD1 is used to set a valid byte length of the transmit buffer.

(3) Transmit Descriptor 2 (TD2)

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	TBA	Transmit Buffer Address	Set the start address of the transmit buffer. When the TD1.TBL bit value is 1 to 16 bytes, align it on a 32-byte boundary.	R/W

TD2 is used to set the start address of the transmit buffer.

37.3.1.2 Receive Descriptor

Figure 37.4 shows the relation between a receive descriptor and receive buffer. The receive frame and receive buffer configuration can be selected from one buffer per frame (single-buffer frame transmission) or multiple buffers per frame (multi-buffer frame transmission) by setting the receive descriptor. If the receive buffer length (RBL) is set to 0, operation indicated by the descriptor is not guaranteed.

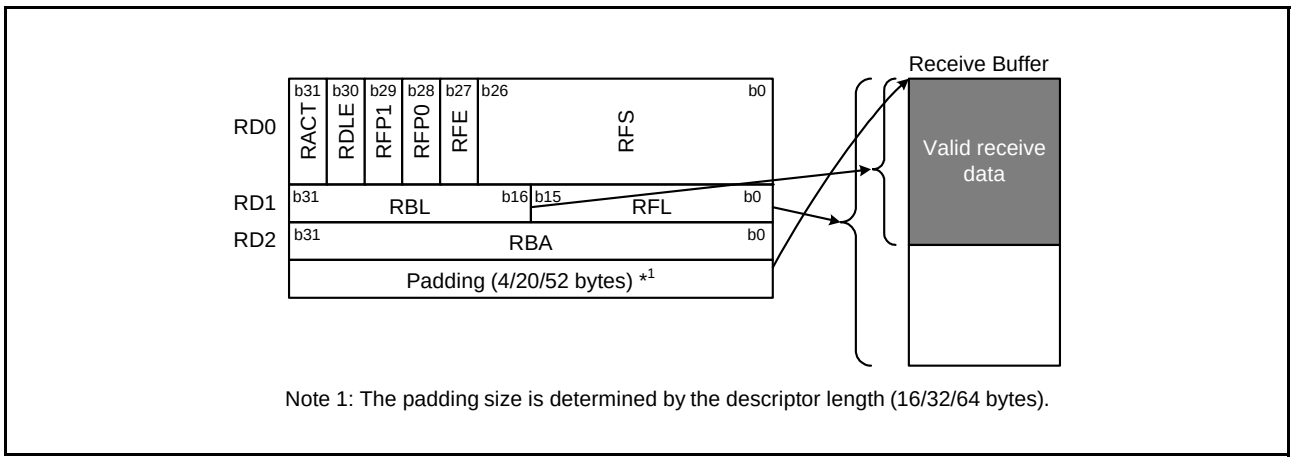


Figure 37.4 Relation between Receive Descriptor and Receive Buffer

(1) Receive Descriptor 0 (RD0)

Bit	Symbol	Bit Name	Description	R/W
b26 to b0	RFS	Receive Frame Status	Set all bits to 0 when creating a descriptor. After write-back, the bits indicate the following: [for EDMACn] RFS26 to RFS10: Reserved RFS9: Receive FIFO overflow (the value is equivalent to the EESR.RFOF flag) RFS8: Receive abort is detected (the value is equivalent to the EESR.RABT flag). RFS7: Multicast address frame is received (the value is equivalent to the EESR.RMAF flag). RFS6 and RFS5: Reserved RFS4: Alignment error is detected (the value is equivalent to the EESR.RRF flag). RFS3: Frame-too-long error (the value is equivalent to the EESR.RTLF flag) RFS2: Frame-too-short error (the value is equivalent to the EESR.RTSF flag) RFS1: PHY-LSI receive error (the value is equivalent to the EESR.PRE flag) RFS0: CRC error (the value is equivalent to the EESR.CERF flag) When each bit becomes 1, it indicates that the corresponding error has occurred during frame reception. When any of the RFS bits becomes 1, the RFE bit also becomes 1 (set the TRSCER register to select whether bits RFS7 and RFS4 are reflected in the RFE bit). When any bit from RFS3 to RFS0 becomes 1, the RFS8 bit also becomes 1. [for the PTPEDMAC] RFS26 to RFS10: Reserved RFS9: Receive FIFO overflow (the value is equivalent to the EESR.RFOF flag) RFS8: Reserved RFS7: Receive port (the value is equivalent to the EESR.RPORT flag) RFS4: PTPV2 packet is received (the value is equivalent to the EESR.PVER flag). The PTPEDMAC can receive only PTP packets. If a non-PTP packet is received, the packet is not transferred to the PTPEDMAC, and it is discarded. RFS3 to RFS0: Type of the received PTP message (the value is equivalent to the EESR.TYPE[3:0] flags) Each bit indicates the status of the received frame.	R/W
b27	RFE	Receive Frame Error	[for EDMACn] 0: No error has occurred in the received frame. 1: An error has occurred in the received frame. [for the PTPEDMAC] Reserved	R/W
b29, b28	RFP[1:0]	Receive Frame Position	b29 b28 0 0: Receive buffer indicated by this descriptor is the middle of a receive frame (frame information is incomplete). 0 1: Receive buffer indicated by this descriptor is the end of a receive frame (frame information is complete). 1 0: Receive buffer indicated by this descriptor is the head of a receive frame (frame information is incomplete). 1 1: Receive buffer indicated by this descriptor is all of a receive frame (one buffer per frame).	R/W
b30	RDLE	Receive Descriptor List End	When this bit is 1, it indicates that this descriptor is the last one of the descriptor list.	R/W
b31	RACT	Receive Descriptor Valid	Indicates that this descriptor is valid.	R/W

Note: Bits for write-back are underlined.

RD0 indicates the receive frame status.

RFE Bit (Receive Frame Error)

When the RFE bit is 1, it indicates that one or more of the RFS bits is 1 (set the TRSCER register to select whether bits RFS7 and RFS4 of EDMACn are reflected in the RFE bit).

RFP[1:0] Bits (Receive Frame Position)

The RFP[1:0] bits indicate which part of a receive frame corresponds to the receive buffer indicated by this descriptor.

RACT Bit (Receive Descriptor Valid)

The RACT bit indicates that this descriptor is valid. The RACT bit is set to 1 by software. This bit becomes 0 when all data has been transferred to the receive buffer indicated by RD2 or when the receive buffer becomes full.

(2) Receive Descriptor 1 (RD1)

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RFL	Receive Frame Length	These bits indicate the length (number of bytes) of the receive frame stored in the buffer. The number of bytes for padding set by the RPADIR register is not included. These bits are written back to the descriptor corresponding to the end of a frame.	R/W
b31 to b16	RBL	Receive Buffer Length	These bits indicate the byte length of the corresponding receive buffer. Set an integral multiple of 32 as the buffer length.	R/W

Note: Bits for write-back are underlined.

RD1 indicates the receive buffer length. When reception is completed, the receive frame length is written back.

(3) Receive Descriptor 2 (RD2)

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	RBA	Receive Buffer Address	These bits indicate the start address of the receive buffer. Align the buffer address on a 32-byte boundary.	R/W

RD2 indicates the start address of the receive buffer.

37.3.2 Transmission

When setting the EDTRR.TR bit to 1 while the ETHERCn.ECMR.TE bit is 1, the EDMAC reads the descriptor following the previously used descriptor in the transmit descriptor list (or the descriptor indicated by the TDLAR register after a reset). When the TACT bit is 1 in the transmit descriptor (TD0), the EDMAC sequentially reads transmit data from the start address of the transmit buffer indicated by transmit descriptor 2 (TD2) and transfers them to the ETHERC via the transmit FIFO. The ETHERC creates a transmit frame and starts transmission to the MII/RMII. When all data indicated by the TD1.TBL bit is transferred, write-back is performed according to the TD0.TFP[1:0] bits as follows:

- When the TD0.TFP[1:0] bits are 00b or 10b (frame information is incomplete), the TD0.TACT bit is written back.
- When the TD0.TFP[1:0] bits are 01b or 11b (frame information is complete), bits TD0.TACT, TD0.TFS, and TD0.TFE are written back.

When the TD0.TACT bit in the read descriptor is 1, the EDMAC continues reading descriptors and transmit frames. When the TD0.TACT bit in the read descriptor is 0, the EDMAC sets the EDTRR.TR bit to 0 and stops transmission.

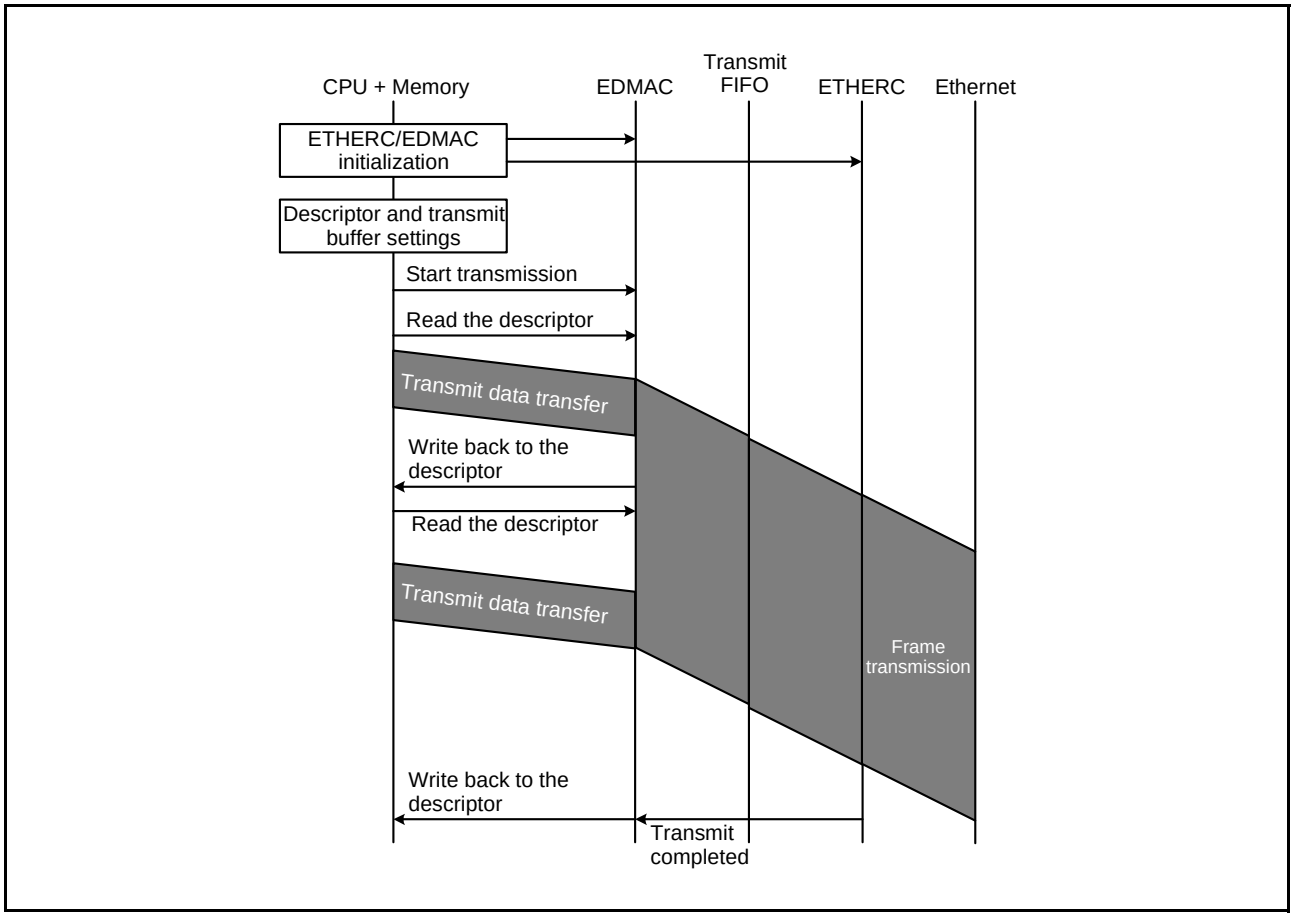


Figure 37.5 Example of Transmission Flow

37.3.3 Reception

When setting the EDRRR.RR bit to 1 while the ETHERCn.ECMR.RE bit is 1, the EDMAC reads the receive descriptor following the previously used descriptor (or descriptor indicated by the RDLAR register after a reset) and then waits for reception. While the RD0.RACT bit is 1, if the data stored to the receive FIFO is 32 bytes or more, or if the end byte of the frame is stored in the receive buffer, the EDMAC transfers data from the receive FIFO to the receive buffer indicated by receive descriptor 2 (RD2). If the data length of the received frame is longer than the buffer length set by the RBL bit in receive descriptor 1 (RD1), the EDMAC writes back 10b or 00b to the RD0.RFP[1:0] bits and 0 to the RD0.RACT bit when the receive buffer becomes full, and then the EDMAC reads the next data. After that, the EDMAC transfers data to another receive buffer. When the frame reception is completed or when the frame reception is aborted by an error, the EDMAC writes back 11b or 01b to the RD0.RFP[1:0] bits, 0 to the RD0.RACT bit, and the receive frame length to the RD1.RFL bit*1. When the RMCR.RNR bit is 1, the EDMAC reads the next descriptor and waits for reception. When the RNR bit is 0, the EDMAC sets the EDRRR.RR bit to 0 and stops reception.

Note 1. When the frame reception is aborted if the data stored in the receive FIFO is less than 16 bytes, the data is not transferred, but discarded. In this case, the corresponding error flag does not become 1.

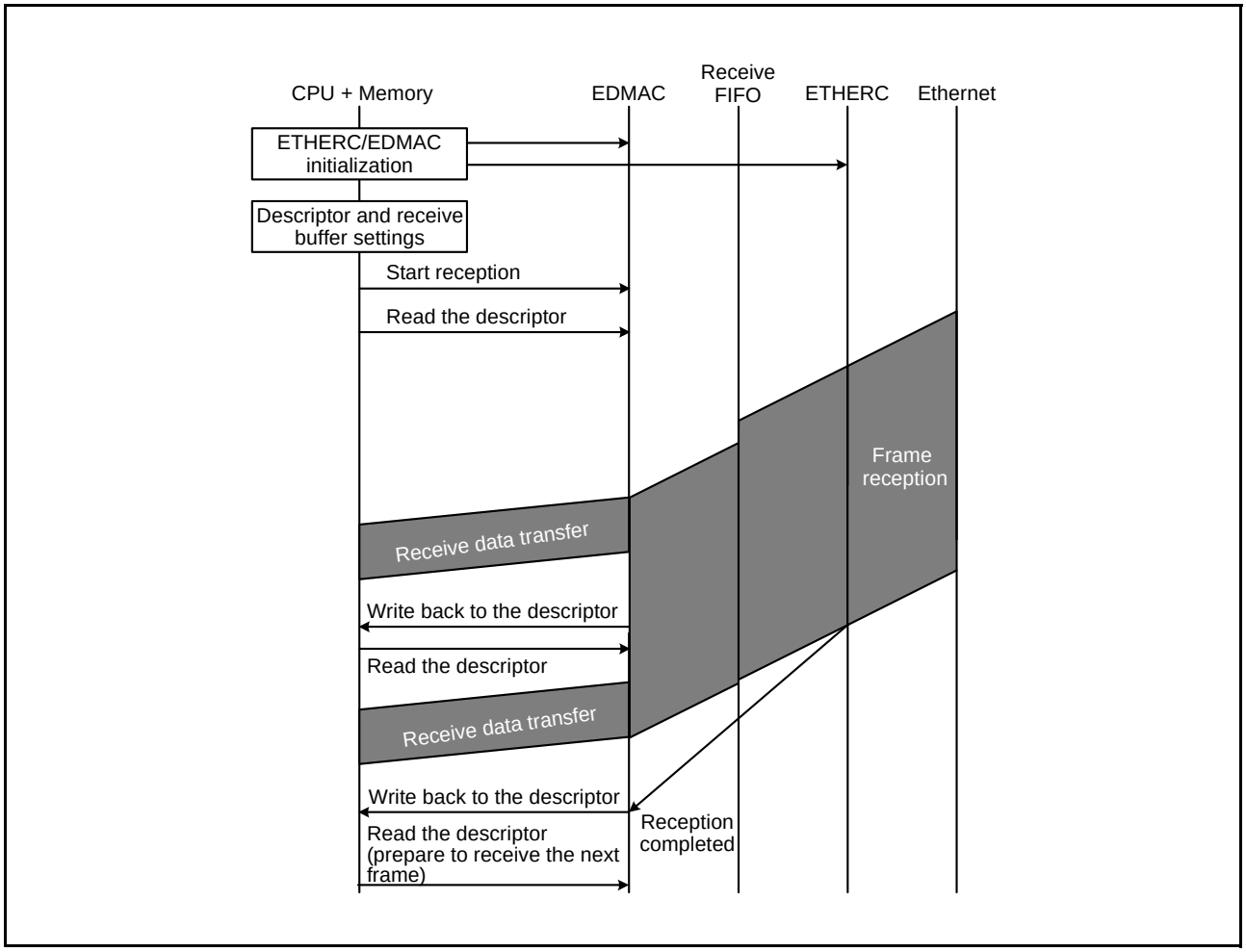


Figure 37.6 Example of Reception Flow

37.3.4 Multi-Buffer Frame Transmission

37.3.4.1 Error Processing While Transmitting a Multi-Buffer Frame

If an error occurs during multi-buffer frame transmission, the EDMAC performs the processing shown in Figure 37.7. In the figure, when the TACT bit of transmit descriptor 0 (TD0) is 0, the descriptor indicates that all data in the buffer has been successfully transmitted. When the TACT bit is 1, the descriptor indicates that data in the buffer has not yet been transmitted. If a frame transmit error *1 occurs in the head or middle of the frame while the TD0.TACT bit is 1, the EDMAC stops data transmission from the transmit FIFO and EDMAC data transfer, and sets the TD0.TACT bit to 0. After that, the EDMAC reads the next descriptor to see if the descriptor indicates the middle of the fame (TD0.TFP[1:0] bits are 00b) or the end of the frame (TD0.TFP[1:0] bits are 01b). When the descriptor indicates the middle of the frame, the EDMAC sets the TD0.TACT bit to 0 and reads the next descriptor. When the descriptor indicates the end of the frame, the EDMAC not only sets the TD0.TACT bit to 0, but also writes back to bits TD0.TFE and TD0.TFS. After an error occurs, data in the buffer is not transmitted until write-back to the descriptor for the end of the frame. When the corresponding transmit error interrupt is enabled in the EESIPR register, an interrupt request is generated immediately after write-back to the descriptor for the end of the frame.

Note 1. For EDMACn, a transmit retry-over condition, late collision, or loss of carrier is detected, or a carrier is not detected. For the PTPEDMAC, the MAC address does not match the set value.

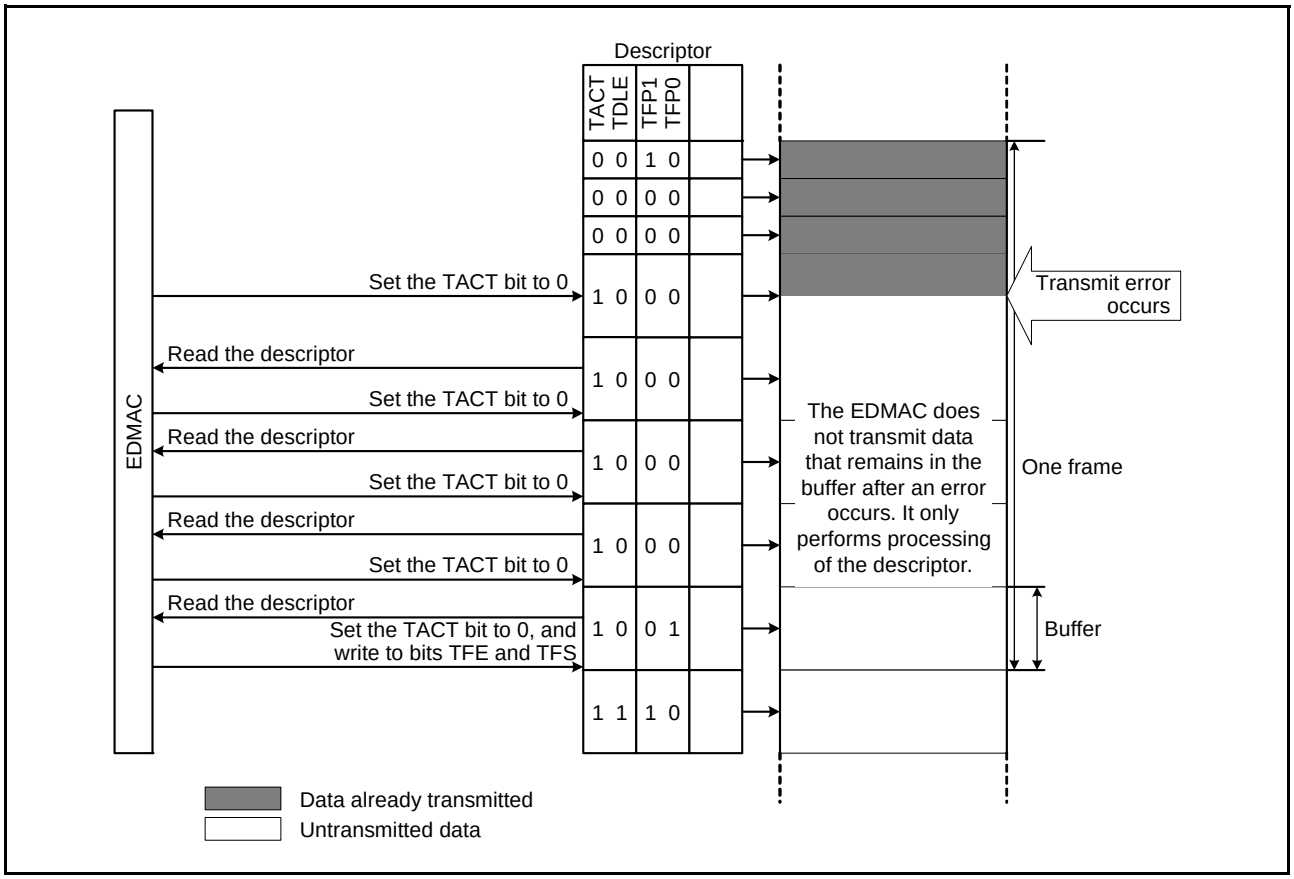


Figure 37.7 EDMAC Operation After a Transmit Error Occurs

37.3.4.2 Error Processing While Receiving a Multi-Buffer Frame

If an error occurs during multi-buffer frame reception, the EDMAC performs the processing shown in Figure 37.8. In the figure, when the RACT bit of receive descriptor 0 (RD0) is 0, the descriptor indicates that data has been successfully received in the buffer. When the RACT bit is 1, the descriptor indicates that data has not yet been received in the buffer. If a frame receive error *1 occurs, the EDMAC stops receiving data of the frame, but it transfers data that has already been stored in the receive FIFO to the receive buffer.

When the receive buffer becomes full during transfer, the EDMAC sets the RACT bit to 0 and the RFP[1:0] bits to 10b or 00b and reads the next descriptor. After all data in the receive FIFO has been transferred, the EDMAC writes back the status to the descriptor.

When the corresponding receive error interrupt is enabled in the EESIPR register, an interrupt request is generated immediately after write-back to the descriptor. When there is a request to receive a new frame, the EDMAC continues reception using the descriptor following the descriptor where the error occurred.

Note 1. For EDMACn, a CRC error, PHY-LSI receive error, frame-too-short error, frame-too-long error, or alignment error is detected. For the PTPEDMAC, a parity error is detected.

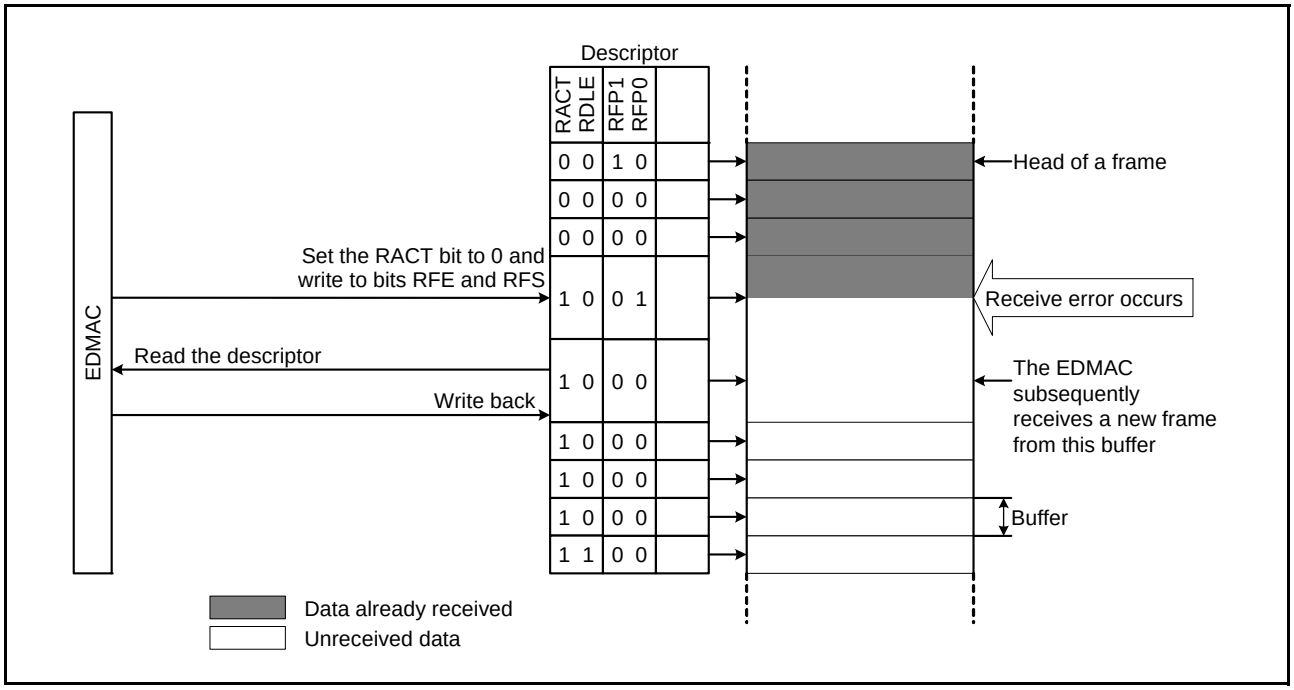


Figure 37.8 EDMAC Operation After a Receive Error Occurs

37.3.5 EDMAC Channel Priority

Priority of the three EDMAC channels (EDMAC0, EDMAC1, PTPEDMAC) is determined in a round-robin fashion. Each time transfer of one channel is completed, the channel becomes the lowest priority. This operation is shown in Figure 37.9. After a reset, the priority is EDMAC0 > EDMAC1 > PTPEDMAC.

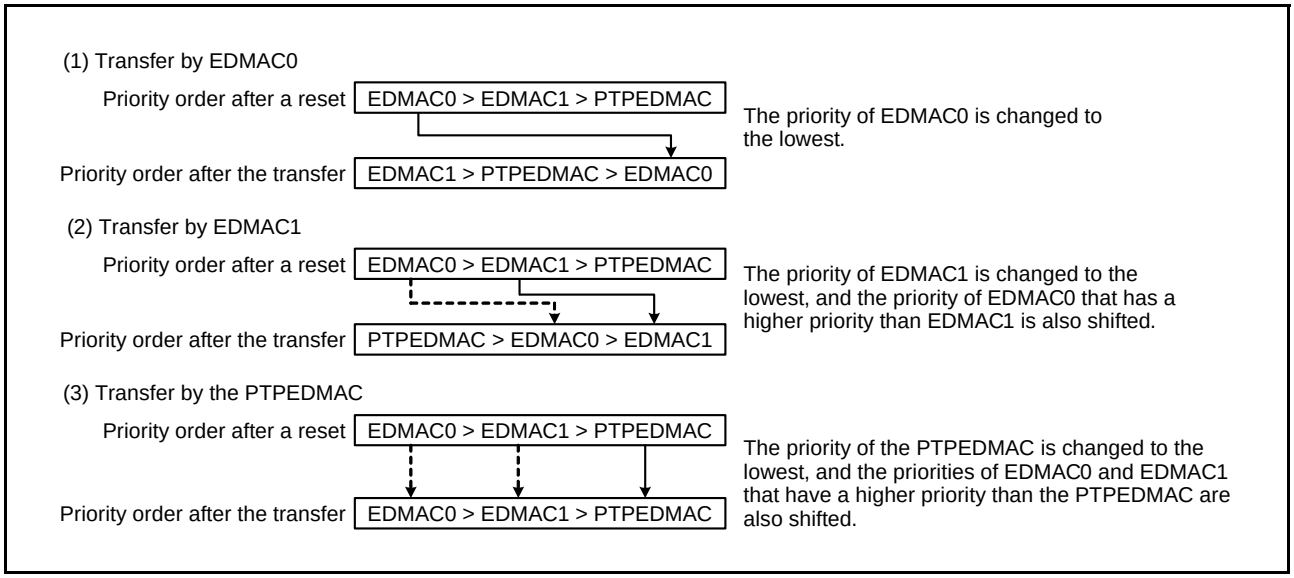


Figure 37.9 Operation in Round-Robin Fashion

Figure 37.10 shows change of the channel priority order when transfer requests are concurrently generated to EDMAC0 and the PTPEDMAC and a transfer request to the EDMAC1 while EDMAC0 is transferring data.

1. Transfer requests are concurrently generated to EDMAC0 and the PTPEDMAC.
2. Since EDMAC0 has a higher priority than the PTPEDMAC, EDMAC0 starts a transfer (PTPEDMAC waits to transfer).
3. A transfer request is generated to EDMAC1 during the transfer by EDMAC0 (EDMAC1 and the PTPEDMAC wait to transfer).
4. After EDMAC0 ends the transfer, the priority of EDMAC0 is changed to the lowest.
5. Since EDMAC1 has higher priority than the PTPEDMAC at this time, EDMAC1 starts a transfer (PTPEDMAC waits to transfer).
6. After EDMAC1 ends the transfer, the priority of EDMAC1 is changed to the lowest.
7. The PTPEDMAC starts a transfer.
8. After the PTPEDMAC ends the transfer, the priority of PTPEDMAC is changed to the lowest.

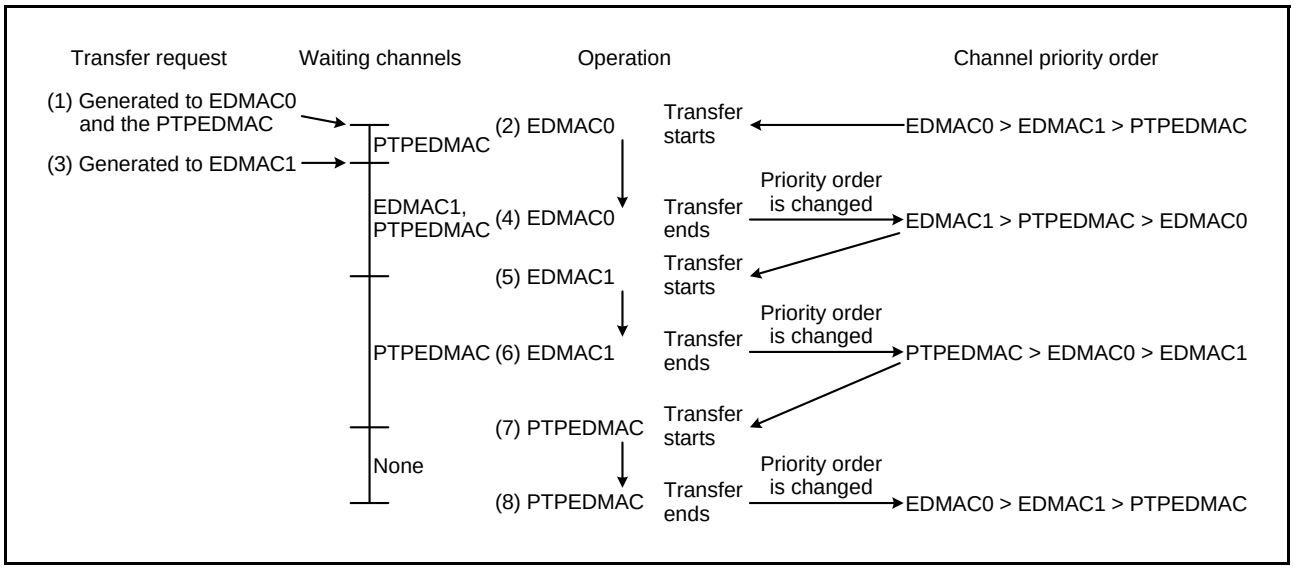


Figure 37.10 Channel Priority in Round-Robin Fashion

37.4 Interrupts

When any of the status flags in the EESR register becomes 1 while the corresponding interrupt request enable bit in the EESIPR register is 1, EDMACn outputs an EINTn interrupt request or the PTPEDMAC outputs a PINT interrupt request to the CPU.

37.5 Usage Notes

37.5.1 Setting the Module-Stop Function

Module stop control register B (MSTPCRB) is used to enable/disable the EDMAC operation. After a reset, the EDMAC is stopped. After exiting the module-stop state, registers can be accessed. Refer to section 11, Low Power Consumption for details.

37.5.2 Stopping the EDMAC during Operations

When stopping EDMAC operation by using a WAIT instruction or module-stop function while the EDMAC is running, confirm that bits EDTRR.TR and EDRRR.RR are 0. If the EDMAC is stopped while the EDTRR.TR or EDRRR.RR bit 1, the data for the frame that is being transmitted or received may not be complete, and EDMAC operation after exiting the sleep mode or module-stop state is not guaranteed.

37.5.3 Illegal Address Access Detection

When the illegal address access detection on the EDMAC operation is to be used, set the BSC.BEREN.IGAEN bit and ICU.IER02.IEN0 bit to 1.

If the BSC.BERSR1.MST[2:0] bits are 110b when a bus error interrupt occurs, set the EDMR.SWR bit to 1 (EDMAC and ETHERC are reset) and then configure the EDMAC and ETHERC again.

38. USB 2.0 FS Host/Function Module (USBb)

38.1 Overview

This MCU incorporates a USB 2.0 FS host/function module.
The USB module is a USB controller that is equipped to operate as a host controller or function controller. The module supports full-speed and low-speed (only for the host controller) transfer as defined in Universal Serial Bus (USB) Specification 2.0. The module has an internal USB transceiver and supports all of the transfer types defined in USB Specification 2.0.
The USB has buffer memory for data transfer, providing a maximum of 10 pipes. PIPE1 to PIPE9 can be assigned any endpoint number based on peripheral devices used for communication or based on the user system.
Table 38.1 shows the specifications of the USB.

Table 38.1 USB Specifications

Item	Specifications
Features	• USB Device Controller (UDC) and transceiver for USB 2.0 are incorporated. Host controller, function controller, and On-The-Go (OTG) are supported (one channel) • The host controller and the function controller can be switched by software. • Self-power mode or bus power mode can be selected.
	When the host controller is selected: • Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) are supported • Automatic scheduling for SOF and packet transmissions • Programmable intervals for isochronous and interrupt transfers • Multiple peripheral devices can be connected for communication via a one-stage hub.
	When the function controller is selected: • Full-speed transfer (12 Mbps)*1 is supported • Control transfer stage control function • Device state control function • Auto response function for SET_ADDRESS request • SOF interpolation function
Communication data transfer type	• Control transfer • Bulk transfer • Interrupt transfer • Isochronous transfer
Pipe configuration	• Buffer memory for USB communication is provided. • Up to 10 pipes can be selected (including the default control pipe). • PIPE1 to PIPE9 can be assigned any endpoint number.
	Transfer conditions that can be set for each pipe: • PIPE0: Control transfer, 64-byte single buffer • PIPE1 and PIPE2: 64-byte double buffer can be specified for bulk transfer 256-byte double buffer for isochronous transfer • PIPE3 to PIPE5: Bulk transfer, 64-byte double buffer • PIPE6 to PIPE9: Interrupt transfer, 64-byte single buffer
Others	• Reception ending function using transaction count • Function that changes the BRDY interrupt event notification timing (BFRE) • Function that automatically clears the buffer memory after the data for the pipe specified at the DnFIFO (n = 0, 1) port has been read (DCLRM) • NAK setting function for response PID generated by end of transfer (SHTNAK) • On-chip pull-up and pull-down resistors of D+/D–
Low power consumption function	Module stop state can be set.

Note 1. When the function controller is selected, low-speed transfer (1.5 Mbps) is not supported.

Figure 38.1 shows a block diagram of the USB.

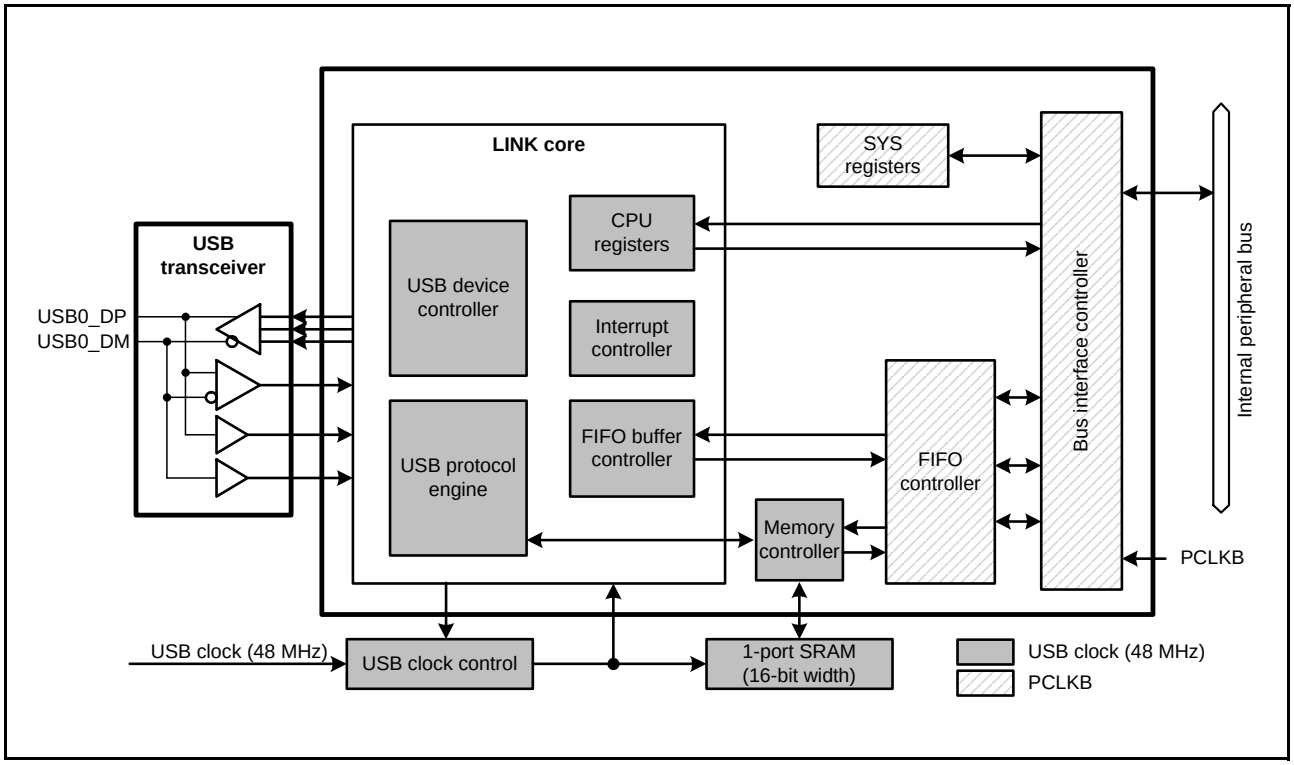


Figure 38.1 USB Block Diagram

Table 38.2 lists the I/O pins of the USB.

Table 38.2 USB Pin Configuration

Port	Pin Name	I/O	Function
USB	USB0_DP	I/O	D+ I/O pin of the USB on-chip transceiver This pin should be connected to the D+ pin of the USB bus.
	USB0_DM	I/O	D– I/O pin of the USB on-chip transceiver This pin should be connected to the D– pin of the USB bus.
	USB0_VBUS	Input	USB cable connection monitor pin This pin should be connected to VBUS of the USB bus. The VBUS pin status (connected or disconnected) can be detected when the USB module is operating as a function controller.
	USB0_EXICEN	Output	Low-power control signal for external power supply (OTG) chip
	USB0_VBUSEN	Output	VBUS (5 V) supply enable signal for external power supply chip
	USB0_OVRCURA USB0_OVRCURB	Input	External overcurrent detection signals should be connected to these pins. VBUS comparator signals should be connected to these pins when the OTG power supply chip is connected.
	USB0_ID	Input	MicroAB connector ID input signal should be connected to this pin during operation in OTG mode.
Common	VCC_USB	Input	USB power supply pin
	VSS_USB	Input	USB ground pin

38.2 Register Descriptions

38.2.1 System Configuration Control Register (SYSCFG)

Address(es): 000A 0000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKE	—	—	—	DCFM	DRPD	DPRPU	—	—	—	USBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	USBE	USB Operation Enable	0: USB operation is disabled. 1: USB operation is enabled.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DPRPU	D+ Line Resistor Control	0: Pulling up the line is disabled. 1: Pulling up the line is enabled.	R/W
b5	DRPD	D+/D– Line Resistor Control	0: Pulling down the lines is disabled. 1: Pulling down the lines is enabled.	R/W
b6	DCFM	Controller Function Select	0: Function controller is selected. 1: Host controller is selected.	R/W
b9 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10	SCKE	USB Clock Enable*1	0: Stops supplying the clock signal to the USB. 1: Enables supplying the clock signal to the USB.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. After writing 1 to the SCKE bit, read it and confirm it is set to 1.

USBE Bit (USB Operation Enable)

The USBE bit enables or disables operation of the USB.

Modifying the USBE bit from 1 to 0 initializes the register bits listed in Table 38.3.

This bit should be modified while the SCKE bit is 1.

When the host controller is selected, this bit should be set to 1 after setting the DRPD bit to 1, eliminating SYSSTS0.LNST[1:0] flag chattering, and confirming that the USB bus state is stabilized.

Table 38.3 Registers Initialized by Writing 0 to the SYSCFG.USBE Bit

Selected Function	Register	Bit	Remarks
Function controller	SYSSTS0	LNST[1:0]	The value is retained when the host controller is selected.
	DVSTCTR0	RHST[2:0]	
	INTSTS0	DVSQ[2:0]	The value is retained when the host controller is selected.
	USBADDR	USBADDR[6:0]	The value is retained when the host controller is selected.
	USBREQ	BREQUEST[7:0], BMREQUESTTYPE[7:0]	The value is retained when the host controller is selected.
	USBVAL	WVALUE[15:0]	The value is retained when the host controller is selected.
	USBINDX	WINDEX[15:0]	The value is retained when the host controller is selected.
	USBLENG	WLENGTH[15:0]	The value is retained when the host controller is selected.
Host controller	DVSTCTR0	RHST[2:0]	
	FRMNUM	FRNM[10:0]	The value is retained when the function controller is selected.

DPRPU Bit (D+ Line Resistor Control)

The DPRPU bit enables or disables pulling up the D+ line when the function controller is selected.
When the DPRPU bit is set to 1 while the function controller is selected, the bit forces a pull-up of the D+ line to notify the USB host of connection. Modifying the DPRPU bit from 1 to 0 allows the USB to release the D+ line, thus notifying the USB host of disconnection.
This bit should be set to 1 if the function controller is selected, and should be set to 0 if the host controller is selected.

DRPD Bit (D+/D– Line Resistor Control)

The DRPD bit enables or disables pulling down D+ and D– lines when the host controller is selected.
This bit should be set to 1 if the host controller is selected, and should be set to 0 if the function controller is selected.

DCFM Bit (Controller Function Select)

The DCFM bit selects the function of the USB.
This bit should be modified when the DPRPU and DRPD bits are all 0.

SCKE Bit (USB Clock Enable)

The SCKE bit stops or enables supplying 48-MHz clock signals to the USB.
When this bit is 0, only SYSCFG can be read from and written to; the other registers related to the USB cannot be read from or written to.

38.2.2 System Configuration Status Register 0 (SYSSTS0)

Address(es): 000A 0004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVCMON[1:0]		—	—	—	—	—	—	—	HTACT	SOFEA	—	—	IDMON	LNST[1:0]	
Value after reset:	0*1	0*1	0	0	0	0	0	0	0	0	0	0	0	0*1	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LNST[1:0]	USB Data Line Status Monitor Flag	Refer to Table 38.4.	R
b2	IDMON	External ID0 Input Pin Monitor Flag	0: USB0_ID pin is low 1: USB0_ID pin is high	R
b4, b3	—	Reserved	These bits are read as 0 and cannot be modified.	R
b5	SOFEA	SOF Active Monitor Flag When the Host Controller is Selected	Indicates the SOF output status. 0: SOF output is stopped. 1: SOF output is operating.	R
b6	HTACT	USB Host Sequencer Status Monitor Flag	0: Host sequencer of the USB is completely stopped. 1: Host sequencer of the USB is not completely stopped.	R
b13 to b7	—	Reserved	These bits are read as 0 and cannot be modified.	R
b15, b14	OVCMON[1:0]	External USB0_OVRCURA/ USB0_OVRCURB Input Pin Monitor Flag	The OVCMON[1] flag indicates the status of the USB0_OVRCURA pin. The OVCMON[0] flag indicates the status of the USB0_OVRCURB pin.	R

Note 1. Depends on the status of the USB0_OVRCURA/USB0_OVRCURB and USB0_ID pins.

LNST[1:0] Flags (USB Data Line Status Monitor Flag)

The LNST[1:0] flags indicate the state of the USB data lines (D+ and D– lines). Refer to Table 38.4.

The LNST[1:0] flags should be read after the connection processing (SYSCFG.DPRPU bit = 1) when the function controller is selected; whereas after enabling pull-down of the lines (SYSCFG.DRPD bit = 1) when the host controller is selected.

SOFEA Flag (SOF Active Monitor Flag When the Host Controller is Selected)

When the USB module is suspended while host mode is used, this bit can be used to confirm whether the last SOF has been output after DVSTCTR0.UACT is set to 0.

Confirm that both the HTACT and SOFEA flags are 0 before stopping the USB module while SYSCFG.USBE is 0 and setting the SYSCFG.SCKE bit to 0 to stop the clock during host mode communication.

HTACT Flag (USB Host Sequencer Status Monitor Flag)

The HTACT flag is 0 when the host sequencer of the USB is completely stopped.

Confirm that the HTACT flag is 0 before setting this controller to the USB suspended state while DVSTCTR0.UACT bit is 0 and setting the SCKE bit to 0 to stop the clock during host mode communication.

OVCMON[1:0] Flags (External USB0_OVRCURA/ USB0_OVRCURB Input Pin Monitor Flag)

The OVCMON[1:0] flags indicate the status of overcurrent from an external power supply chip.

Table 38.4 Status of USB Data Bus Lines (D+ Line, D– Line)

LNST[1:0] Flags	During Low-Speed Operation (Only in Host Controller Operation)	During Full-Speed Operation
00b	SE0	SE0
01b	K-State	J-State
10b	J-State	K-State
11b	SE1	SE1

38.2.3 Device State Control Register 0 (DVSTCTR0)

Address(es): 000A 0008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	HNPBTOA	EXICEN	VBUSEN	WKUP	RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RHST[2:0]	USB Bus Reset Status Flag	• When the host controller is selected b2 b0 0 0 0: Communication speed not determined (powered state or no connection) 1 x x: USB bus reset in progress 0 0 1: Low-speed connection 0 1 0: Full-speed connection • When the function controller is selected b2 b0 0 0 0: Communication speed not determined 0 0 1: USB bus reset in progress 0 1 0: USB bus reset in progress or full-speed connection	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	UACT	USB Bus Enable	0: Downstream port is disabled (SOF transmission is disabled). 1: Downstream port is enabled (SOF transmission is enabled).	R/W
b5	RESUME	Resume Output	0: Resume signal is not output. 1: Resume signal is output.	R/W
b6	USBRST	USB Bus Reset Output	0: USB bus reset signal is not output. 1: USB bus reset signal is output.	R/W
b7	RWUPE	Wakeup Detection Enable	0: Downstream port wakeup is disabled. 1: Downstream port wakeup is enabled.	R/W
b8	WKUP	Wakeup Output	0: Remote wakeup signal is not output. 1: Remote wakeup signal is output.	R/W
b9	VBUSEN	USB0_VBUSEN Output Pin Control	0: External USB0_VBUSEN pin outputs low. 1: External USB0_VBUSEN pin outputs high.	R/W
b10	EXICEN	USB0_EXICEN Output Pin Control	0: External USB0_EXICEN pin outputs low. 1: External USB0_EXICEN pin outputs high.	R/W
b11	HNPBTOA	Host Negotiation Protocol (HNP) Control	This bit is used when switching from device B to device A while in OTG mode. If the HNPBTOA bit is 1, the internal function control keeps the suspended state until the HNP processing ends even though SYSCFG.DPRPU = 0 or SYSCFG.DCFM = 1.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don't care

RHST[2:0] Flags (USB Bus Reset Status Flag)

The RHST[2:0] flags indicate the status of the USB bus reset.
When the host controller is selected, the RHST[2:0] flags indicate 100b after the USBRST bit has been set to 1 by software.
The USB fixes the value of the RHST[2:0] flags when 0 is written to the USBRST bit by software and the USB completes SE0 driving.
When the function controller is selected, the RHST[2:0] flags indicate 010b (connection while DPRPU = 1) when the USB detects the USB bus reset, and a DVST interrupt is generated.

UACT Bit (USB Bus Enable)

The UACT bit enables operation of the USB bus (controls the SOF packet transmission to the USB bus) when the host controller is selected.
With this bit set to 1, the USB puts the USB port to the USB-bus enabled state and performs SOF output and data transmission and reception.
This module starts outputting SOF packets within one frame after 1 has been written to the UACT bit by software.
With this bit set to 0, the USB enters the idle state after outputting SOF packets.
The USB sets the UACT bit to 0 on any of the following conditions.

- A DTCH interrupt is detected during communication (while UACT = 1).
- An EOFERR interrupt is detected during communication (while UACT = 1).

Writing 1 to this bit should be done at the end of the USB reset processing (writing 0 to the USBRST bit) or at the end of the resume processing from the suspended state (writing 0 to the RESUME bit).
This bit should be set to 0 if the function controller is selected.

RESUME Bit (Resume Output)

The RESUME bit controls the resume signal output when the host controller is selected.
Setting the RESUME bit to 1 allows the USB to drive the port to the K-state and output the resume signal.
The USB sets the RESUME bit to 1 on detecting the remote wakeup signal while RWUPE is 1 in the USB suspended state.
The USB continues outputting K-state while the RESUME bit = 1 (until the RESUME bit is set to 0 by software). The RESUME bit should be 1 (= resume period) for the time defined by USB Specification 2.0.
This bit should be set to 1 in the suspended state.
Write 1 to the UACT bit simultaneously with the end of the resume processing (writing 0 to the RESUME bit).
This bit should be set to 0 if the function controller is selected.

USBRST Bit (USB Bus Reset Output)

The USBRST bit controls the USB bus reset signal output when the host controller is selected.
When the host controller is selected, setting this bit to 1 allows the USB to drive SE0 of the USB port to reset the USB bus.
The USB continues outputting SE0 while USBRST = 1 (until the USBRST bit is set to 1 by software). The USBRST bit should be 1 (= USB bus reset period) for the time defined by USB Specification 2.0.
Writing 1 to this bit during communication (the UACT bit = 1) or during the resume processing (the RESUME bit = 1) prevents the USB from starting the USB bus reset processing until both the UACT and RESUME bits become 0.
Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (writing 0 to the USBRST bit).
This bit should be set to 0 if the function controller is selected.

RWUPE Bit (Wakeup Detection Enable)

The RWUPE bit enables or disables the downstream port peripheral device to use the remote wakeup function (resume signal output) when the host controller is selected.

With this bit set to 1, on detecting the remote wakeup signal, the USB detects the resume signal (K-state for 2.5 μs) from the downstream port device and performs the resume processing (drives the port to the K-state).

With this bit set to 0, the USB ignores the detected remote wakeup signal (K-state) from the peripheral device connected to the USB port.

While the RWUPE bit is 1, the internal clock should not be stopped even in the suspended state (SYSCFG.SCKE bit should be set to 1).

This bit should be set to 0 if the function controller is selected.

WKUP Bit (Wakeup Output)

The WKUP bit enables or disables outputting the remote wakeup signal (resume signal) to the USB bus when the function controller is selected.

The USB controls the output time of a remote wakeup signal. When this bit is set to 1, the USB sets this bit to 0 after outputting the 10-ms K-state.

According to USB Specification 2.0, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. If the USB writes 1 to this bit right after detection of the suspended state, the K-state will be output after 2 ms.

Do not write 1 to this bit, unless the device state is in the suspended state (INTSTS0.DVSQ[2:0] flags = 1xxb) and the USB host enables the remote wakeup signal. When this bit is set to 1, the internal clock must not be stopped even in the suspended state (write 1 to this bit while the SYSCFG.SCKE bit = 1).

This bit should be set to 0 if the host controller is selected.

HNPBTOA Bit (Host Negotiation Protocol (HNP) Control)

The HNPBTOA bit is used when switching from device B to device A while in OTG mode. If the HNPBTOA bit is 1, the internal function control keeps the suspended state until the HNP processing ends even though the SYSCFG.DPRPU bit = 0 or SYSCFG.DCFM = 1 is set. Even if the falling edge of the D+ signal is detected at this time, no resume (RESM) interrupt is generated.

After this bit is set to 1, write 0 to this bit by software to terminate the HNP processing when connection to the host (pull-up on the target side) or timeout of the HNP processing is detected.

38.2.4 CFIFO Port Register (CFIFO)
D0FIFO Port Register (D0FIFO)
D1FIFO Port Register (D1FIFO)

(1) When the MBW bit is 1

Address(es): CFIFO 000A 0014h, D0FIFO 000A 0018h, D1FIFO 000A 001Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	FIFO Port	This port is used for reading receive data from the FIFO buffer and writing transmit data to the FIFO buffer.	R/W

(2) When the MBW bit is 0

Address(es): CFIFO 000A 0014h, D0FIFO 000A 0018h, D1FIFO 000A 001Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	L[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	L[7:0]	FIFO Port	This port is used for reading receive data from the FIFO buffer and writing transmit data to the FIFO buffer.	R/W

There are three FIFO ports: CFIFO, D0FIFO, and D1FIFO. Each FIFO port is configured of a port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer memory and writing of data to the FIFO buffer memory, a port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that is used to select the pipe assigned to the FIFO port, and a port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR).
Each FIFO port has the following precautions.

- The FIFO buffer for DCP (control transfer) should be accessed through the CFIFO port.
- Accessing a FIFO buffer using DMA/DTC transfer should be performed through the D0FIFO or D1FIFO port.
- The D0FIFO and D1FIFO ports can also be accessed by the CPU.
- When using functions specific to a FIFO port, the pipe number (selected pipe) specified by the CURPIPE[3:0] bits in the port select register cannot be changed (when the DMA/DTC transfer function is used, etc.).
- The same pipe should not be assigned to two or more FIFO ports.
- There are two FIFO buffer states: the access right for a FIFO buffer can be on the CPU side or on the Serial Interface Engine (SIE) side. When the access right for a FIFO buffer is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

FIFO Port Bit

Accessing the FIFO port bits allows reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer.

Each FIFO port register can be accessed only while the FRDY flag in each FIFO port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in a FIFO port register depend on the settings of the corresponding MBW bit of the FIFO port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL).

When the MBW bit is 1 (16-bit width), the data arrangement may differ from the data arrangement on the RAM depending on the value of the MDE.MDE[2:0] bits and the setting of the BIGEND bit (CFIFOSEL.BIGEND, D0FIFOSEL.BIGEND, or D1FIFOSEL.BIGEND). Table 38.5 lists the endian operation in 16-bit access. Note that if the total number of transmit data bytes is odd, access the L[7:0] bits in bytes when writing the last data.

When the MBW bit is 0 (8-bit width), access the L[7:0] bits in bytes.

Table 38.5 Endian Operation in 16-Bit Access

MDE.MDE[2:0] bits	CFIFOSEL.BIGEND Bit D0FIFOSEL.BIGEND Bit D1FIFOSEL.BIGEND Bit	Bits 15 to 8	Bits 7 to 0	Remarks
000b (big endian)	0 (little endian)	Data in address N + 1	Data in address N	Bytes reversed
	1 (big endian)	Data in address N	Data in address N + 1	
111b (little endian)	0 (little endian)	Data in address N + 1	Data in address N	
	1 (big endian)	Data in address N	Data in address N + 1	Bytes reversed

38.2.5 CFIFO Port Select Register (CFIFOSEL)
D0FIFO Port Select Register (D0FIFOSEL)
D1FIFO Port Select Register (D1FIFOSEL)

- CFIFOSEL

Address(es): 000A 0020h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCNT	REW	—	—	—	MBW	—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE [3:0]	CFIFO Port Access Pipe Specification	b3 b0 0 0 0 0: DCP (Default control pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Settings other than above are prohibited.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	ISEL	CFIFO Port Access Direction When DCP is Selected	0: Reading from the buffer memory is selected. 1: Writing to the buffer memory is selected.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BIGEND	CFIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	MBW	CFIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	REW	Buffer Pointer Rewind	0: The buffer pointer is not rewound. 1: The buffer pointer is rewound.	R/W*1
b15	RCNT	Read Count Mode	0: The DTLN[8:0] flags (CFIFOCTR.DTLN[8:0], D0FIFOCTR.DTLN[8:0], D1FIFOCTR.DTLN[8:0]) are cleared when all of the receive data has been read from the CFIFO. (In double buffer mode, the DTLN[8:0] bit value is cleared when all the data has been read from only a single plane.) 1: The DTLN[8:0] flags are decremented each time the receive data is read from the CFIFO.	R/W

Note 1. Only 0 can be read.

The same pipe should not be specified by the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

The pipe number should not be changed while the DMA/DTC transfer is enabled.

CURPIPE[3:0] Bits (CFIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number using which data is read or written through the CFIFO port.
After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.
Do not set the same pipe number to the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.
Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.

ISEL Bit (CFIFO Port Access Direction When DCP is Selected)

After writing to the ISEL bit with the DCP being a selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process.
Set this bit and the CURPIPE[3:0] bits simultaneously.

MBW Bit (CFIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the CFIFO port.
When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read.
When the selected pipe is in the receiving direction, set the CURPIPE[3:0] bits to a different value once, and then set these bits and the MBW bit simultaneously. For the procedure for modifying the CURPIPE[3:0] bits, follow the description of these bits.
When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory.
An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

REW Bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.
When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed).
Do not set the REW bit to 1 simultaneously with modifying the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY flag is 1.
To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

- D0FIFOSEL, D1FIFOSEL

Address(es): D0FIFOSEL 000A 0028h, D1FIFOSEL 000A 002Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCNT	REW	DCLRM	DREQE	—	MBW	—	BIGEND	—	—	—	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE[3:0]	FIFO Port Access Pipe Specification	b3 b0 0 0 0 0: DCP (Default control pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Settings other than above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BIGEND	FIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	MBW	FIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	DREQE	DMA/DTC Transfer Request Enable	0: DMA/DTC transfer request is disabled. 1: DMA/DTC transfer request is enabled.	R/W
b13	DCLRM	Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read	0: Auto buffer clear mode is disabled. 1: Auto buffer clear mode is enabled.	R/W
b14	REW	Buffer Pointer Rewind	0: The buffer pointer is not rewound. 1: The buffer pointer is rewound.	R/W*1
b15	RCNT	Read Count Mode	0: The DTLN[8:0] flags (CFIFOCTR.DTLN[8:0], D0FIFOCTR.DTLN[8:0], D1FIFOCTR.DTLN[8:0]) are cleared when all of the receive data has been read from the DnFIFO. (In double buffer mode, the DTLN bit Value is cleared when all the data has been read from only a single plane.) 1: The DTLN[8:0] flags are decremented each time the receive data is read from the DnFIFO. (n = 0, 1)	R/W

Note 1. Only 0 can be read.

The same pipe should not be specified by the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

The pipe number should not be changed while the DMA/DTC transfer is enabled.

CURPIPE[3:0] Bits (FIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number using which data is read or written through the D0FIFO port or D1FIFO port.
After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.
Do not set the same pipe number to the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.
Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.

MBW Bit (FIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the D0FIFO port or D1FIFO port.
When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read.
When the selected pipe is in the receiving direction, set the CURPIPE[3:0] bits to a different value once, and then set these bits and the MBW bit simultaneously. For the procedure for modifying the CURPIPE[3:0] bits, follow the description of these bits.
When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory.
An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

DREQE Bit (DMA/DTC Transfer Request Enable)

The DREQE bit enables or disables the DMA/DTC transfer request to be issued.
Before setting the DREQE bit to 1 to enable the DMA/DTC transfer request to be issued, set the CURPIPE[3:0] bits.
When modifying the setting of the CURPIPE[3:0] bits, set this bit to 0 first.

DCLRM Bit (Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read)

The DCLRM bit enables or disables the buffer memory to be cleared automatically after data has been read using the selected pipe.
With this bit set to 1, the USB sets the BCLR bit to 1 for the FIFO buffer of the selected pipe on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or on receiving a short packet and reading the data while the PIPECFG.BFRE bit is 1.
When using the USB with the SOFCFG.BRDYM bit set to 1, set this bit to 0.

REW Bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.
When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed).
Do not set the REW bit to 1 simultaneously with modifying the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY flag is 1.
To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

RCNT Bit (Read Count Mode)

The RCNT bit specifies the read mode for the value in the CFIFOCTR.DTLN bit.
When accessing DnFIFO with the PIPECFG.BFRE bit set to 1, set the RCNT bit to 0.

38.2.6 CFIFO Port Control Register (CFIFOCTR)
D0FIFO Port Control Register (D0FIFOCTR)
D1FIFO Port Control Register (D1FIFOCTR)

Address(es): CFIFOCTR 000A 0022h, D0FIFOCTR 000A 002Ah, D1FIFOCTR 000A 002Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	DTLN[8:0]	Receive Data Length Flag	Indicate the length of the receive data. These bits indicate different values depending on the setting of the RCNT bit in the port select register. For details, refer to the description on the DTLN[8:0] flags shown below.	R
b12 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	FRDY	FIFO Port Ready Flag	0: FIFO port access is disabled. 1: FIFO port access is enabled.	R
b14	BCLR	CPU Buffer Clear	0: Does not operate. 1: Clears the buffer memory on the CPU side.	R/W*1
b15	BVAL	Buffer Memory Valid	0: Invalid 1: Writing ended	R/W

Note 1. Only 0 can be read.

Registers CFIFOCTR, D0FIFOCTR, and D1FIFOCTR correspond to CFIFO, D0FIFO, and D1FIFO, respectively.

DTLN[8:0] Flags (Receive Data Length Flag)

The DTLN[8:0] flags indicate the length of the receive data.
While the FIFO buffer is being read, the DTLN[8:0] flags indicate different values depending on the DnFIFOSEL.RCNT bit (n = 0, 1) value as described below.

- RCNT = 0
  The USB sets the DTLN[8:0] flags to indicate the length of the receive data until the CPU or DMAC/DTC has read all the received data from a single FIFO buffer plane.
  While the PIPECFG.BFRE bit = 1, these bits retain the length of the receive data until the BCLR bit is set to 1 even after all the data has been read.
- RCNT = 1
  The USB decrements the value indicated by the DTLN[8:0] flags each time data is read from the FIFO buffer. (The value is decremented by one when the MBW bit is 0, and by two when the MBW bit is 1.)
  The USB sets these bits to 0 when all the data has been read from one FIFO buffer plane. However, in double buffer mode, if data has been received in one FIFO buffer plane before all the data has been read from the other plane, the USB sets these bits to indicate the length of the receive data in the former plane when all the data has been read from the latter plane.

FRDY Flag (FIFO Port Ready Flag)

The FRDY flag indicates whether the FIFO port can be accessed by the CPU or DMAC/DTC.
In the following cases, the USB sets the FRDY flag to 1 but data cannot be read via the FIFO port because there is no data to be read. In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty.
- A short packet is received and the data is completely read while the PIPECFG.BFRE bit = 1.

BCLR Bit (CPU Buffer Clear)

The BCLR bit should be set to 1 to clear the FIFO buffer on the CPU side for the selected pipe.
When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USB clears only one plane of the FIFO buffer even when both planes are read-enabled.
When the selected pipe is the DCP, setting the BCLR bit to 1 allows the USB to clear the FIFO buffer regardless of whether the FIFO buffer is on the CPU side or SIE side. To clear the buffer on the SIE side, set the DCPCTR.PID[1:0] bits for the DCP to 00b (NAK) before setting the BCLR bit to 1.
When the selected pipe is in the transmitting direction, if 1 is written to the BVAL bit and the BCLR bit simultaneously, the USB clears the data that has been written before it, enabling transmission of a zero-length packet.
When the selected pipe is not the DCP, writing 1 to the BCLR bit should be done while the FRDY flag in the FIFO port control register is 1 (set by the USB).

BVAL Bit (Buffer Memory Valid)

The BVAL bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe selected using the CURPIPE[3:0] bits (selected pipe).
When the selected pipe is in the transmitting direction, set the BVAL bit to 1 in the following cases. Then, the USB switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.

- To transmit a short packet, set the BVAL bit to 1 after data has been written.
- To transmit a zero-length packet, set the BVAL bit to 1 before data is written to the FIFO buffer.

When data of the maximum packet size has been written for the pipe in continuous transfer mode, the USB sets the BVAL bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.
Writing 1 to the BVAL bit should be done while the FRDY flag is 1 (set by the USB).
When the selected pipe is in the receiving direction, do not set the BVAL bit to 1.

38.2.7 Interrupt Enable Register 0 (INTENB0)

Address(es): 000A 0030h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BRDYE	Buffer Ready Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	NRDYE	Buffer Not Ready Response Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b10	BEMPE	Buffer Empty Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b11	CTRE	Control Transfer Stage Transition Interrupt Enable *1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b12	DVSE	Device State Transition Interrupt Enable*1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b13	SOFE	Frame Number Update Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b14	RSME	Resume Interrupt Enable*1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15	VBSE	VBUS Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W

Note 1. The RSME, DVSE, and CTRE bits can be set to 1 only when the function controller is selected; do not set these bits to 1 to enable the corresponding interrupt output when the host controller is selected.

On detecting the interrupt corresponding to the bit in the INTENB0 register to which 1 has been set by software, the USB generates the USB interrupt request.

The USB sets 1 to each status bit in the INTSTS0 register when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in the INTENB0 register (regardless of whether the interrupt output is enabled or disabled).

While the status bit in the INTSTS0 register corresponding to the interrupt source indicates 1, the USB generates the USB interrupt request when the corresponding interrupt enable bit in the INTENB0 register is modified from 0 to 1 by software.

38.2.8 Interrupt Enable Register 1 (INTENB1)

Address(es): 000A 0032h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	—	—	—	EOFERRE	SIGNE	SACKE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SACKE	Setup Transaction Normal Response Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	SIGNE	Setup Transaction Error Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	EOFERRE	EOF Error Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ATTCHE	Connection Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b12	DTCHE	Disconnection Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	BCHGE	USB Bus Change Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15	OVRCRE	Overcurrent Input Change Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W

Note: The bits in the INTENB1 register can be set to 1 only when the host controller is selected; do not set these bits to 1 to enable the corresponding interrupt output when the function controller is selected.

The INTENB1 register specifies the interrupt masks when the host controller is selected, and for the setup transaction. On detecting the interrupt corresponding to the bit in the INTENB1 register to which 1 has been set by software, the USB generates the USB interrupt request.

The USB sets 1 to each status bit in the INTSTS1 register when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in the INTENB1 register (regardless of whether the interrupt output is enabled or disabled).

While the status bit in the INTSTS1 register corresponding to the interrupt source indicates 1, the USB generates the USB interrupt request when the corresponding interrupt enable bit in the INTENB1 register is modified from 0 to 1 by software.

When the function controller is selected, the interrupts should not be enabled.

38.2.9 BRDY Interrupt Enable Register (BRDYENB)

Address(es): 000A 0036h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BRDYE	PIPE8BRDYE	PIPE7BRDYE	PIPE6BRDYE	PIPE5BRDYE	PIPE4BRDYE	PIPE3BRDYE	PIPE2BRDYE	PIPE1BRDYE	PIPE0BRDYE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BRDYE	BRDY Interrupt Enable for PIPE0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1BRDYE	BRDY Interrupt Enable for PIPE1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2BRDYE	BRDY Interrupt Enable for PIPE2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3BRDYE	BRDY Interrupt Enable for PIPE3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4BRDYE	BRDY Interrupt Enable for PIPE4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5BRDYE	BRDY Interrupt Enable for PIPE5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6BRDYE	BRDY Interrupt Enable for PIPE6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7BRDYE	BRDY Interrupt Enable for PIPE7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8BRDYE	BRDY Interrupt Enable for PIPE8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9BRDYE	BRDY Interrupt Enable for PIPE9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The BRDYENB register enables or disables the INTSTS0.BRDY flag to be set to 1 when the BRDY interrupt is detected for each pipe.

On detecting the BRDY interrupt for the pipe corresponding to the bit in the BRDYENB register to which 1 has been set by software, the USB sets 1 to the corresponding BRDYSTS.PIPEnBRDY flag (n = 0 to 9) and the INTSTS0.BRDY flag. If INTENB0.BRDYE = 1 at this time, the USB generates the BRDY interrupt request.

While at least one PIPEnBRDY flag indicates 1, the USB generates the BRDY interrupt request when the corresponding interrupt enable bit in the BRDYENB register is modified from 0 to 1 by software.

38.2.10 NRDY Interrupt Enable Register (NRDYENB)

Address(es): 000A 0038h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9NRDYE	PIPE8NRDYE	PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	PIPE3NRDYE	PIPE2NRDYE	PIPE1NRDYE	PIPE0NRDYE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0NRDYE	NRDY Interrupt Enable for PIPE0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1NRDYE	NRDY Interrupt Enable for PIPE1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2NRDYE	NRDY Interrupt Enable for PIPE2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3NRDYE	NRDY Interrupt Enable for PIPE3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4NRDYE	NRDY Interrupt Enable for PIPE4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5NRDYE	NRDY Interrupt Enable for PIPE5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6NRDYE	NRDY Interrupt Enable for PIPE6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7NRDYE	NRDY Interrupt Enable for PIPE7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8NRDYE	NRDY Interrupt Enable for PIPE8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9NRDYE	NRDY Interrupt Enable for PIPE9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The NRDYENB register enables or disables the INTSTS0.NRDY flag to be set to 1 when the NRDY interrupt is detected for each pipe.

On detecting the NRDY interrupt for the pipe corresponding to the bit in the NRDYENB register to which 1 has been set by software, the USB sets 1 to the corresponding NRDYSTS.PIPEnNRDY flag (n = 0 to 9) and the INTSTS0.NRDY flag. If INTENB0.NRDYE = 1 at this time, the USB generates the NRDY interrupt request.

While at least one PIPEnNRDY flag indicates 1, the USB generates the NRDY interrupt request when the corresponding interrupt enable bit in the NRDYENB register is modified from 0 to 1 by software.

38.2.11 BEMP Interrupt Enable Register (BEMPENB)

Address(es): 000A 003Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BEMPE	PIPE8BEMPE	PIPE7BEMPE	PIPE6BEMPE	PIPE5BEMPE	PIPE4BEMPE	PIPE3BEMPE	PIPE2BEMPE	PIPE1BEMPE	PIPE0BEMPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BEMPE	BEMP Interrupt Enable for PIPE0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1BEMPE	BEMP Interrupt Enable for PIPE1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2BEMPE	BEMP Interrupt Enable for PIPE2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3BEMPE	BEMP Interrupt Enable for PIPE3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4BEMPE	BEMP Interrupt Enable for PIPE4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5BEMPE	BEMP Interrupt Enable for PIPE5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6BEMPE	BEMP Interrupt Enable for PIPE6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7BEMPE	BEMP Interrupt Enable for PIPE7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8BEMPE	BEMP Interrupt Enable for PIPE8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9BEMPE	BEMP Interrupt Enable for PIPE9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The BEMPENB register enables or disables the INTSTS0.BEMP flag to be set to 1 when the BEMP interrupt is detected for each pipe.

On detecting the BEMP interrupt for the pipe corresponding to the bit in the BEMPENB register to which 1 has been set by software, the USB sets 1 to the corresponding BEMPSTS.PIPEnBEMP flag (n = 0 to 9) and the INTSTS0.BEMP flag. If INTENB0.BEMPE = 1 at this time, the USB generates the BEMP interrupt request.

While at least one PIPEnBEMP flag in the BEMPSTS register indicates 1, the USB generates the BEMP interrupt request when the corresponding interrupt enable bit in the BEMPENB register is modified from 0 to 1 by software.

38.2.12 SOF Output Configuration Register (SOFCFG)

Address(es): 000A 003Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TRNENSEL	—	BRDYM	—	EDGESTS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	EDGESTS	Edge Interrupt Output Status Monitor Flag*1	Indicates 1 when the edge interrupt output signal is in the middle of the edge processing.	R
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	BRDYM	BRDY Interrupt Status Clear Timing	0: Software clears the status. 1: The USB clears the status when data has been read from the FIFO buffer or data has been written to the FIFO buffer.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	TRNENSEL	Transaction-Enabled Time Select*1	0: For non-low-speed communication 1: For low-speed communication	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Confirm that this bit is 0 before stopping the clock supply to the USB module.

EDGESTS Flag (Edge Interrupt Output Status Monitor Flag)

The EDGESTS flag indicates 1 when the edge interrupt output signal is in the middle of the edge processing.
Confirm that this flag is 0 before stopping the clock supply to the USB.

BRDYM Bit (BRDY Interrupt Status Clear Timing)

The BRDYM bit specifies the timing for clearing the BRDY interrupt status for each pipe.

TRNENSEL Bit (Transaction-Enabled Time Select)

The TRNENSEL bit selects, for full-speed or low-speed communication, the transaction-enabled time in which the USB issues tokens in a frame via the port.
Set the TRNENSEL bit to 1 when a low-speed device is connected.
The TRNENSEL bit is valid only when the host controller is selected.
This bit should be set to 0 if the function controller is selected.

38.2.13 Interrupt Status Register 0 (INTSTS0)

Address(es): 000A 0040h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
Value after reset:	0	0	0	0/1*1	0	0	0	0	0*2	0*3	0*3	0/1*3	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CTSQ[2:0]	Control Transfer Stage Flag	b2 b0 0 0 0: Idle or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error	R
b3	VALID	USB Request Reception Flag	0: Setup packet is not received. 1: Setup packet is received.	R/W
b6 to b4	DVSQ[2:0]	Device State Flag	b6 b4 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state 1 x x: Suspended state	R
b7	VBSTS	VBUS Input Status Flag	0: USB0_VBUS pin is low. 1: USB0_VBUS pin is high.	R
b8	BRDY	Buffer Ready Interrupt Status Flag	0: BRDY interrupts are not generated. 1: BRDY interrupts are generated.	R
b9	NRDY	Buffer Not Ready Interrupt Status Flag	0: NRDY interrupts are not generated. 1: NRDY interrupts are generated.	R
b10	BEMP	Buffer Empty Interrupt Status Flag	0: BEMP interrupts are not generated. 1: BEMP interrupts are generated.	R
b11	CTRT	Control Transfer Stage Transition Interrupt Status Flag*5	0: Control transfer stage transition interrupts are not generated. 1: Control transfer stage transition interrupts are generated.	R/W*4
b12	DVST	Device State Transition Interrupt Status Flag*5	0: Device state transition interrupts are not generated. 1: Device state transition interrupts are generated.	R/W*4
b13	SOFR	Frame Number Refresh Interrupt Status Flag	0: SOF interrupts are not generated. 1: SOF interrupts are generated.	R/W*4
b14	RESM	Resume Interrupt Status Flag *5,*6	0: Resume interrupts are not generated. 1: Resume interrupts are generated.	R/W*4
b15	VBINT	VBUS Interrupt Status Flag*6	0: VBUS interrupts are not generated. 1: VBUS interrupts are generated.	R/W*4

x: Don't care

Note 1. The value is 0 when the MCU is reset and 1 after a USB bus reset.
Note 2. The value is 1 when the USB0_VBUS pin is high and 0 when the USB0_VBUS pin is low.
Note 3. The value is 000b when the MCU is reset and 001b after a USB bus reset.
Note 4. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID flag, write 0 only to the flags to be cleared; write 1 to the other flags. Do not write 0 to the status flags indicating 0.
Note 5. The status of the RESM, DVST, and CTRT flags are changed only when the function controller is selected. Set the corresponding interrupt enable bits to 0 (disabled) when the host controller is selected.
Note 6. A change in the status indicated by the VBINT and RESM flags can be detected even while the clock supply is stopped (the SCKE bit = 0), and the interrupts are output when the corresponding interrupt enable bits are enabled. Clearing the status through software should be done after enabling the clock supply.

CTSQ[2:0] Flags (Control Transfer Stage Flag)

When the host controller is selected, the read value is invalid.

VALID Flag (USB Request Reception Flag)

When the host controller is selected, the read value is invalid.

DVSQ[2:0] Flags (Device State Flag)

The DVSQ[2:0] flags are initialized by a USB bus reset.
When the host controller is selected, the read value is invalid.

BRDY Flag (Buffer Ready Interrupt Status Flag)

Indicates the BRDY interrupt status.
The USB sets the BRDY flag to 1 when at least one PIPEnBRDY flag (n = 0 to 9) is set to 1 among the PIPEnBRDY flags. These bits correspond to the BRDYENB.PIPEnBRDYE bits (n = 0 to 9) to which 1 has been set, when the USB detects the BRDY interrupt status in at least one pipe among the pipes for which the BRDY interrupt output is enabled by software.
For the conditions for PIPEnBRDY status assertion, refer to section 38.3.3.1, BRDY Interrupt.
The USB sets the BRDY flag to 0 when 0 is written by software to all the PIPEnBRDY flags corresponding to the PIPEnBRDYE bits that have been set to 1.
The BRDY flag cannot be set to 0 even if 0 is written to this bit by software.

NRDY Flag (Buffer Not Ready Interrupt Status Flag)

The USB sets the NRDY flag to 1 when at least one PIPEnNRDY flag (n = 0 to 9) is set to 1 among the PIPEnNRDY flags corresponding to the PIPEnNRDYE bits (n = 0 to 9) to which 1 has been set (when the USB detects the NRDY interrupt status in at least one pipe among the pipes for which software enables the NRDY interrupt output).
For the conditions for PIPEnNRDY status assertion, refer to section 38.3.3.2, NRDY Interrupt.
The USB sets the NRDY flag to 0 when 0 is written by software to all the PIPEnNRDY flags corresponding to the PIPEnNRDYE bits that have been set to 1.
The NRDY flag cannot be set to 0 even if 0 is written to this bit by software.

BEMP Flag (Buffer Empty Interrupt Status Flag)

The USB sets the BEMP flag to 1 when at least one PIPEnBEMP flag (n = 0 to 9) is set to 1 among the PIPEnBEMP flags corresponding to the PIPEnBEMPE bits (n = 0 to 9) to which 1 has been set (when the USB detects the BEMP interrupt status in at least one pipe among the pipes for which the BEMP interrupt output is enabled by software).
For the conditions for PIPEnBEMP status assertion, refer to section 38.3.3.3, BEMP Interrupt.
The USB sets the BEMP flag to 0 when 0 is written by software to all the PIPEnBEMP flags corresponding to the PIPEnBEMPE bits that have been set to 1.
The BEMP flag cannot be set to 0 even if 0 is written to this bit by software.

CTRT Flag (Control Transfer Stage Transition Interrupt Status Flag)

When the function controller is selected, the USB updates the value of the CTSQ[2:0] flags and sets the CTRT flag to 1 on detecting a change in the control transfer stage.
When a control transfer stage transition interrupt is generated, clear the status before the USB detects the next control transfer stage transition.
When the host controller is selected, the read value is invalid.

DVST Flag (Device State Transition Interrupt Status Flag)

When the function controller is selected, the USB updates the DVSQ[2:0] value and sets the DVST flag to 1 on detecting a change in the device state.
When a device state transition interrupt is generated, clear the status before the USB detects the next device state transition.
When the host controller is selected, the read value is invalid.

SOFR Flag (Frame Number Refresh Interrupt Status Flag)

(1) When the host controller is selected
The USB sets the SOFR flag to 1 on updating the frame number when the DVSTCTR0.UACT bit has been set to 1 by software. (A frame number refresh interrupt is detected every 1 ms.)
(2) When the function controller is selected
The USB sets the SOFR flag to 1 on updating the frame number. (A frame number refresh interrupt is detected every 1 ms.)
The USB can detect an SOFR interrupt through the internal interpolation function even when a damaged SOF packet is received from the USB host.

RESM Flag (Resume Interrupt Status Flag)

When the function controller is selected, the USB sets the RESM flag to 1 on detecting the falling edge of the signal on the USB0_DP pin in the suspended state (DVSQ[2:0] = 1xxb).
When the host controller is selected, the read value is invalid.

VBINT Flag (VBUS Interrupt Status Flag)

The USB sets the VBINT flag to 1 on detecting a level change (high to low or low to high) in the USB0_VBUS pin input value. The USB sets the VBSTS flag to indicate the USB0_VBUS pin input value. When the VBUS interrupt is generated, use software to repeat reading the VBSTS flag until the same value is read three or more times, and eliminate chattering.

38.2.14 Interrupt Status Register 1 (INTSTS1)

Address(es): 000A 0042h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCR	BCHG	—	DTCH	ATTCH	—	—	—	—	EOFERR	SIGN	SACK	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SACK	Setup Transaction Normal Response Interrupt Status Flag	0: SACK interrupts are not generated. 1: SACK interrupts are generated.	R/W *1
b5	SIGN	Setup Transaction Error Interrupt Status Flag	0: SIGN interrupts are not generated. 1: SIGN interrupts are generated.	R/W *1
b6	EOFERR	EOF Error Detection Interrupt Status Flag	0: EOFERR interrupts are not generated. 1: EOFERR interrupts are generated.	R/W *1
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ATTCH	ATTCH Interrupt Status Flag	0: ATTCH interrupts are not generated. 1: ATTCH interrupts are generated.	R/W *1
b12	DTCH	USB Disconnection Detection Interrupt Status Flag	0: DTCH interrupts are not generated. 1: DTCH interrupts are generated.	R/W *1
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	BCHG	USB Bus Change Interrupt Status Flag *2	0: BCHG interrupts are not generated. 1: BCHG interrupts are generated.	R/W *1
b15	OVRCR	Overcurrent Input Change Interrupt Status Flag*2	0: OVRCR interrupts are not generated. 1: OVRCR interrupts are generated.	R/W *1

Note 1. To clear the status indicated by the flags in the INTSTS1 register, write 0 only to the flags to be cleared; write 1 to the other flags.

Note 2. A change in the status indicated by the OVRCR or BCHG flag can be detected even while the clock supply is stopped (while the SYSCFG.SCKE bit = 0), and the interrupt is output when the corresponding interrupt enable bit is enabled. Clearing the status through software should be done after setting the SYSCFG.SCKE bit to 1.
No interrupts other than those indicated by the BCHG and OVRCR flags can be detected while the clock supply is stopped (while the SYSCFG.SCKE bit = 0).

The INTSTS1 register is used to confirm the status of each interrupt when the host controller is selected.
The various status change interrupts indicated by the flags in the INTSTS1 register should be enabled only when the host controller is selected.

SACK Flag (Setup Transaction Normal Response Interrupt Status Flag)

Indicates the status of the setup transaction normal response interrupt when the host controller is selected.
The USB detects the SACK interrupt when ACK response is returned from the peripheral device during the setup transactions issued by the USB, and sets the SACK flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the SACK interrupt.
When the function controller is selected, the read value is invalid.

SIGN Flag (Setup Transaction Error Interrupt Status Flag)

Indicates the status of the setup transaction error interrupt when the host controller is selected.
The USB detects the SIGN interrupt when ACK response is not returned from the peripheral device three consecutive times during the setup transactions issued by this module, and sets the SIGN flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the SIGN interrupt.
Specifically, the USB detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions.

- Timeout is detected by the USB when the peripheral device has returned no response.
- A damaged ACK packet is received.
- A handshake other than ACK (NAK, NYET, or STALL) is received.

When the function controller is selected, the read value is invalid.

EOFERR Flag (EOF Error Detection Interrupt Status Flag)

Indicates the status of the EOFERR interrupt when the host controller is selected.
The USB detects the EOFERR interrupt on detecting that communication is not completed at the EOF2 timing prescribed by USB Specification 2.0, and sets the EOFERR flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the EOFERR interrupt.
After detecting the EOFERR interrupt, the USB controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the USB port should be terminated by software and perform re-enumeration of the USB port.

- Modifies the DVSTCTR0.UACT bit for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

When the function controller is selected, the read value is invalid.

ATTCH Flag (ATTCH Interrupt Status Flag)

Indicates the status of the ATTCH interrupt when the host controller is selected.
The USB detects the ATTCH interrupt on detecting J-state or K-state of the full-speed or low-speed signal level for 2.5 µs, and sets the ATTCH flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.
Specifically, the USB detects the ATTCH interrupt on any of the following conditions.

- K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5 µs.
- J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 µs.

When the function controller is selected, the read value is invalid.

DTCH Flag (USB Disconnection Detection Interrupt Status Flag)

Indicates the status of the USB disconnection detection interrupt when the host controller is selected.
The USB detects the DTCH interrupt on detecting USB bus disconnection, and sets the DTCH flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.
The USB detects bus disconnection based on USB Specification 2.0.
After detecting the DTCH interrupt, the USB controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). All the pipes in which communications are currently carried out for the USB port should be terminated by software and make a transition to the wait state for bus connection to the USB port (wait state for ATTCH interrupt generation).

- Modifies the DVSTCTR0.UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the idle state.

When the function controller is selected, the read value is invalid.

BCHG Flag (USB Bus Change Interrupt Status Flag)

Indicates the status of the USB bus change interrupt.
The USB detects the BCHG interrupt when a change in the full-speed or low-speed signal level occurs on the USB port (a change from J-state, K-state, or SE0 to J-state, K-state, or SE0), and sets the BCHG flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.
The USB sets the LNST[1:0] flags to indicate the current input state of the USB port. When the BCHG interrupt is generated, use software to repeat reading the LNST[1:0] flags until the same value is read three or more times, and eliminate chattering.
A change in the USB bus state can be detected even while the internal clock supply is stopped.
When the function controller is selected, the read value is invalid.

OVRCR Flag (Overcurrent Input Change Interrupt Status Flag)

Indicates the status of the USB0_OVRCURA and USB0_OVRCURB input pin change interrupt.
The USB detects the OVRCR interrupt when a change (high to low or low to high) occurs in at least one of the input values to the USB0_OVRCURA and USB0_OVRCURB pins, and sets the OVRCR flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.

38.2.15 BRDY Interrupt Status Register (BRDYSTS)

Address(es): 000A 0046h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BRDY	PIPE8BRDY	PIPE7BRDY	PIPE6BRDY	PIPE5BRDY	PIPE4BRDY	PIPE3BRDY	PIPE2BRDY	PIPE1BRDY	PIPE0BRDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BRDY	BRDY Interrupt Status Flag for PIPE0*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1BRDY	BRDY Interrupt Status Flag for PIPE1*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2BRDY	BRDY Interrupt Status Flag for PIPE2*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3BRDY	BRDY Interrupt Status Flag for PIPE3*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4BRDY	BRDY Interrupt Status Flag for PIPE4*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5BRDY	BRDY Interrupt Status Flag for PIPE5*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6BRDY	BRDY Interrupt Status Flag for PIPE6*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7BRDY	BRDY Interrupt Status Flag for PIPE7*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8BRDY	BRDY Interrupt Status Flag for PIPE8*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9BRDY	BRDY Interrupt Status Flag for PIPE9*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the SOFCFG.BRDYM bit is set to 0, to clear the status indicated by the flags in the BRDYSTS register, write 0 only to the flags to be cleared; write 1 to the other flags.

Note 2. When the SOFCFG.BRDYM bit is set to 0, clearing BRDY Interrupts should be done before accessing the FIFO.

38.2.16 NRDY Interrupt Status Register (NRDYSTS)

Address(es): 000A 0048h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9NRDY	PIPE8NRDY	PIPE7NRDY	PIPE6NRDY	PIPE5NRDY	PIPE4NRDY	PIPE3NRDY	PIPE2NRDY	PIPE1NRDY	PIPE0NRDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0NRDY	NRDY Interrupt Status Flag for PIPE0	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1NRDY	NRDY Interrupt Status Flag for PIPE1	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2NRDY	NRDY Interrupt Status Flag for PIPE2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3NRDY	NRDY Interrupt Status Flag for PIPE3	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4NRDY	NRDY Interrupt Status Flag for PIPE4	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5NRDY	NRDY Interrupt Status Flag for PIPE5	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6NRDY	NRDY Interrupt Status Flag for PIPE6	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7NRDY	NRDY Interrupt Status Flag for PIPE7	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8NRDY	NRDY Interrupt Status Flag for PIPE8	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9NRDY	NRDY Interrupt Status Flag for PIPE9	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To clear the status indicated by the flags in the NRDYSTS register, write 0 only to the flags to be cleared; write 1 to the other flags.

38.2.17 BEMP Interrupt Status Register (BEMPSTS)

Address(es): 000A 004Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BEMP	PIPE8BEMP	PIPE7BEMP	PIPE6BEMP	PIPE5BEMP	PIPE4BEMP	PIPE3BEMP	PIPE2BEMP	PIPE1BEMP	PIPE0BEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BEMP	BEMP Interrupt Status Flag for PIPE0	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1BEMP	BEMP Interrupt Status Flag for PIPE1	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2BEMP	BEMP Interrupt Status Flag for PIPE2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3BEMP	BEMP Interrupt Status Flag for PIPE3	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4BEMP	BEMP Interrupt Status Flag for PIPE4	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5BEMP	BEMP Interrupt Status Flag for PIPE5	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6BEMP	BEMP Interrupt Status Flag for PIPE6	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7BEMP	BEMP Interrupt Status Flag for PIPE7	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8BEMP	BEMP Interrupt Status Flag for PIPE8	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9BEMP	BEMP Interrupt Status Flag for PIPE9	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To clear the status indicated by the flags in the BEMPSTS register, write 0 only to the flags to be cleared; write 1 to the other flags.

38.2.18 Frame Number Register (FRMNUM)

Address(es): 000A 004Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRN	CRCE	—	—	—	FRNM[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b10 to b0	FRNM[10:0]	Frame Number Flag	Latest frame number	R
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	CRCE	Receive Data Error Flag	0: No error 1: An error occurred	R/W*1
b15	OVRN	Overrun/Underrun Detection Status Flag	0: No error 1: An error occurred	R/W*1

Note 1. To clear the status, write 0 only to the bits to be cleared; write 1 to the other bits.

FRNM[10:0] Flags (Frame Number Flag)

These bits indicate the latest frame number for the USB after the issuing of an SOF packet every 1 ms or writing to the FRNM[10:0] flags at the SOF packet reception.

CRCE Flag (Receive Data Error Flag)

Indicates whether a CRC error or bit stuffing error has been detected in the pipe during isochronous transfer.
The CRCE flag can be set to 0 by writing 0 to the CRCE flag by software. Here, 1 should be written to the other bits in FRMNUM.
On detecting a CRC error, the USB generates the internal NRDY interrupt request.

OVRN Flag (Overrun/Underrun Detection Status Flag)

Indicates whether an overrun/underrun error has been detected in the pipe during isochronous transfer.
The OVRN flag can be set to 0 by writing 0 to the OVRN flag by software. Here, 1 should be written to the other bits in FRMNUM.

(1) When the host controller is selected

The USB sets the OVRN flag to 1 on any of the following conditions.

- For the isochronous transfer pipe in the transmitting direction, the time to issue an OUT token comes before all the transmit data has been written to the FIFO buffer.
- For the isochronous transfer pipe in the receiving direction, the time to issue an IN token comes when no FIFO buffer planes are empty.

(2) When the function controller is selected

The USB sets the OVRN flag to 1 on any of the following conditions.

- For the isochronous transfer pipe in the transmitting direction, the IN token is received before all the transmit data has been written to the FIFO buffer.
- For the isochronous transfer pipe in the receiving direction, the OUT token is received when no FIFO buffer planes are empty.

38.2.19 Device State Change Register (DVCHGR)

Address(es): 000A 004Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DVCHG	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b14 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	DVCHG	Device State Change	0: Disables the writing to the USBADDR.STSRECOV[3:0] bits and USBADDR.USBADDR[6:0] bits. 1: Enables the writing to the USBADDR.STSRECOV[3:0] bits and USBADDR.USBADDR[6:0] bits.	R/W

For details, refer to section 38.3.1.5, Release from Deep Software Standby Mode Due to USB Suspend/ Resume Interrupts.

38.2.20 USB Address Register (USBADDR)

Address(es): 000A 0050h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	STSRECOV[3:0]				—	USBADDR[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	USBADDR[6:0]	USB Address	When the function controller is selected, these bits indicate the USB address assigned by the host when the SET_ADDRESS request is successfully processed.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b11 to b8	STSRECOV[3:0]	Status Recovery	• Recovery when the function controller is selected b11 b8 1 0 0 1: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 001b (Default state) 1 0 1 0: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 010b (Address state) 1 0 1 1: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 011b (Configured state) Settings other than above are prohibited. • Recovery when the host controller is selected b11 b8 1 0 0 0: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b) Settings other than above are prohibited.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

USBADDR[6:0] Bits (USB Address)

On detecting the USB bus reset, the USB sets the USBADDR[6:0] bits to 00h.

The writing to these bits is enabled while the DVCHGR.DVCHG bits are set to 1. On returning from the USB power shut-off, the operation can resume to the USB address before the shut-off by the software.

When the host controller is selected, the USBADDR[6:0] bits are invalid.

The USBADDR[6:0] bits are initialized when a USB bus reset is detected.

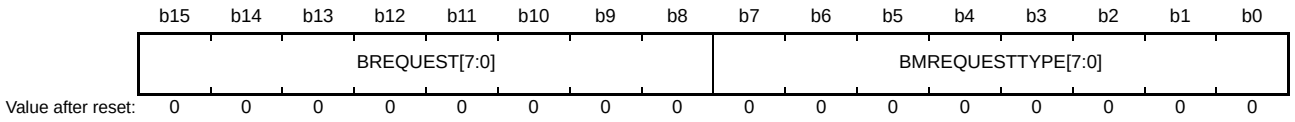
STSRECOV[3:0] Bits (Status Recovery)

These bits are used to resume the state of the internal sequencer on returning from the USB power shut-off. For details, refer to section 38.3.1.5, Release from Deep Software Standby Mode Due to USB Suspend/Resume Interrupts.

The writing to the STSRECOV[3:0] bits is enabled while the DVCHGR.DVCHG bit is set to 1.

38.2.21 USB Request Type Register (USBREQ)

Address(es): 000A 0054h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	BMREQUESTTYPE[7:0]	Request Type	These bits store the USB request bmRequestType value.	R/W *1
b15 to b8	BREQUEST[7:0]	Request	These bits store the USB request bRequest value.	R/W *1

Note 1. When the function controller is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller is selected, these bits can be read from and written to.

The USBREQ register stores setup requests for control transfers.
When the function controller is selected, the values of bRequest and bmRequestType that have been received are stored.
When the host controller is selected, the values of bRequest and bmRequestType to be transmitted are set.
The USBREQ register is initialized by a USB bus reset.

BMREQUESTTYPE[7:0] Bits (Request Type)

These bits hold the value of the bmRequestType field of a USB request.

- When the host controller is selected:
  Set these bits to the value of the USB request data in setup transactions for transmission. Do not overwrite the value of the BMREQUESTTYPE[7:0] bits while the DCPCTR.SUREQ bit = 1.
- When the function controller is selected:
  These bits indicate the value of the USB request data in setup transactions for reception. Writing to the bits has no effect.

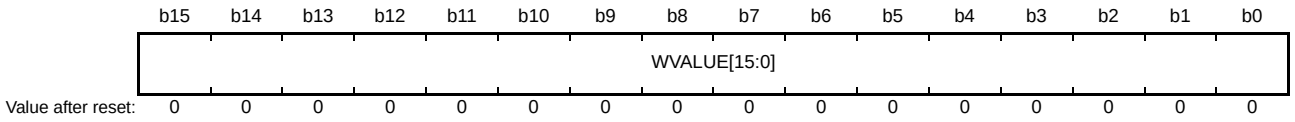
BREQUEST[7:0] Bits (Request)

These bits store bRequest value of the USB request.

- When the host controller is selected:
  Set these bits to the value of the USB request data in setup transactions for transmission. Do not overwrite the value of the BREQUEST[7:0] bits while the DCPCTR.SUREQ bit = 1.
- When the function controller is selected:
  These bits indicate the value of the USB request data in setup transactions for reception. Writing to the bits has no effect.

38.2.22 USB Request Value Register (USBVAL)

Address(es): 000A 0056h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	WVALUE[15:0]	Value	These bits store the USB request wValue value.	R/W *1

Note 1. When the function controller is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller is selected, these bits can be read from and written to.

When the function controller is selected, the value of wValue that has been received is stored in the USBVAL register.
When the host controller is selected, the value of wValue to be transmitted is set.
The USBVAL register is initialized by a USB bus reset.

WVALUE[15:0] Bits (Value)

These bits store wRequest value of the USB request.

- When the host controller is selected:
  Set these bits to the value of the wValue field in USB requests of setup transactions for transmission. Do not overwrite the value of the WVALUE[15:0] bits while the DCPCTR.SUREQ bit = 1.
- When the function controller is selected:
  These bits indicate the value of the wValue field in USB requests received in setup transactions for reception. Writing to the WVALUE[15:0] bits has no effect.

38.2.23 USB Request Index Register (USBINDX)

Address(es): 000A 0058h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WINDEX[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	WINDEX[15:0]	Index	These bits store the USB request wIndex value.	R/W *1

Note 1. When the function controller is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller is selected, these bits can be read from and written to.

The USBINDX register stores setup requests for control transfers.
When the function controller is selected, the value of wIndex that has been received is stored. When the host controller is selected, the value of wIndex to be transmitted is set.
The USBINDX register is initialized by a USB bus reset.

WINDEX[15:0] Bits (Index)

These bits hold the value of the wIndex field of a USB request.

- When the host controller is selected:
  Set these bits to the value of the wIndex field in USB requests of setup transactions for transmission. Do not overwrite the value of the WINDEX[15:0] bits while the DCPCTR.SUREQ bit = 1.
- When the function controller is selected:
  These bits indicate the value of the wIndex field in USB requests received in setup transactions for reception. Writing to the WINDEX[15:0] bits has no effect.

38.2.24 USB Request Length Register (USBLENG)

Address(es): 000A 005Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WLENGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	WLENGTH[15:0]	Length	These bits store the USB request wLength value.	R/W *1

Note 1. When the function controller is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller is selected, these bits can be read from and written to.

The USBLENG register stores setup requests for control transfers.
When the function controller is selected, the value of wLength that has been received is stored. When the host controller is selected, the value of wLength to be transmitted is set.
The USBLENG register is initialized by a USB bus reset.

WLENGTH[15:0] Bits (Length)

These bits hold the value of the wLength field of a USB request.

- When the host controller is selected:
  Set these bits to the value of the wLength field in USB requests of setup transactions for transmission. Do not overwrite the value of the WLENGTH[15:0] bits while the DCPCTR.SUREQ bit = 1.
- When the function controller is selected:
  These bits indicate the value of the wLength field in USB requests received in setup transactions for reception. Writing to the WLENGTH[15:0] bits has no effect.

38.2.25 DCP Configuration Register (DCPCFG)

Address(es): 000A 005Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SHTNAK	—	—	DIR	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DIR	Transfer Direction*1	0: Data receiving direction 1: Data transmitting direction	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: Pipe continued at the end of transfer 1: Pipe disabled at the end of transfer	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Modify this bit while DCPCTR.PID[1:0] bits are 00b (NAK). Before modifying this bit after modifying the DCPCTR.PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the DCPCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

DIR Bit (Transfer Direction)

When the host controller is selected, the DIR bit sets the transfer direction of the data stage and status stage.
When the function controller is selected, the DIR bit should be set to 0.

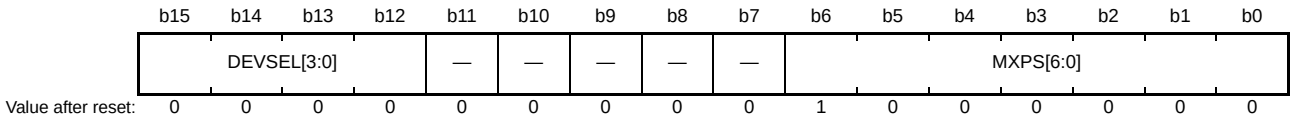
SHTNAK Bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to modify PID[1:0] bits to 00b (NAK) upon the end of transfer when the selected pipe is in the receiving direction.
The SHTNAK bit is valid when the selected pipe in the receiving direction.
When the SHTNAK bit is set to 1, the USB modifies the DCPCTR.PID[1:0] bits for the DCP to 00b (NAK) on determining the end of the transfer. The USB determines that the transfer has ended on the following condition.

- A short packet (including a zero-length packet) is successfully received.

38.2.26 DCP Maximum Packet Size Register (DCPMAXP)

Address(es): 000A 005Eh



Bit	Symbol	Bit Name	Description	R/W
b6 to b0	MXPS[6:0]	Maximum Packet Size*1	These bits set the maximum amount of data (maximum packet size) in payloads for the DCP.	R/W
b11 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*2	b15 b12 0 0 0 0: Address 0000 0 0 0 1: Address 0001 0 0 1 0: Address 0010 0 0 1 1: Address 0011 0 1 0 0: Address 0100 0 1 0 1: Address 0101 Settings other than above are prohibited.	R/W

Note 1. Modify the MXPS[6:0] bits while PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the DCPCTR.PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the DCPCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software. After modifying the MXPS[6:0] bits and the DCP has been set to the CURPIPE[3:0] bits in a port select register, clear the buffer by setting the BCLR bit the port control register to 1.

Note 2. Modify the DEVSEL[3:0] bits while PID[1:0] bits are 00b (NAK) and the DCPCTR.SUREQ bit is 0. To modify these bits after modifying the DCPCTR.PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the DCPCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

MXPS[6:0] Bits (Maximum Packet Size)

The MXPS[6:0] bits specify the maximum amount of data (maximum packet size) in payloads for the DCP. The initial value of the bits is 40h (64 bytes).

Ensure that the setting of the MXPS[6:0] bits is in compliance with USB Specification 2.0.

Do not write to the FIFO buffer or set PID[1:0] = 01b (BUF) while the setting of the MXPS[6:0] bits is 0.

DEVSEL[3:0] Bits (Device Select)

When the host controller is selected, these bits specify the address of the peripheral device which is the communication target during control transfer.

The DEVSEL[3:0] bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in the DEVSEL[3:0] bits. For example, before setting the DEVSEL[3:0] bits to 0010b, the address should be set to DEVADD2.

When the function controller is selected, the DEVSEL[3:0] bits should be set to 0000b.

38.2.27 DCP Control Register (DCPCTR)

Address(es): 000A 0060h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	SUREQ	—	—	SUREQ CLR	—	—	SQCLR	SQSET	SQMON	PBUSY	—	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b2	CCPL	Control Transfer End Enable	0: Invalid 1: Completion of control transfer is enabled.	R/W
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: DCP is not used for the transaction. 1: DCP is used for the transaction.	R
b6	SQMON	Sequence Toggle Bit Monitor Flag	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W*1
b8	SQCLR	Sequence Toggle Bit Clear*2	0: Invalid 1: Specifies DATA0.	R/W*1
b10, b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	SUREQCLR	SUREQ Bit Clear	0: Invalid 1: Clears the SUREQ bit to 0.	R/W
b13, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	SUREQ	Setup Token Transmission	0: Invalid 1: Transmits the setup packet.	R/W
b15	BSTS	Buffer Status Flag	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only 0 can be read.
Note 2. Write 1 to the SQSET and SQCLR bits while PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

PID[1:0] Bits (Response PID)

The PID[1:0] bits control the response type of the USB during control transfer.

(1) When the host controller is selected

Modify the setting of the PID[1:0] bits from 00b (NAK) to 01b (BUF) using the following procedure.

- When the transmitting direction is set
  Write all the transmit data to the FIFO buffer while the DVSTCTR0.UACT bit is 1 and PID[1:0] bits are 00b (NAK), and then write 01b (BUF response). After PID[1:0] have been set to 01b (BUF), the USB executes the OUT transaction.
- When the receiving direction is set
  Check that the FIFO buffer is empty (or empty the buffer) while the DVSTCTR0.UACT bit is 1 and PID[1:0] bits are 00b (NAK), and then set PID[1:0] bits to 01b (BUF). After PID[1:0] bits have been set to 01b (BUF), the USB executes the IN transaction.

The USB modifies the setting of the PID[1:0] bits as follows.

- The USB sets PID[1:0] bits to 11b (STALL) on receiving the data of a size exceeding the maximum packet size when the PID[1:0] bits has been set to 01b (BUF) by software.
- The USB sets PID[1:0] bits to 00b (NAK) on detecting a receive error, such as a CRC error, three consecutive times.
- The USB also sets PID[1:0] bits to 11b (STALL) on receiving the STALL handshake.

(2) When the function controller is selected

The USB modifies the setting of the PID[1:0] bits as follows.

- The USB modifies the PID[1:0] bits to 00b (NAK) on receiving the setup packet. Here, the USB sets the INTSTS0.VALID flag to 1. The setting of the PID[1:0] bits cannot be modified until the VALID flag is set to 0 by software.
- The USB sets PID[1:0] bits to 11b (STALL) on receiving the data of a size exceeding the maximum packet size when the PID[1:0] bits have been set to 01b (BUF) by software.
- The USB sets PID[1:0] bits to 1xb (STALL) on detecting the control transfer sequence error.
- The USB sets PID[1:0] bits to 00b (NAK) on detecting the USB bus reset.

The USB does not check the setting of the PID[1:0] bits while the SET_ADDRESS request is processed.
The PID[1:0] bits are initialized by a USB bus reset.

CCPL Bit (Control Transfer End Enable)

When the function controller is selected, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed.
When the CCPL bit is set to 1 by software while the corresponding PID[1:0] bits are set to 01b (BUF), the USB completes the control transfer status stage.
During control read transfer, the USB transmits the ACK handshake in response to the OUT transaction from the USB host, and transmits the zero-length packet in response to the IN transaction from the USB host during control write or no-data control transfer. However, on detecting the SET_ADDRESS request, the USB operates in auto response mode from the setup stage up to the status stage completion irrespective of the setting of the CCPL bit.
The USB modifies the CCPL bit from 1 to 0 on receiving a new setup packet.
1 cannot be written to the CCPL bit by software while the INTSTS0.VALID flag is 1.
The CCPL bit is initialized by a USB bus reset.
When the host controller is selected, be sure to write 0 to the CCPL bit.

PBUSY Flag (Pipe Busy Flag)

The PBUSY flag indicates whether DCP is used or not for the transaction when USB changes the PID[1:0] bits from 01b (BUF) to 00b (NAK).
The USB modifies the PBUSY flag from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY flag from 1 to 0 upon completion of one transaction.
Reading the PBUSY flag after the PID[1:0] bits have been set to 00b (NAK) by software allows checking whether modification of the pipe settings is possible.
For details, refer to section 38.3.4.1, Pipe Control Register Switching Procedures.

SQMON Flag (Sequence Toggle Bit Monitor Flag)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction during the DCP transfer.
The USB allows the SQMON flag to toggle upon normal completion of the transaction. However, the SQMON flag is not allowed to toggle when a data PID mismatch occurs during the transfer in the receiving direction.
When the function controller is selected, the USB sets the SQMON flag to 1 (specifies DATA1 as the expected value) upon successful reception of the setup packet.
When the function controller is selected, the USB does not reference the SQMON flag during the IN/OUT transaction of the status stage, and does not allow the SQMON flag to toggle upon normal completion.

SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer.
Do not set the SQCLR and SQSET bits to 1 simultaneously.

SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. The SQCLR bit indicates 0.
Do not set the SQCLR and SQSET bits to 1 simultaneously.

SUREQCLR Bit (SUREQ Bit Clear)

When the host controller is selected, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The SUREQCLR bit indicates 0.
Set the SUREQCLR bit to 1 through software when communication has stopped with the SUREQ bit being 1 during the setup transaction. However, for normal setup transactions, the USB automatically clears the SUREQ bit to 0 upon completion of the transaction; therefore, clearing the SUREQ bit through software is not necessary.
Controlling the SUREQ bit through the SUREQCLR bit must be done while the DVSTCTR0.UACT bit is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected.
When the function controller is selected, be sure to write 0 to the SUREQCLR bit.

SUREQ Bit (Setup Token Transmission)

The USB transmits the setup packet by setting the SUREQ bit to 1 when the host controller is selected.
After completing the setup transaction process, the USB generates either the SACK or SIGN interrupt and sets the SUREQ bit to 0.
The USB also sets the SUREQ bit to 0 when software sets the SUREQCLR bit to 1.
Before setting the SUREQ bit to 1, set the DCPMAXP.DEVSEL[3:0] bits, registers USBREQ, USBVAL, USBINDX, and USBLENG appropriately to transmit the desired USB request in the setup transaction. Before setting this bit to 1, check that the PID[1:0] bits for the DCP are set to 00b (NAK). After setting the SUREQ bit to 1, do not modify the DCPMAXP.DEVSEL[3:0] bits, registers USBREQ, USBVAL, USBINDX, or USBLENG until the setup transaction is completed (the SUREQ bit = 1).
Write 1 to the SUREQ bit only when transmitting the setup token; for other purposes, write 0.
When the function controller is selected, be sure to write 0 to the SUREQ bit.

BSTS Flag (Buffer Status Flag)

Indicates whether DCP FIFO buffer access is enabled or disabled.
The meaning of the BSTS flag depends on the setting of ISEL bit in the port select register as shown below.

- When the ISEL bit = 0, the BSTS flag indicates whether the received data can be read from the buffer.
- When the ISEL bit = 1, the BSTS flag indicates whether the data to be transmitted can be written to the buffer.

38.2.28 Pipe Window Select Register (PIPESEL)

Address(es): 000A 0064h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PIPESEL[3:0]	Pipe Window Select	b3 b0 0 0 0 0: No pipe selected 0 0 0 1: PIPE1 0 0 1 0: PIPE2 0 0 1 1: PIPE3 0 1 0 0: PIPE4 0 1 0 1: PIPE5 0 1 1 0: PIPE6 0 1 1 1: PIPE7 1 0 0 0: PIPE8 1 0 0 1: PIPE9 Settings other than above are prohibited.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PIPE1 to PIPE 9 should be set using registers PIPESEL, PIPECFG, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEnTRE, and PIPEnTRN.

After selecting the pipe using the PIPESEL register, functions of the pipe should be set using registers PIPECFG, PIPEMAXP, and PIPEPERI. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set regardless of the pipe selection in the PIPESEL register.

PIPESEL[3:0] Bits (Pipe Window Select)

The PIPESEL[3:0] bits select the pipe number corresponding to registers PIPECFG, PIPEMAXP, and PIPEPERI which data are written to or read from.

Selecting a pipe number through the PIPESEL[3:0] bits allows writing to and reading from registers PIPECFG, PIPEMAXP, and PIPEPERI which correspond to the selected pipe number.

When PIPESEL[3:0] = 0000b, 0 is read from all of the bits in registers PIPECFG, PIPEMAXP, and PIPEPERI. Writing to these bits is invalid.

38.2.29 Pipe Configuration Register (PIPECFG)

Address(es): 000A 0068h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TYPE[1:0]		—	—	—	BFRE	DBLB	—	SHTNAK	—	—	DIR	EPNUM[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	EPNUM[3:0]	Endpoint Number*1	These bits specify the endpoint number for the selected pipe. Setting 0000b means an unused pipe.	R/W
b4	DIR	Transfer Direction*2,*3	0: Receiving direction 1: Transmitting direction	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: Pipe assignment continued at the end of transfer 1: Pipe assignment disabled at the end of transfer	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9	DBLB	Double Buffer Mode*2,*3	0: Single buffer 1: Double buffer	R/W
b10	BFRE	BRDY Interrupt Operation Specification*2,*3	0: BRDY interrupt upon transmitting or receiving data 1: BRDY interrupt upon completion of reading data	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	TYPE[1:0]	Transfer Type*1	• PIPE1 and PIPE2 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Isochronous transfer • PIPE3 to PIPE5 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Setting prohibited • PIPE6 to PIPE9 b15 b14 0 0: Pipe not used 0 1: Setting prohibited 1 0: Interrupt transfer 1 1: Setting prohibited	R/W

Note 1. Modify the TYPE[1:0], SHTNAK, and EPNUM[3:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

Note 2. Modify the BFRE, DBLB, and DIR bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), checking the PIPEnCTR.PBUSY flag through software is not necessary.

Note 3. To modify the BFRE, DBLB, and DIR bits after completing USB communication using the selected pipe, write 1 and then 0 to the PIPEnCTR.ACLRM bit continuously through software to clear the FIFO buffer assigned to the selected pipe while the PID[1:0] bits and CURPIPE[3:0] bits are in the state described in the above note 2.

The PIPECFG register specifies the transfer type, buffer memory access direction, and endpoint numbers for PIPE1 to PIPE9. It also selects single or double buffer mode, and whether to continue or disable pipe operation at the end of transfer.

EPNUM[3:0] Bits (Endpoint Number)

The EPNUM[3:0] bits specify the endpoint number for the selected pipe.
Setting 0000b means an unused pipe.
Do not make the settings such that the combination of the settings of the DIR and EPNUM[3:0] bits should be the same for two or more pipes (EPNUM[3:0] bits = 0000b can be set for all of the pipes).

DIR Bit (Transfer Direction)

The DIR bit specifies the transfer direction for the selected pipe.
When the DIR bit has been set to 0 by software, the USB uses the selected pipe in the receiving direction, and when software has set the DIR bit to 1, the USB uses the selected pipe in the transmitting direction.

SHTNAK Bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to modify the PID[1:0] bits to 00b (NAK) upon the end of transfer when the selected pipe is in the receiving direction.
The SHTNAK bit is valid when the selected pipe is PIPE1 to PIPE5 in the receiving direction.
When the SHTNAK bit has been set to 1 by software for the selected pipe in the receiving direction, the USB modifies the PIPEnCTR.PID[1:0] bits corresponding to the selected pipe to 00b (NAK) on determining the end of the transfer. The USB determines that the transfer has ended on any of the following conditions.

- A short packet (including a zero-length packet) is successfully received.
- The transaction counter is used and the number of packets specified by the counter are successfully received.

DBLB Bit (Double Buffer Mode)

The DBLB bit selects either single or double buffer mode for the FIFO buffer used by the selected pipe.
The DBLB bit is valid when PIPE1 to PIPE5 are selected.

BFRE Bit (BRDY Interrupt Operation Specification)

The BFRE bit specifies the BRDY interrupt generation timing from the USB to the CPU with respect to the selected pipe.
When the BFRE bit has been set to 1 by software and the selected pipe is in the receiving direction, the USB detects the transfer completion and generates the BRDY interrupt on having read the relevant packet.
When the BRDY interrupt is generated with the above conditions, 1 should be written to the BCLR bit in the port control register by software. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.
When the BFRE bit has been set to 1 by software and the selected pipe is in the transmitting direction, the USB does not generate the BRDY interrupt.
For details, refer to section 38.3.3.1, BRDY Interrupt.

TYPE[1:0] Bits (Transfer Type)

The TYPE[1:0] bits select the transfer type for the pipe selected by the PIPESEL.PIPESEL[3:0] bits (selected pipe).
Before setting the PID[1:0] bits to 01b (BUF) for the selected pipe (before starting USB communication using the selected pipe), set the TYPE[1:0] bits to a value other than 00b.

38.2.30 Pipe Maximum Packet Size Register (PIPEMAXP)

Address(es): 000A 006Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEVSEL[3:0]				—	—	—	MXPS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0/1 *1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	MXPS[8:0]	Maximum Packet Size*2	• PIPE1 and PIPE2: 1 byte (001h) to 256 bytes (100h) • PIPE3 to PIPE5: 8 bytes (008h), 16 bytes (010h), 32 bytes (020h), 64 bytes (040h) (Bits [8:7] and [2:0] are not provided.) • PIPE6 to PIPE9: 1 byte (001h) to 64 bytes (040h) (Bits [8:7] are not provided.)	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*3	b3 b0 0 0 0 0: Address 0000 0 0 0 1: Address 0001 0 0 1 0: Address 0010 0 0 1 1: Address 0011 0 1 0 0: Address 0100 0 1 0 1: Address 0101 Settings other than above are prohibited.	R/W

Note 1. The value of these bits is 0000h when no pipe is selected with the PIPESEL.PIPESEL[3:0] bits and 0040h when a pipe is selected.

Note 2. Modify the MXPS[8:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

Note 3. Modify the DEVSEL[3:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK). To modify these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The PIPEMAXP register specifies the maximum packet size for PIPE1 to PIPE9.

MXPS[8:0] Bits (Maximum Packet Size)

The MXPS[8:0] bits specify the maximum data payload (maximum packet size) for the selected pipe.

These bits should be set to the appropriate value for each transfer type based on USB Specification 2.0. Note that the maximum value of PIPE1 and PIPE2 is 256. While MXPS[8:0] = 000h, do not write to the FIFO buffer or do not set the PID[1:0] bits to 01b (BUF).

DEVSEL[3:0] Bits (Device Select)

When the host controller is selected, these bits specify the USB device address of the peripheral device which is the communication target.

The DEVSEL[3:0] bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in the DEVSEL[3:0] bits. For example, before setting the DEVSEL[3:0] bits to 0010b, the address should be set to DEVADD2.

When the function controller is selected, the DEVSEL[3:0] bits should be set to 0000b.

38.2.31 Pipe Cycle Control Register (PIPEPERI)

Address(es): 000A 006Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	IITV[2:0]	Interval Error Detection Interval *1	Specify the interval error detection timing for the selected pipe in terms of frames, which is expressed as nth power of 2.	R/W
b11 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	IFIS	Isochronous IN Buffer Flush	0: The buffer is not flushed. 1: The buffer is flushed.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Modify the IITV[2:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK). To modify these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The PIPEPERI register selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfer, and sets the interval error detection interval for PIPE1 to PIPE9.

IITV[2:0] Bits (Interval Error Detection Interval)

Before modifying the IITV[2:0] bits after USB communication has been completed with the IITV[2:0] bits set to a certain value, set the PID[1:0] bits to 00b (NAK) and then set the PIPEnCTR.ACLRM bit to 1 to initialize the interval timer.

The IITV[2:0] bits are invalid for PIPE3 to PIPE5; set the IITV[2:0] bits to 000b for PIPE3 to PIPE5.

IFIS Bit (Isochronous IN Buffer Flush)

Specifies whether to flush the buffer when the pipe selected by the PIPESEL.PIPESEL[3:0] bits (selected pipe) is used for isochronous IN transfers.

When the function controller is selected and the selected pipe is for isochronous IN transfers, the USB automatically clears the FIFO buffer when the USB fails to receive the IN token from the USB host within the interval set by the IITV[2:0] bits in terms of frames.

In double buffer mode (the PIPECFG.DBLB bit = 1), the USB only clears the data in the plane used earlier.

The USB clears the FIFO buffer on receiving the SOF packet immediately after the frame in which the USB has expected to receive the IN token. Even if the SOF packet is damaged, the USB also clears the FIFO buffer at the right timing to receive the SOF packet by using the internal interpolation function.

When the host controller is selected, set the IITV[2:0] bits to 000b.

When the selected pipe is not for isochronous transfer, set the IITV[2:0] bits to 000b.

38.2.32 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9)

- PIPEnCTR (n = 1 to 5)

Address(es): PIPE1CTR 000A 0070h, PIPE2CTR 000A 0072h, PIPE3CTR 000A 0074h, PIPE4CTR 000A 0076h, PIPE5CTR 000A 0078h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: The relevant pipe is not used for the transaction. 1: The relevant pipe is used for the transaction.	R
b6	SQMON	Sequence Toggle Bit Confirmation	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*2	0: Write disabled 1: Specifies DATA1.	R/W*1
b8	SQCLR	Sequence Toggle Bit Clear *2	0: Write disabled 1: Specifies DATA0.	R/W*1
b9	ACLRM	Auto Buffer Clear Mode*3	0: Disabled 1: Enabled (all buffers are initialized)	R/W
b10	ATREPM	Auto Response Mode*2	0: Auto response is disabled. 1: Auto response is enabled.	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	INBUFM	Transmit Buffer Monitor Flag	0: There is no data to be transmitted in the buffer memory. 1: There is data to be transmitted in the buffer memory.	R
b15	BSTS	Buffer Status Flag	0: Buffer access by the CPU is disabled. 1: Buffer access by the CPU is enabled.	R

Note 1. Only 0 can be read.
Note 2. Modify the ATREPM bit or write 1 to the SQCLR or SQSET bit while the PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
Note 3. Modify the ACLRM bit while PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying this bit after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The PIPEnCTR register can be set regardless of the pipe selection in the PIPESEL register.

PID[1:0] Bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the relevant pipe.
The default setting of the PID[1:0] bits are 00b (NAK). Modify the setting of the PID[1:0] bits to 01b (BUF) to use the relevant pipe for USB transfer. Table 38.6 and Table 38.7 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB depending on the PID[1:0] bit setting.
After modifying the setting of the PID[1:0] bits through software from 01b (BUF) to 00b (NAK) during USB communication using the relevant pipe, check that the PBUSY flag is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
The USB modifies the setting of the PID[1:0] bits in the following cases.

- The USB sets the PID[1:0] bits to 00b (NAK) on recognizing the completion of the transfer when the relevant pipe is in the receiving direction and the PIPECFG.SHTNAK bit for the selected pipe has been set to 1 by software.
- The USB sets the PID[1:0] bits to 11b (STALL) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a USB bus reset when the function controller is selected.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a receive error, such as a CRC error, three consecutive times when the host controller is selected.
- The USB sets the PID[1:0] bits to 11b (STALL) on receiving the STALL handshake when the host controller is selected.

To specify each response type, set the PID[1:0] bits as follows.

- To make a transition from NAK (00b) to STALL, set 10b.
- To make a transition from BUF (01b) to STALL, set 11b.
- To make a transition from STALL (11b) to NAK, set 10b and then 00b.
- To make a transition from STALL (11b) to BUF, set 10b, 00b, and then 01b.
- To make a transition from STALL (10b) to BUF, set 00b and then 01b.

PBUSY Flag (Pipe Busy Flag)

The PBUSY flag indicates whether the relevant pipe is being currently used or not for the transaction.
The USB modifies the PBUSY flag from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY flag from 1 to 0 upon completion of one transaction.
Reading the PBUSY flag after the PID[1:0] bits have been set to 00b (NAK) by software allows checking whether modification of the pipe settings is possible.
For details, refer to section 38.3.4.1, Pipe Control Register Switching Procedures.

SQMON Bit (Sequence Toggle Bit Confirmation)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
When the relevant pipe is not for the isochronous transfer, the USB allows the SQMON flag to toggle upon normal completion of the transaction. However, the SQMON flag is not allowed to toggle when a data PID mismatch occurs during the transfer in the receiving direction.

SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQSET bit to 1 through software allows the USB to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQSET bit to 0.

SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQCLR bit to 1 through software allows the USB to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQCLR bit to 0.

ACLRM Bit (Auto Buffer Clear Mode)

Enables or disables auto buffer clear mode for the relevant pipe.
To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously.
Table 38.8 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.

ATREPM Bit (Auto Response Mode)

Enables or disables auto response mode for the relevant pipe.
When the function controller is selected and the relevant pipe is for bulk transfer, the ATREPM bit can be set to 1.
When the ATREPM bit is set to 1, the USB responds to the token from the USB host as described below.
(1) When the relevant pipe is for bulk IN transfer (the PIPECFG.TYPE[1:0] bits = 01b and the PIPECFG.DIR bit = 1)
When the ATREPM bit = 1 and PID[1:0] = 01b (BUF), the USB transmits a zero-length packet in response to the IN token.
The USB updates (allows toggling of) the sequence toggle bit (data PID) each time the USB receives ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received.).
In this case, the USB does not generate the BRDY or BEMP interrupt.
(2) When the relevant pipe is for bulk OUT transfer (the PIPECFG.TYPE[1:0] bits = 01b and the PIPECFG.DIR bit = 0)
When the ATREPM bit = 1 and PID[1:0] = 01b (BUF), the USB returns NAK in response to the OUT token and generates the NRDY interrupt.
For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode.
When the relevant pipe is for isochronous transfer, be sure to set the ATREPM bit to 0.
When the host controller is selected, be sure to set the ATREPM bit to 0.

INBUFM Flag (Transmit Buffer Monitor Flag)

Indicates the relevant FIFO buffer status when the relevant pipe is in the transmitting direction.
When the relevant pipe is in the transmitting direction (the PIPECFG.DIR bit = 1), the USB sets the INBUFM flag to 1 when the CPU or DMAC/DTC completes writing data to at least one FIFO buffer plane.
The USB sets the INBUFM flag to 0 when the USB completes transmitting the data from the FIFO buffer plane to which all the data has been written. In double buffer mode (the PIPECFG.DBLB bit = 1), the USB sets the INBUFM flag to 0 when the USB completes transmitting the data from the two FIFO buffer planes before the CPU or DMAC/DTC completes writing data to one FIFO buffer plane.
The INBUFM flag indicates the same value as the BSTS flag when the relevant pipe is in the receiving direction (the PIPECFG.DIR bit = 0).

BSTS Flag (Buffer Status Flag)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS flag depends on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DnFIFOSEL.DCLRM bits as shown in Table 38.9.

Table 38.6 Operation of USB depending on PID[1:0] Bits Setting (When Host Controller is Selected)

PID[1:0] Bits	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB
00b (NAK)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.
01b (BUF)	Bulk or interrupt	Operation does not depend on the setting.	Issues tokens while the DVSTCTR0.UACT bit is 1 and the FIFO buffer corresponding to the relevant pipe is ready for transmission and reception. Does not issue tokens while the DVSTCTR0.UACT bit is 0 or the FIFO buffer corresponding to the relevant pipe is not ready for transmission or reception.
	Isochronous	Operation does not depend on the setting.	Issues tokens irrespective of the status of the FIFO buffer corresponding to the relevant pipe.
10b (STALL) or 11b (STALL)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.

Table 38.7 Operation of USB depending on PID[1:0] Bits Setting (When Function Controller is Selected)

PID[1:0] Bits	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB
00b (NAK)	Bulk or interrupt	Operation does not depend on the setting.	Returns NAK in response to the token from the USB host.
	Isochronous	Operation does not depend on the setting.	Returns nothing in response to the token from the USB host.
01b (BUF)	Bulk	Receiving direction (DIR bit = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Interrupt	Receiving direction (DIR bit = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Bulk or interrupt	Transmitting direction (DIR bit = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Returns NAK if not ready.
	Isochronous	Receiving direction (DIR bit = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception. Discards data if not ready.
	Isochronous	Transmitting direction (DIR bit = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Transmits the zero-length packet if not ready.
10b (STALL) or 11b (STALL)	Bulk or interrupt	Operation does not depend on the setting.	Returns STALL in response to the token from the USB host.
	Isochronous	Operation does not depend on the setting.	Returns nothing in response to the token from the USB host.

Table 38.8 Information Cleared by USB by Setting ACLRM = 1

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the relevant pipe (both FIFO buffer planes are cleared when double buffer mode is selected)	When the pipe is to be initialized
2	The interval count value when the relevant pipe is for isochronous transfer	When the interval count value is to be reset
3	Internal flags concerning the PIPECFG.BFRE bit	When the PIPECFG.BFRE setting is modified
4	FIFO buffer toggle control	When the PIPECFG.DBLB setting is modified
5	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

Table 38.9 Operation of BSTS Flag

DIR Bit	BFRE Bit	DCLRM Bit	BSTS Flag Function
0	0	0	The received data can be read from the FIFO buffer. The received data has been completely read from the FIFO buffer.
		1	Setting prohibited
	1	0	The received data can be read from the FIFO buffer. The BCLR bit in the port control register has been set to 1 by software after the received data has been completely read from the FIFO buffer.
		1	The received data can be read from the FIFO buffer. The received data has been completely read from the FIFO buffer.
1	0	0	The transmit data can be written to the FIFO buffer. The transmit data has been completely written to the FIFO buffer.
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

- PIPEnCTR (n = 6 to 9)

Address(es): PIPE6CTR 000A 007Ah, PIPE7CTR 000A 007Ch, PIPE8CTR 000A 007Eh, PIPE9CTR 000A 0080h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	—	—	—	—	—	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: The relevant pipe is not used at the USB bus. 1: The relevant pipe is used at the USB bus.	R
b6	SQMON	Sequence Toggle Bit Confirmation Flag	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W *1
b8	SQCLR	Sequence Toggle Bit Clear *2	0: Invalid 1: Specifies DATA0.	R/W *1
b9	ACLRM	Auto Buffer Clear Mode*2,*3	0: Auto buffer clear mode is disabled. 1: Auto buffer clear mode is enabled (all buffers are initialized)	R/W
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	BSTS	Buffer Status Flag	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only 0 can be read. Only 1 can be written.
Note 2. Write 1 to the SQCLR or SQSET bit while the PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
Note 3. Modify the ACLRM bit while the PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying this bit after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

PID[1:0] Bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the relevant pipe.
The default setting of the PID[1:0] bits are 00b (NAK). Modify the setting of the PID[1:0] bits to 01b (BUF) to use the relevant pipe for USB transfer. Table 38.6 and Table 38.7 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB depending on the setting of the PID[1:0] bits.
After modifying the setting of the PID[1:0] bits through software from 01b (BUF) to 00b (NAK) during USB communication using the relevant pipe, check that the PBUSY flag is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
The USB modifies the setting of the PID[1:0] bits in the following cases.

- The USB sets the PID[1:0] bits to 11b (STALL) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a USB bus reset when the function controller is selected.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a receive error, such as a CRC error, three consecutive times when the host controller is selected.
- The USB sets the PID[1:0] bits to 11b (STALL) on receiving the STALL handshake when the host controller is selected.

To specify each response type, set the PID[1:0] bits as follows.

- To make a transition from NAK (00b) to STALL, set 10b.
- To make a transition from BUF (01b) to STALL, set 11b.
- To make a transition from STALL (11b) to NAK, set 10b and then 00b.
- To make a transition from STALL (11b) to BUF, set 10b, 00b, and then 01b.
- To make a transition from STALL (10b) to BUF, set 00b and then 01b.

PBUSY Flag (Pipe Busy Flag)

The PBUSY flag indicates whether the relevant pipe is being currently used or not for the transaction.
The USB modifies the PBUSY flag from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY flag from 1 to 0 upon completion of one transaction.
Reading the PBUSY flag after the PID[1:0] bits has been set to 00b (NAK) by software allows checking whether modification of the pipe settings is possible.

SQMON Flag (Sequence Toggle Bit Confirmation Flag)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
The USB allows the SQMON flag to toggle upon normal completion of the transaction. However, the SQMON flag is not allowed to toggle when a data PID mismatch occurs during the transfer in the receiving direction.

SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQSET bit through software allows the USB to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQSET bit to 0.

SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQCLR bit to 1 through software allows the USB to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQCLR bit to 0.

ACLRM Bit (Auto Buffer Clear Mode)

Enables or disables auto buffer clear mode for the relevant pipe.

To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously.

Table 38.10 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.

BSTS Flag (Buffer Status Flag)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS flag depends on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DnFIFOSEL.DCLRM bits as shown in Table 38.9.

Table 38.10 Information Cleared by USB by Setting the ACLRM Bit = 1

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the selected pipe	When the pipe is to be initialized
2	The interval count value when the selected pipe is for interrupt transfer and the host controller is selected	When the interval count value is to be reset
3	Internal flags concerning the PIPECFG.BFRE bit	When the PIPECFG.BFRE setting is modified
4	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

38.2.33 PIPEn Transaction Counter Enable Register (PIPEnTRE) (n = 1 to 5)

Address(es): PIPE1TRE 000A 0090h, PIPE2TRE 000A 0094h, PIPE3TRE 000A 0098h, PIPE4TRE 000A 009Ch, PIPE5TRE 000A 00A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TRCLR	Transaction Counter Clear	0: Invalid 1: The current counter value is cleared.	R/W
b9	TRENB	Transaction Counter Enable	0: Transaction counter is disabled. 1: Transaction counter is enabled.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Modify each bit in the PIPEnTRE register while the PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

TRCLR Bit (Transaction Counter Clear)

Clears the current value of the transaction counter corresponding to the relevant pipe and then sets the TRCLR bit to 0.

TRENB Bit (Transaction Counter Enable)

Enables or disables the transaction counter.

For the pipe in the receiving direction, setting the TRENB bit to 1 after setting the total number of the packets to be received in the PIPEnTRN.TRNCNT[15:0] bits through software allows the USB to control hardware as described below on having received the number of packets equal to the setting of the TRNCNT[15:0] bits.

- While the PIPECFG.SHTNAK bit is 1, the USB modifies the PID[1:0] bits to 00b (NAK) for the corresponding pipe on having received the number of packets equal to the setting of the TRNCNT[15:0] bits.
- While the PIPECFG.BFRE bit is 1, the USB asserts the BRDY interrupt on having received the number of packets equal to the setting of the TRNCNT[15:0] bits and then reading the last received data.

For the pipe in the transmitting direction, set the TRENB bit to 0.

When the transaction counter is not used, set the TRENB bit to 0.

When the transaction counter is used, set the TRNCNT[15:0] bits before setting the TRENB bit to 1. Set the TRENB bit to 1 before receiving the first packet to be counted by the transaction counter.

38.2.34 PIPEn Transaction Counter Register (PIPEnTRN) (n = 1 to 5)

Address(es): PIPE1TRN 000A 0092h, PIPE2TRN 000A 0096h, PIPE3TRN 000A 009Ah, PIPE4TRN 000A 009Eh, PIPE5TRN 000A 00A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TRNCNT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TRNCNT[15:0]	Transaction Counter	• When written to: Specifies the total of packets (number of transactions) to be received in corresponding PIPE. • When read from: Indicates the specified number of transactions if the PIPEnTRE.TRENB bit is 0. Indicates the number of currently counted transactions if the PIPEnTRE.TRENB bit is 1.	R/W

The PIPEnTRN register retains the setting by a USB bus reset.

TRNCNT[15:0] Bits (Transaction Counter)

The USB increments the value of the TRNCNT[15:0] bits by one when all of the following conditions are satisfied on receiving the packet.

- The PIPEnTRE.TRENB bit = 1
- (TRNCNT[15:0] set value ≠ current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the PIPEMAXP.MXPS[8:0] bits.

The USB sets the value of the TRNCNT[15:0] bits to 0 when any of the following conditions are satisfied.

(1) All of the following conditions are satisfied.

- The PIPEnTRE.TRENB bit = 1
- (TRNCNT[15:0] set value = current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the PIPEMAXP.MXPS[8:0] bits.

(2) All of the following conditions are satisfied.

- The PIPEnTRE.TRENB bit = 1
- The USB has received a short packet.

(3) All of the following conditions are satisfied.

- The PIPEnTRE.TRENB bit = 1
- The PIPEnTRE.TRCLR bit has been set to 1 by software.

For the pipe in the transmitting direction, set the TRNCNT[15:0] bits to 0.
When the transaction counter is not used, set the TRNCNT[15:0] bits to 0.
Setting the number of transactions to be transferred to the TRNCNT[15:0] bits is only enabled when the PIPEnTRE.TRENB bit is 0. To modify the number of transactions to be transferred, set the TRCLR bit to 1 (to clear the current counter value) before setting the PIPEnTRE.TRENB bit to 1.

38.2.35 Device Address n Configuration Register (DEVADDn) (n = 0 to 5)

Address(es): DEVADD0 000A 00D0h, DEVADD1 000A 00D2h, DEVADD2 000A 00D4h, DEVADD3 000A 00D6h, DEVADD4 000A 00D8h, DEVADD5 000A 00DAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	USBSPD[1:0]		—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7, b6	USBSPD[1:0]	Transfer Speed of Communication Target Device	b7 b6 0 0: DEVADDn is not used 0 1: Low-speed 1 0: Full-speed 1 1: Setting prohibited	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The DEVADDn register specifies the transfer speed of the peripheral device which is the communication target for PIPE0 to PIPE9.

When the host controller is selected, the bits in the DEVADDn register should be set before starting communication using each pipe.

The bits in the DEVADDn register should be modified while no valid pipes are using the settings of the bits. Valid pipes refer to the pipes satisfying both of the following conditions:

- The DEVADDn register is selected by the DEVSEL[3:0] bits.
- The PID[1:0] bits are set to 01b (BUF) for the selected pipe or the selected pipe is the DCP with the DCPCTR.SUREQ bit set to 1.

USBSPD[1:0] Bits (Transfer Speed of Communication Target Device)

The USBSPD[1:0] bits specify the USB transfer speed of the communication target peripheral device.

Set these bits to 10b when a full-speed device is connected via the HUB.

When the host controller is selected, the USB refers to the setting of the USBSPD[1:0] bits to generate packets.

When the function controller is selected, set these bits to 00b.

38.2.36 PHY Cross Point Adjustment Register (PHYSLEW)

Address(es): 000A 00F0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	SLEWF01	SLEWF00	SLEWR01	SLEWR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	SLEWR00	Driver Cross Point Adjustment 00	0: When the host controller is selected. 1: When the function controller is selected.	R/W
b1	SLEWR01	Driver Cross Point Adjustment 01	0: When the function controller is selected. 1: When the host controller is selected.	R/W
b2	SLEWF00	Driver Cross Point Adjustment 00	Set this bit to 1.	R/W
b3	SLEWF01	Driver Cross Point Adjustment 01	0: When the function controller is selected. 1: When the host controller is selected.	R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The PHYSLEW register adjusts the cross point of the driver.
Set the value (0000000Eh when the host controller is selected or 00000005h when the function controller is selected) to the register before starting the USB operation.

38.2.37 Deep Standby USB Transceiver Control/Pin Monitoring Register (DPUSR0R)

Address(es): 000A 0400h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DVBSTS0	—	DOVCB0	DOVCA0	—	—	DM0	DP0
Value after reset:	0	0	0	0	0	0	0	0	x	0	x	x	0	0	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	FIXPHY0	DRPD0	—	RPUE0	SRPC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	SRPC0	USB Single End Receiver Control	0: Input through the D+ and D– inputs is disabled. 1: Input through the D+ and D– inputs is enabled.	R/W
b1	RPUE0*1	D+ Pull-Up Resistor Control	0: Disables D+ pull-up resistor. 1: Enables D+ pull-up resistor.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	DRPD0*1	D+/D– Pull-Down Resistor Control	0: Disables D+/D– pull-down resistor. 1: Enables D+/D– pull-down resistor.	R/W
b4	FIXPHY0	USB Transceiver Output Fix	0: The outputs are fixed in normal mode and on return from deep software standby mode. 1: The outputs are fixed on transitions to deep software standby mode.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	DP0	USB D+ Input Flag	Indicates the D+ input signal of the USB.	R
b17	DM0	USB D– Input Flag	Indicates the D– input signal of the USB.	R
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	DOVCA0	USB OVRCURA Input Flag	Indicates the OVRCURA input signal of the USB.	R
b21	DOVCB0	USB OVRCURB Input Flag	Indicates the OVRCURB input signal of the USB.	R
b22	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b23	DVBSTS0	USB VBUS Input Flag	Indicates the VBUS input signal of the USB.	R
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Use this bit during operation in deep standby mode. For details, refer to section 38.3.1.5, Release from Deep Software Standby Mode Due to USB Suspend/Resume Interrupts.

SRPC0 Bit (USB Single End Receiver Control)

The SRPC0 bit controls the D+ and D– inputs of the USB transceiver.

This bit is only valid when the FIXPHY0 bit is 1.

FIXPHY0 Bit (USB Transceiver Output Fix)

The FIXPHY0 bit keeps the outputs of the USB transceiver disabled.

38.2.38 Deep Standby USB Suspend/Resume Interrupt Register (DPUSR1R)

Address(es): 000A 0404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DVBINT0	—	DOVRCRB0	DOVRCRA0	—	—	DMINT0	DPINT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	DVBSE0	—	DOVRCRBE0	DOVRCRAE0	—	—	DMINTE0	DPINTE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DPINTE0	USB D+ Interrupt Enable/Clear	0: Recovery from deep software standby mode is disabled. 1: Recovery from deep software standby mode is enabled.	R/W
b1	DMINTE0	USB D– Interrupt Enable/Clear	0: Recovery from deep software standby mode is disabled. 1: Recovery from deep software standby mode is enabled.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DOVRCRAE0	USB OVRCURA Interrupt Enable/Clear	0: Recovery from deep software standby mode is disabled. 1: Recovery from deep software standby mode is enabled.	R/W
b5	DOVRCRBE0	USB OVRCURB Interrupt Enable/Clear	0: Recovery from deep software standby mode is disabled. 1: Recovery from deep software standby mode is enabled.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	DVBSE0	USB VBUS Interrupt Enable/Clear	0: Recovery from deep software standby mode is disabled. 1: Recovery from deep software standby mode is enabled.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	DPINT0	USB D+ Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode. 1: The system has returned from deep software standby mode.	R
b17	DMINT0	USB D– Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode. 1: The system has returned from deep software standby mode.	R
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	DOVRCRA0	USB OVRCURA Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode. 1: The system has returned from deep software standby mode.	R
b21	DOVRCRB0	USB OVRCURB Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode. 1: The system has returned from deep software standby mode.	R
b22	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b23	DVBINT0	USB VBUS Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode. 1: The system has returned from deep software standby mode.	R
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DPINTE0 Bit (USB D+ Interrupt Enable/Clear)

This bit enables or disables triggering of recovery from deep software standby mode by the D+ input of the USB. Writing 0 to the DPINTE0 bit while the DPINT0 flag is 1 sets the DPINT0 flag to 0.

DMINTE0 Bit (USB D– Interrupt Enable/Clear)

This bit enables or disables triggering of recovery from deep software standby mode by the D– input of the USB. Writing 0 to the DMINTE0 bit while the DMINT0 flag is 1 sets the DMINT0 flag to 0.

DOVRCRAE0 Bit (USB OVRCURA Interrupt Enable/Clear)

This bit enables or disables triggering of recovery from deep software standby mode by the OVRCURA input of the USB. Writing 0 to the DOVRCRAE0 bit while the DOVRCRA0 flag is 1 sets the DOVRCRA0 flag to 0.

DOVRCRBE0 Bit (USB OVRCURB Interrupt Enable/Clear)

This bit enables or disables triggering of recovery from deep software standby mode by the OVRCURB input of the USB. Writing 0 to the DOVRCRBE0 bit while the DOVRCRB0 flag is 1 sets the DOVRCRB0 flag to 0.

DVBSE0 Bit (USB VBUS Interrupt Enable/Clear)

This bit enables or disables triggering of recovery from deep software standby mode by the VBUS input of the USB. Writing 0 to the DVBSE0 bit while the DVBINT0 flag is 1 sets the DVBINT0 flag to 0.

DPINT0 Flag (USB D+ Interrupt Source Recovery Flag)

This bit indicates that the system has returned from deep software standby mode due to the D+ input of the USB. Recovery from deep software standby mode due to D+ input is only enabled when the DPINTE0 bit is 1.
Writing 0 to the DPINTE0 bit while the DPINT0 flag is 1 sets the DPINT0 flag to 0.

DMINT0 Flag (USB D– Interrupt Source Recovery Flag)

This bit indicates that the system has returned from deep software standby mode due to the D– input of the USB. Recovery from deep software standby mode due to D– input is only enabled when the DMINTE0 bit is 1.
Writing 0 to the DPINTE0 bit while the DMINT0 flag is 1 sets the DMINT0 flag to 0.

DOVRCRA0 Flag (USB OVRCURA Interrupt Source Recovery Flag)

This bit indicates that the system has returned from deep software standby mode due to the OVRCURA input of the USB. Recovery from deep software standby mode due to OVRCURA input is only enabled when the DOVRCRAE0 bit is 1.
Writing 0 to the DOVRCRAE0 bit while the DOVRCRA0 flag is 1 sets the DOVRCRA0 flag to 0.

DOVRCRB0 Flag (USB OVRCURB Interrupt Source Recovery Flag)

This bit indicates that the system has returned from deep software standby mode due to the OVRCURB input of the USB. Recovery from deep software standby mode due to OVRCURB input is only enabled when the DOVRCRBE0 bit is 1.
Writing 0 to the DOVRCRBE0 bit while the DOVRCRB0 flag is 1 sets the DOVRCRB0 flag to 0.

DVBINT0 Flag (USB VBUS Interrupt Source Recovery Flag)

This indicates that the system has returned from deep software standby mode due to the VBUS input of the USB. Recovery from deep software standby mode due to VBUS input is only enabled when the DVBSE0 bit is 1.
Writing 0 to the DVBSE0 bit while the DVBINT0 flag is 1 sets the DVBINT0 flag to 0.

38.3 Operation

38.3.1 System Control

This section describes the register settings that are necessary for initialization of this module and power consumption control.

38.3.1.1 Setting Data to the USB Related Register

Setting the SYSCFG.USBE bit to 1 after starting the clock supply to the USB (SYSCFG.SCKE bit = 1) enables and starts USB operation.

38.3.1.2 Controller Function Selection

For the USB, the host or function controller can be selected using the SYSCFG.DCFM bit. Note that the DCFM bit should be modified in the initial settings immediately after a reset is released or when pulling up of the D+ line and pulling down of the D+ and D– lines are disabled (the SYSCFG.DPRPU bit = 0 and DRPD bit = 0).

38.3.1.3 Controlling USB Data Bus Resistors

The USB has pull-up and pull-down resistors for the D+ and D– lines. Pull up or pull down these lines by setting the SYSCFG.DPRPU and DRPD bits.
When the function controller is selected, confirm that connection to the USB host is made, then set the SYSCFG.DPRPU bit to 1 and pull up the D+ line (during full-speed).
When the SYSCFG.DPRPU bit is set to 1 during communication with the PC, the USB module disables the pull-up resistor of the USB data line, thus notifying the USB host of disconnection.
When the host controller is selected, set the SYSCFG.DRPD bit and pull down the D+ and D– lines.

Table 38.11 USB Data Bus Resistor Control

SYSCFG register				
DRPD bit	**DPRPU bit**	**D–**	**D+**	**Function**
0	0	Open	Open	Not in use
0	1	Open	Pull-up	When operating as the function controller (in full-speed)
1	0	Pull-down	Pull-down	When operating as the host controller
1	1	—	—	Setting prohibited

38.3.1.4 Example of USB External Connection Circuit

Figure 38.2 shows an example of OTG connection of the USB connector in the self-powered state.
The USB controls the signals for enabling a pull-up resistor for the D+ signal and pull-down resistors for the D+ and D– signals. These signals can be pulled up or down using the SYSCFG.DPRPU and SYSCFG.DRPD bits.
When the function controller is selected and the DPRPU bit is set to 0 during communication with the host controller, the pull-up resistor of the USB data line is disabled, making it possible to notify the USB host of the device disconnection.

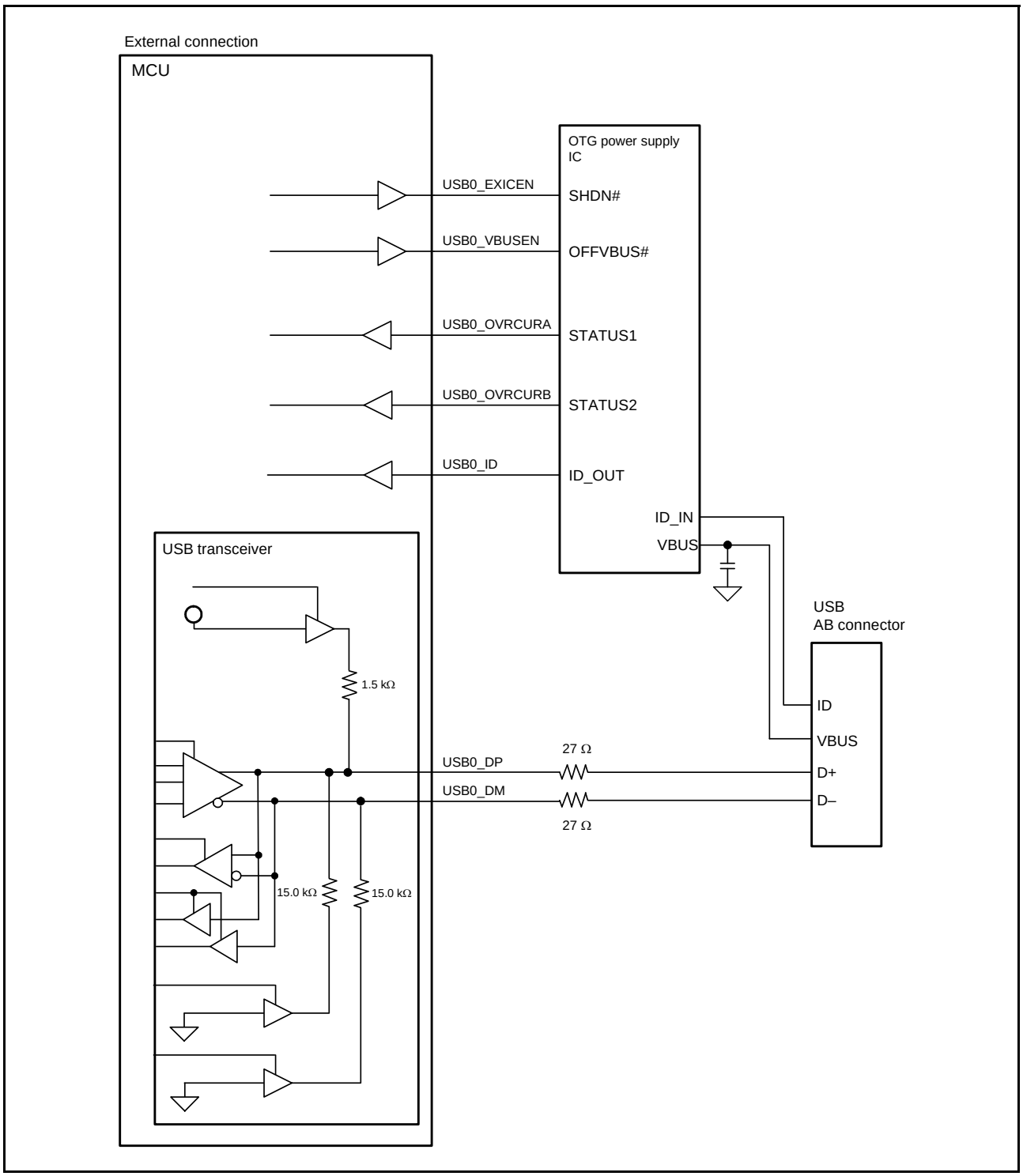


Figure 38.2 Sample OTG Connection of USB Connector in Self-Powered State

Figure 38.3 shows an example of functional connection of the USB connector in the self-powered state.

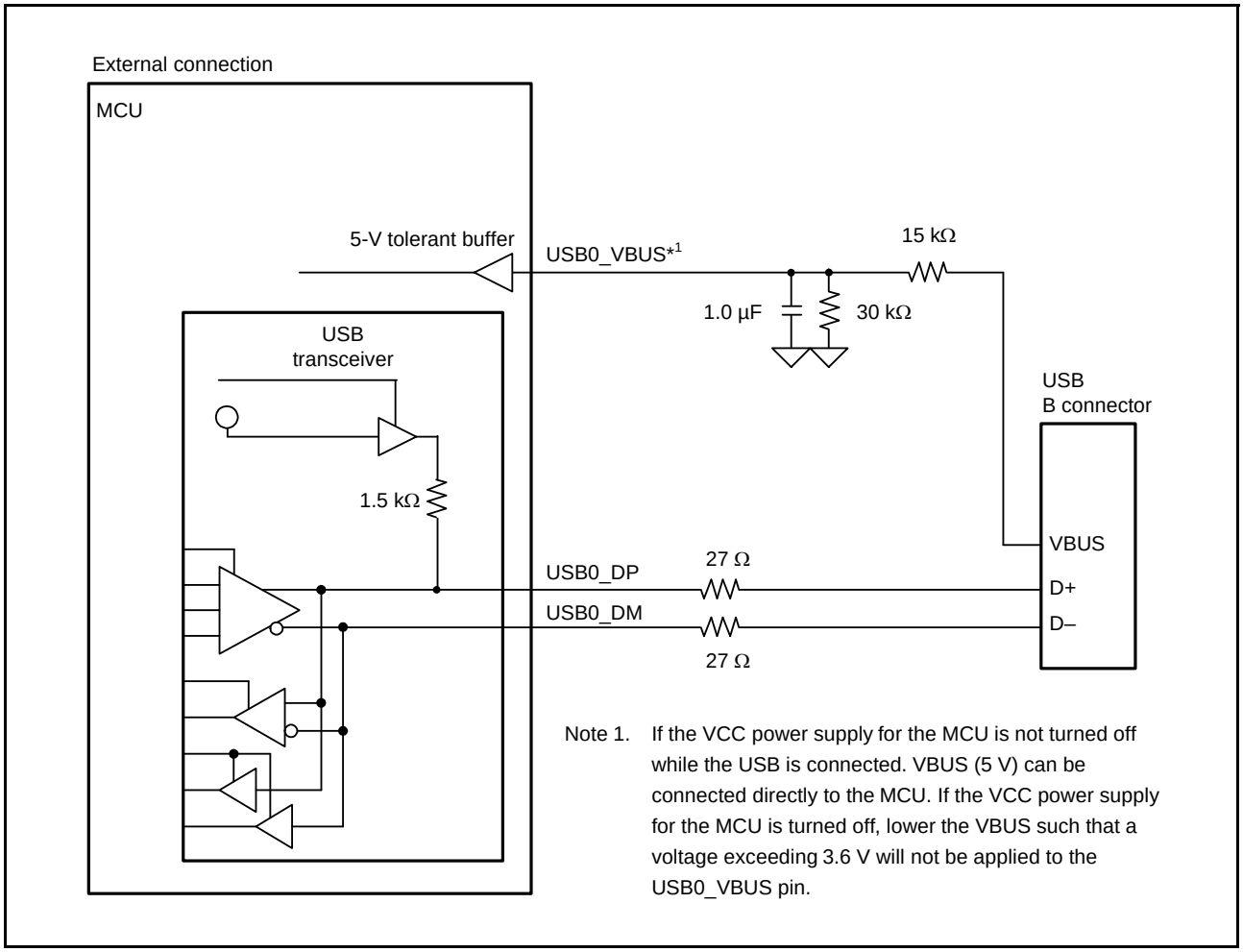


Figure 38.3 Functional Connection of USB Connector in Self-Powered State

Figure 38.4 shows an example of host connection of the USB connector.

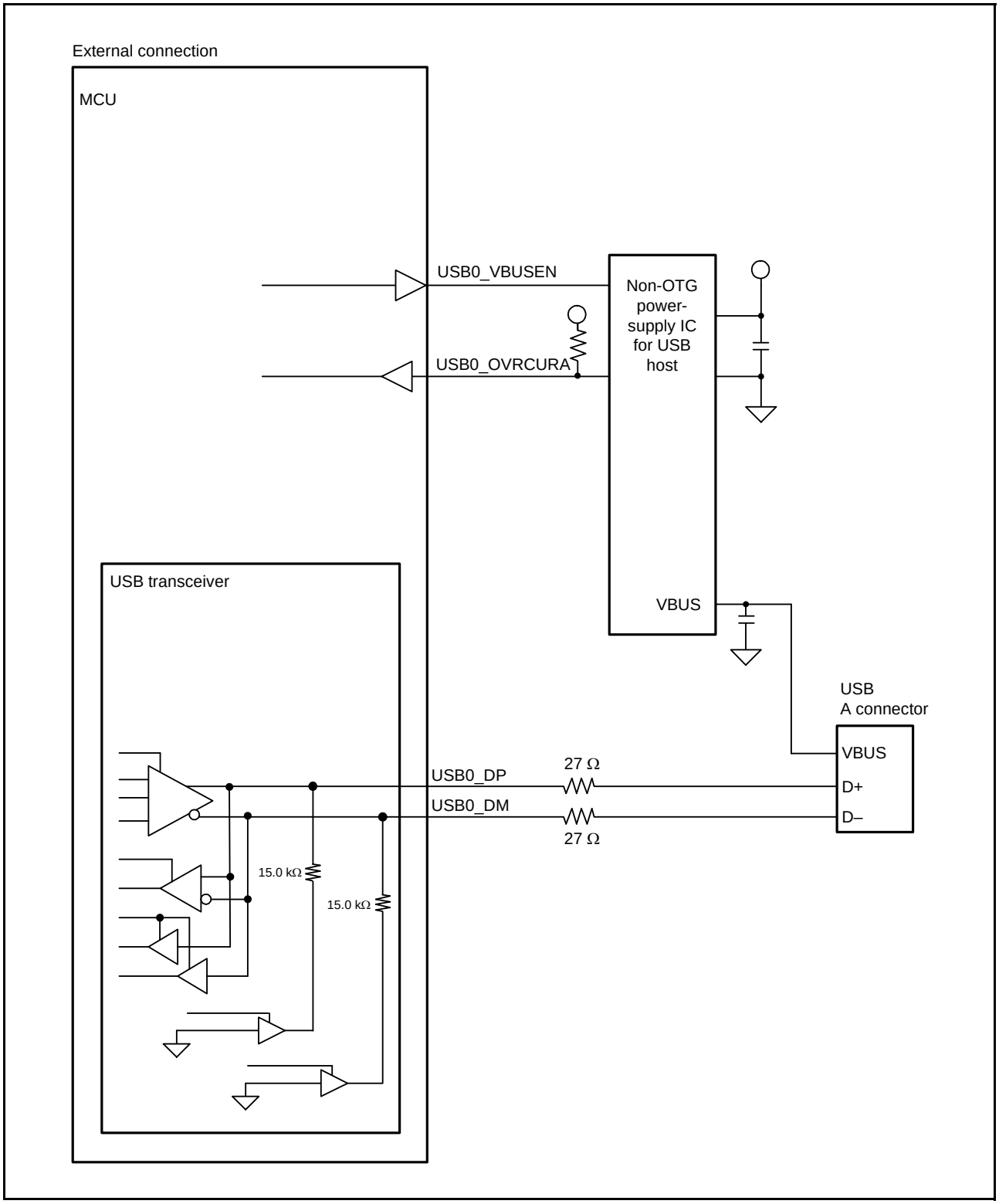


Figure 38.4 Sample Host Connection of USB Connector

Figure 38.5 shows an example of functional connection of the USB connector in bus powered state.

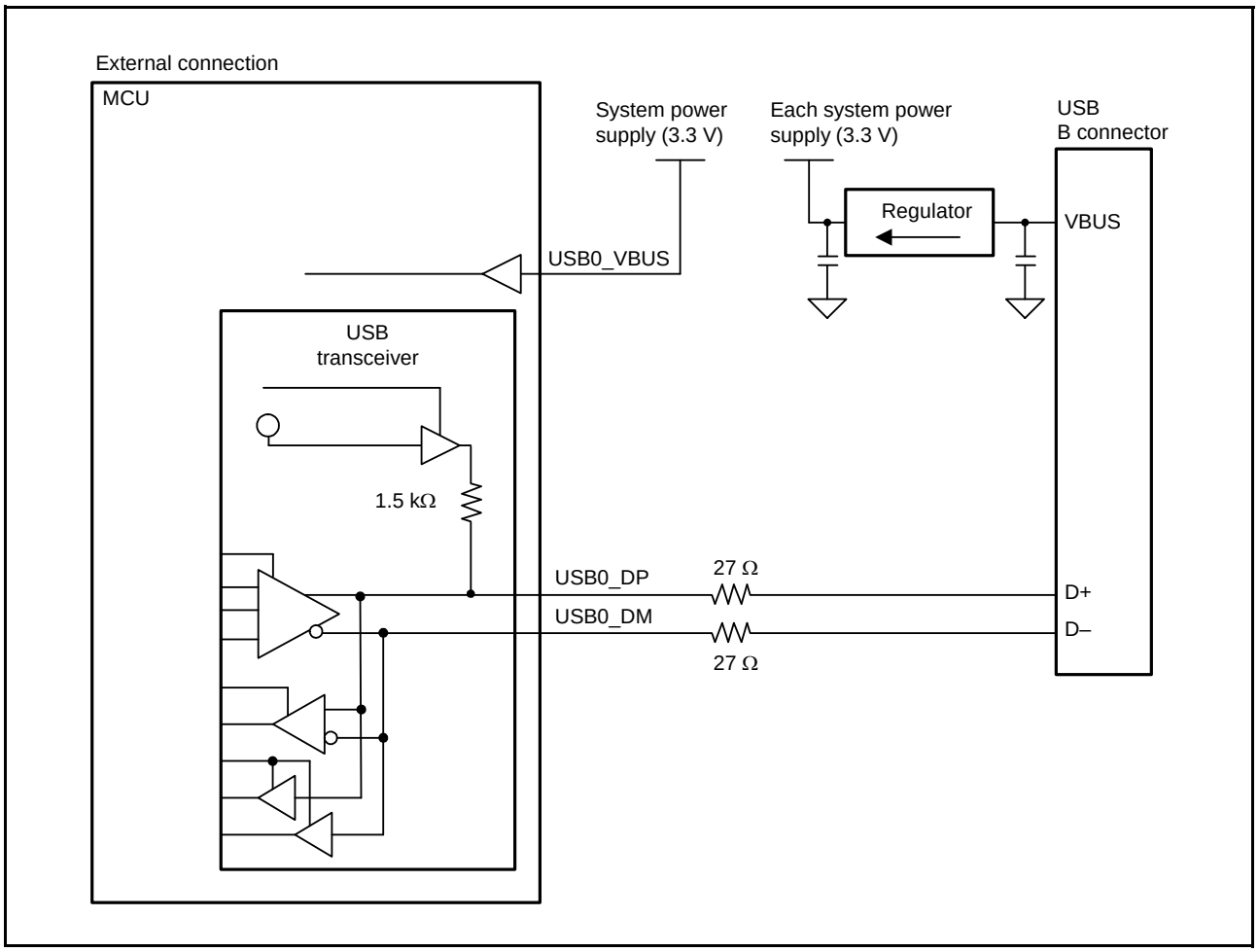


Figure 38.5 Functional Connection Sample of USB Connector in Bus Powered State

The examples of external circuits given in this section are simplified circuits, and their operation in every system is not guaranteed.

38.3.1.5 Release from Deep Software Standby Mode Due to USB Suspend/Resume Interrupts

Deep software standby mode can be canceled by a USB suspend/resume interrupt.
A USB suspend/resume interrupt is detected at the USB resume detecting unit. The USB resume detecting unit controls and monitors the I/O pins for USB to detect USB suspend/resume interrupts.
Figure 38.6 shows a schematic diagram of connection between the USB resume detecting unit and the I/O pins for USB.

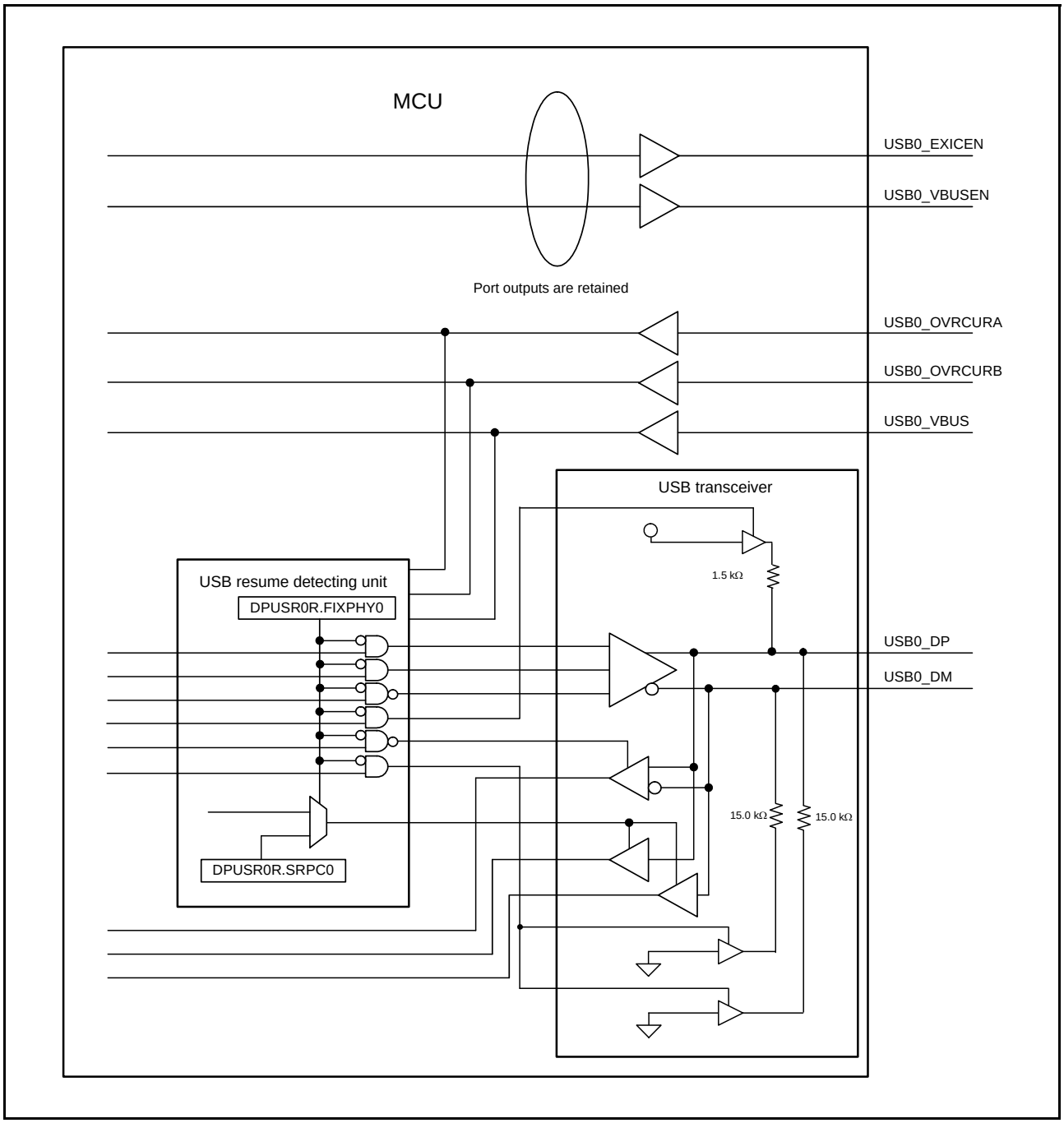


Figure 38.6 Overview of Connection between USB Resume Detecting Unit and USB I/O Pins

Table 38.12 shows the USB suspend/resume interrupt sources and their corresponding I/O pins.

Table 38.12 USB Suspend/Resume Interrupt Sources and Corresponding I/O Pins

USB Operating Mode	Source	Pin Name
Function/OTG	Resume	USB0_DP
Host/OTG	Connection/Disconnection	USB0_DP/USB0_DM
Function	Connection/Disconnection	USB0_VBUS
Host	Overcurrent detection	USB0_OVRCURA
OTG	Overcurrent detection	USB0_OVRCURA/USB0_OVRCURB

Figure 38.7 shows a flowchart for setting the USB when entering deep software standby mode while the host controller or function controller is selected. Figure 38.8 shows a flowchart for setting the USB when canceling deep software standby mode while the host controller is selected. Figure 38.9 shows a flowchart for setting the USB when canceling deep software standby mode while the function controller is selected.

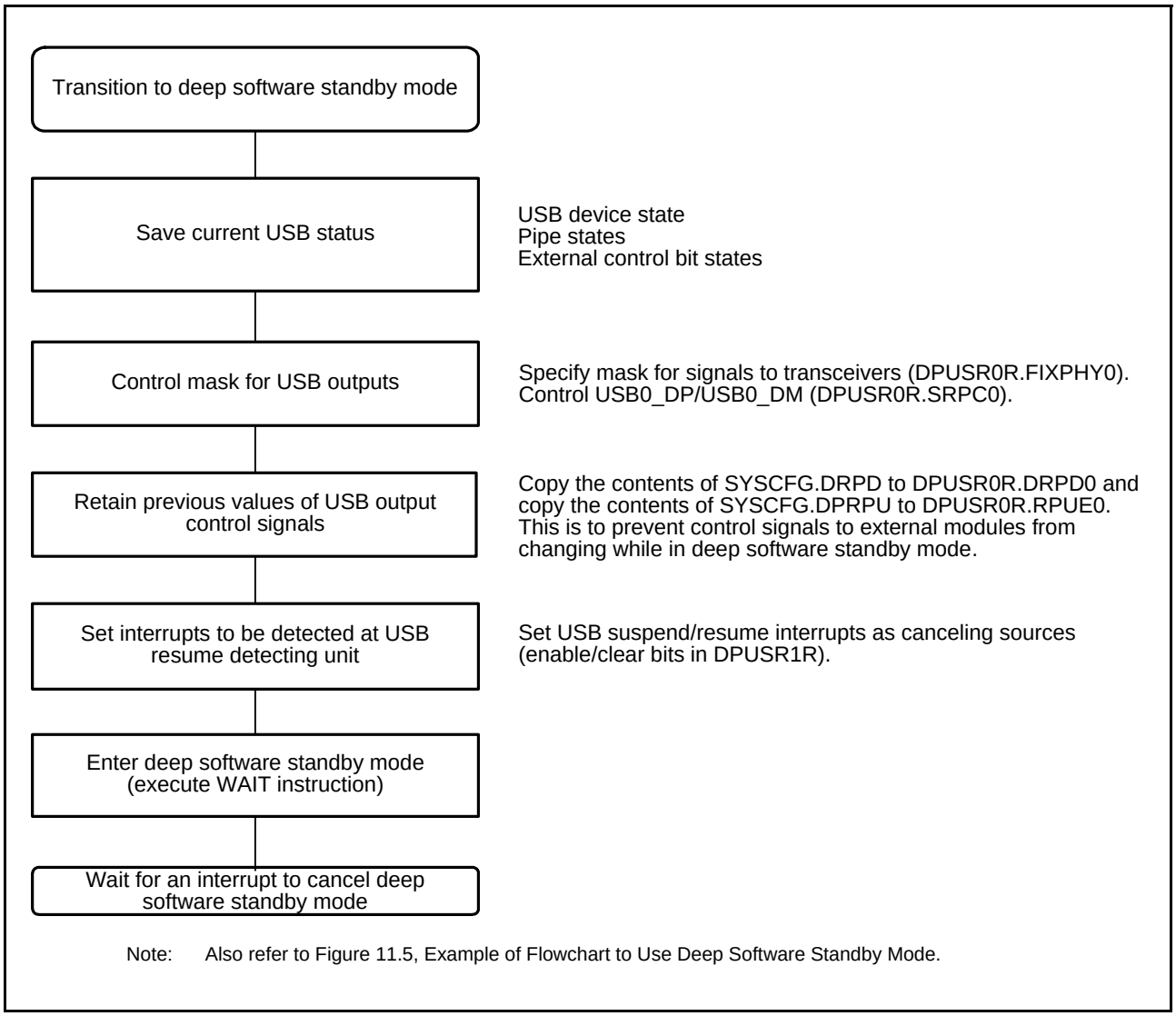


Figure 38.7 USB Setting Flowchart for Transition to Deep Software Standby Mode as Host/Function Controller

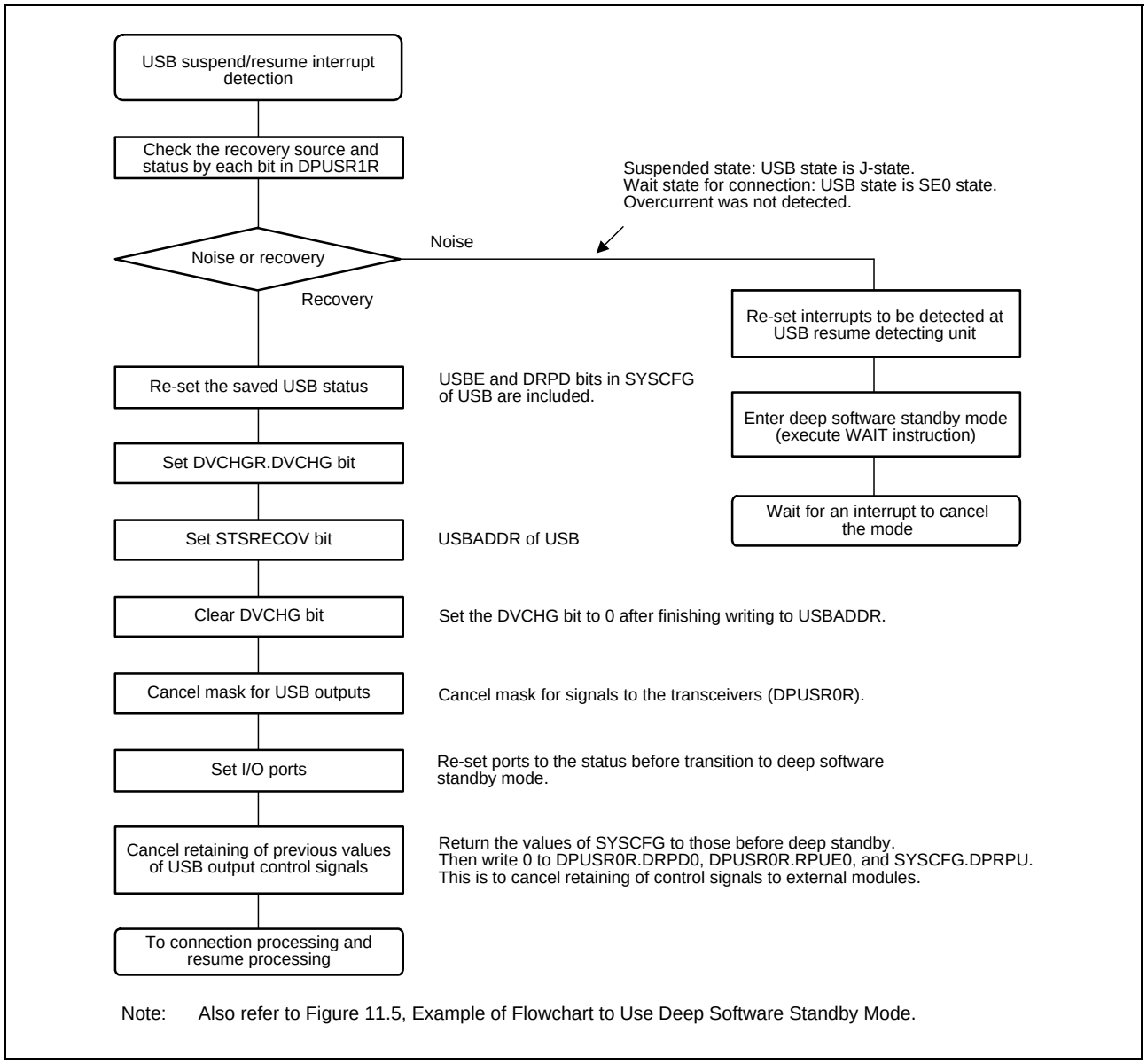
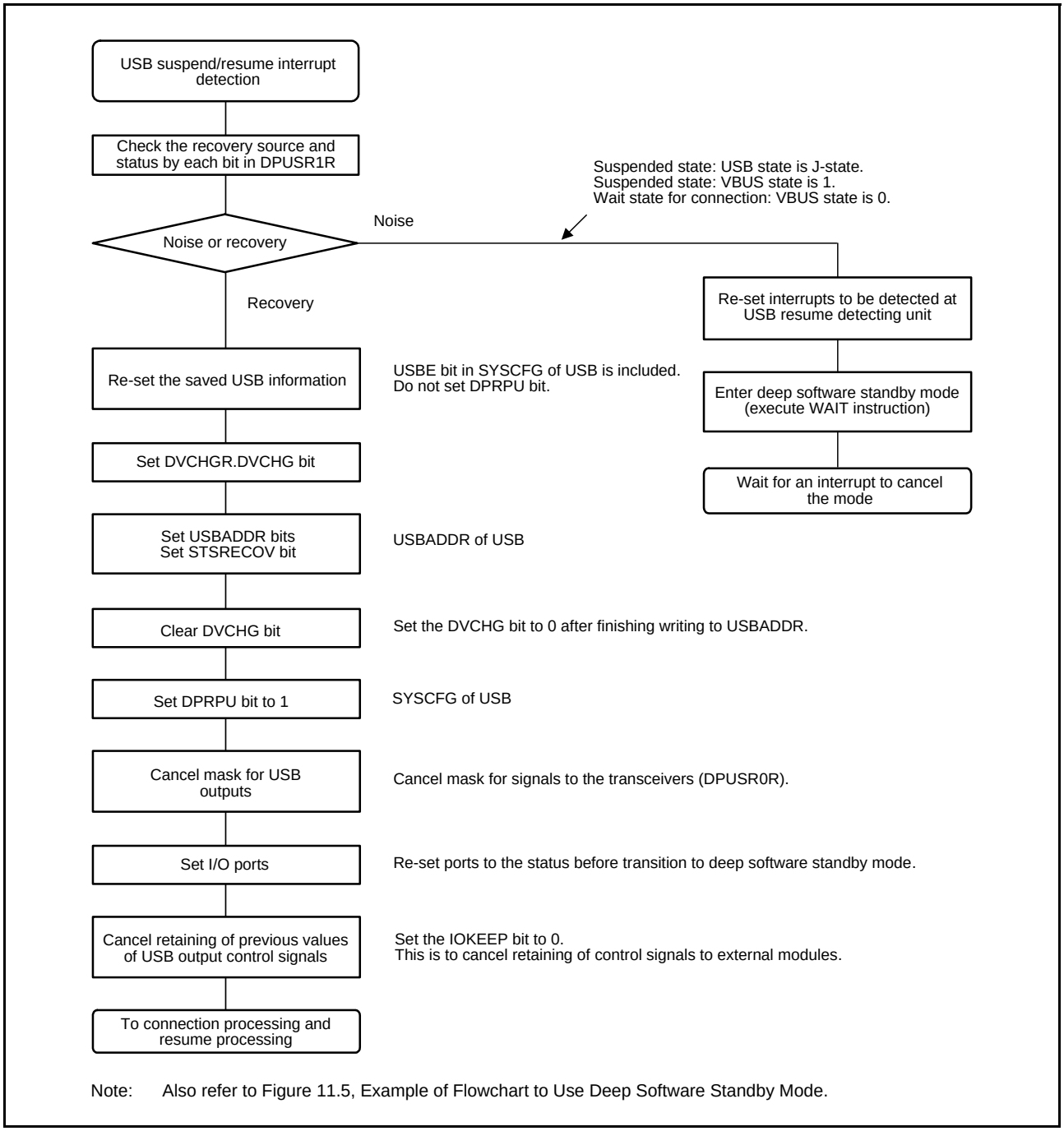


Figure 38.8 USB Setting Flowchart for Canceling Deep Software Standby Mode as Host Controller



Note: Also refer to Figure 11.5, Example of Flowchart to Use Deep Software Standby Mode.

Figure 38.9 USB Setting Flowchart for Canceling Deep Software Standby Mode as Function Controller

38.3.2 Interrupt Sources

Table 38.13 lists the interrupt sources in the USB.

When an interrupt generation condition is satisfied and the interrupt output is enabled using the corresponding interrupt enable register, a USB interrupt request is issued the Interrupt Controller (ICU) and an USB interrupt will be generated.

Table 38.13 Interrupt Sources

Bit to be Set	Name	Interrupt Source	Function That Generates the Interrupt	Status Flag
VBINT	VBUS interrupt	• When a change in the state of the USB0_VBUS input pin has been detected (low to high or high to low)	Host/function*1	INTSTS0. VBSTS
RESM	Resume interrupt	• When a change in the state of the USB bus has been detected in the suspended state (J-state to K-state or J-state to SE0)	Function	—
SOFR	Frame number update interrupt	[Host controller is selected] • When an SOF packet with a different frame number has been transmitted [Function controller is selected] • When an SOF packet with a different frame number has been received	Host/function	—
DVST	Device state transition interrupt	• When a device state transition has been detected (any of the following conditions) A USB bus reset detected Suspend state detected SET_ADDRESS request received SET_CONFIGURATION request received	Function	INTSTS0. DVSQ[2:0]
CTRT	Control transfer stage transition interrupt	• When a stage transition has been detected in control transfer (any of the following conditions) Setup stage completed Control write transfer status stage transition Control read transfer status stage transition Control transfer completed A control transfer sequence error occurred	Function	INTSTS0. CTSQ[2:0]
BEMP	Buffer empty interrupt	• When transmission of all data in the buffer memory has been completed and the buffer has become empty • When a packet larger than the maximum packet size has been received	Host/function	BEMPSTS. PIPEnBEMP
NRDY	Buffer not ready interrupt	[Host controller is selected] • When STALL has been received from the peripheral device for the issued token • When a response has not been received correctly from the peripheral device for the issued token (no response was returned three consecutive times or a packet reception error occurred three consecutive times) • When an overrun/underrun occurred during isochronous transfer [Function controller is selected] • When NAK has been returned for an IN or OUT token while the PID[1:0] bits are 01b (BUF) • When a CRC error or a bit stuffing error occurred during data reception in isochronous transfer • When an overrun/underrun occurred during data reception in isochronous transfer	Host/function	NRDYSTS. PIPEnNRDY
BRDY	Buffer ready interrupt	• When the buffer has become ready (reading or writing is enabled)	Host/function	BRDYSTS. PIPEnBRDY
OVRCR	Overcurrent input change interrupt	• When a change in the state of the USB0_OVRCURA or USB0_OVRCURB input pin has been detected (low to high or high to low)	Host	INTSTS1. OVRCR
BCHG	Bus change interrupt	• When a change of USB bus state has been detected	Host/function	SYSSTS0. LNST[1:0]
DTCH	Disconnection detection during full-speed operation	• When disconnection of a peripheral device has been detected in full-speed operation	Host	DVSTCTR0. RHST[2:0]
ATTCH	Device connection detection	• When J-state or K-state is detected on the USB port for 2.5 µs. Used for checking whether a peripheral device is connected.	Host	—
EOFERR	EOF error detection	• When an EOF error of a peripheral device has been detected	Host	—
SACK	Normal setup operation	• When the normal response (ACK) for the setup transaction has been received	Host	—
SIGN	Setup error	• When a setup transaction error (no response or ACK packet corruption) was detected three consecutive times	Host	—

Note 1. Though this interrupt can be generated while the host function is selected, it is not usually used with the host function.

Figure 38.10 shows the circuits related to the interrupts in the USB.

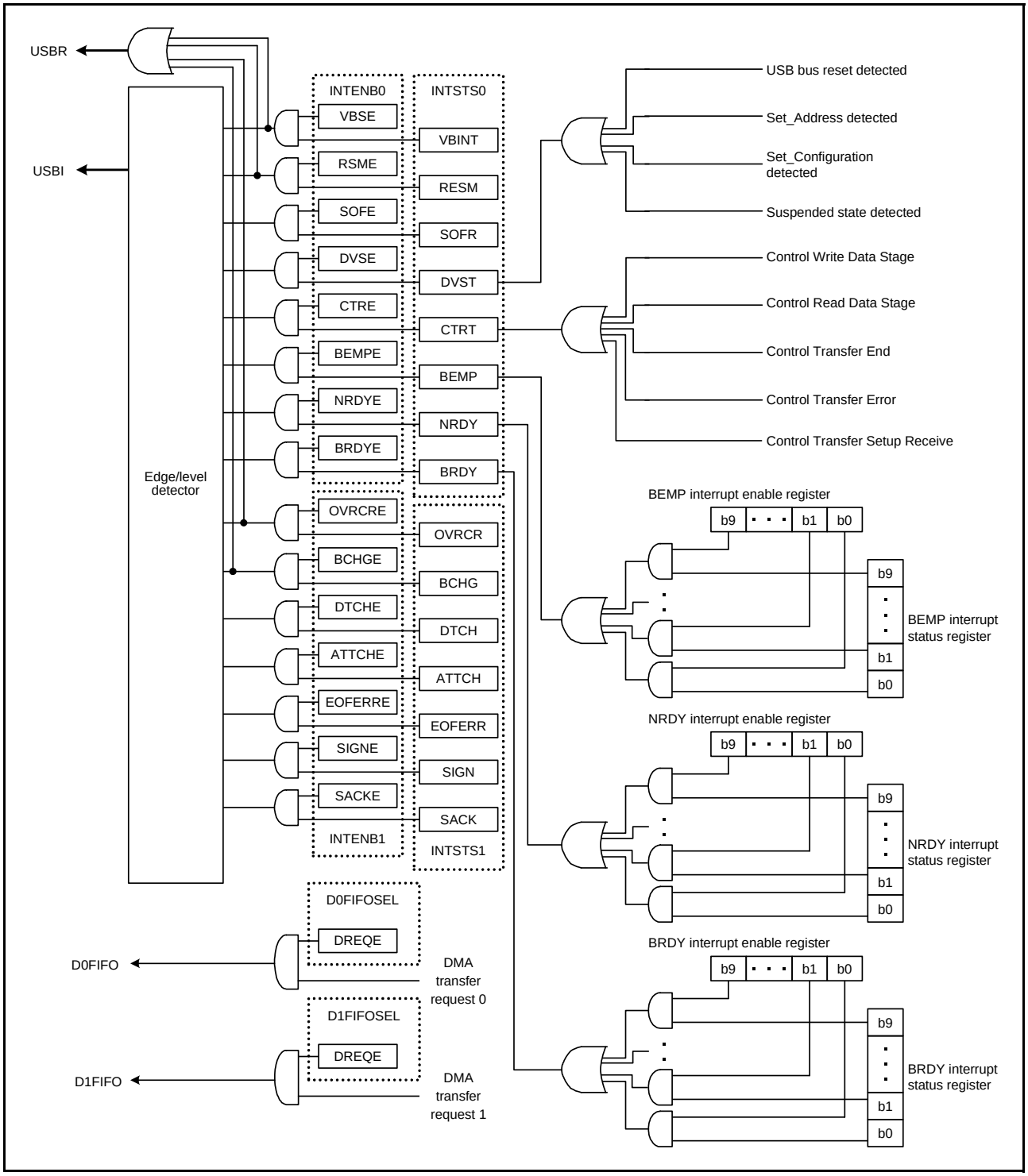


Figure 38.10 Circuits Related to Interrupts in USB

Table 38.14 shows the interrupts generated in the USB0.

Table 38.14 USB Interrupts

Interrupt Name	Interrupt Status Flag	DTC Activation	DMAC Activation	Priority
D0FIFO	DMA/DTC transfer request 0	Possible	Possible	High
D1FIFO	DMA/DTC transfer request 1	Possible	Possible	
USBI	VBUS interrupt, resume interrupt, frame number update interrupt, device state transition interrupt, control transfer stage transition interrupt, buffer empty interrupt, buffer not ready interrupt, buffer ready interrupt, overcurrent input change interrupt, bus change interrupt, disconnection detection during full-speed operation, device connection detection, EOF error detection, normal setup operation, and setup error	Not possible	Not possible	Low
USBR	VBUS interrupt, resume interrupt, overcurrent input change interrupt, and bus change interrupt	Not possible	Not possible	—

38.3.3 Interrupt Descriptions

38.3.3.1 BRDY Interrupt

The BRDY interrupt is generated when either of the host controller or function controller is selected. The following shows the conditions under which the USB sets 1 to a corresponding bit in the BRDYSTS register. Under this condition, the USB generates a BRDY interrupt if software has set 1 to the BRDYENB.PIPEnBRDYE bit that corresponds to the pipe and 1 to the INTENB0.BRDYE bit.
The conditions for generating and clearing the BRDY interrupt depend on the settings of the SOFCFG.BRDYM bit and PIPECFG.BFRE bit for each pipe as described below.

(1) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.
On any of the following conditions, the USB generates an internal BRDY interrupt request trigger and sets 1 to the BRDYSTS.PIPEnBRDY flag corresponding to the pertinent pipe.

(a) For the pipe in the transmitting direction:

- When the DIR bit is changed from 0 to 1 by software.
- When packet transmission is completed using the pertinent pipe while write-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS flag is read as 0).
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode.
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is completed.
- When the hardware flushes the buffer of the pipe for isochronous transfers.
- When 1 is written to the PIPEnCTR.ACLRM bit, which causes the FIFO buffer to make transition from the write-disabled to write-enabled state.

No request trigger is generated for the DCP (that is, during data transmission for control transfers).

(b) For the pipe in the receiving direction:

- When packet reception is completed successfully thus enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS flag is read as 0).
  No request trigger is generated for the transaction in which data PID mismatch has occurred.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode.
  No request trigger is generated until completion of reading data from the currently-read FIFO buffer even if reception by the other FIFO buffer is completed.

When the function controller is selected, the BRDY interrupt is not generated in the status stage of control transfers.
The BRDY interrupt status of the pertinent pipe can be set to 0 by writing 0 to the corresponding PIPEnBRDY flag through software. In this case, 1s should be written to the PIPEnBRDY flags for the other pipes.
Clear the BRDY status before accessing the FIFO buffer.

(2) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 1

With these settings, the USB generates a BRDY interrupt on completion of reading all data for a single transfer using the pipe in the receiving direction, and sets 1 to the bit in the BRDYSTS register corresponding to the pertinent pipe.
On any of the following conditions, the USB determines that the last data for a single transfer has been received.

- When a short packet including a zero-length packet is received.
- When the PIPEn transaction counter register (PIPEnTRN) is used and the number of packets specified by the PIPEnTRN.TRNCNT[15:0] bits are completely received.

When the pertinent data is completely read after any of the above conditions has been satisfied, the USB determines that all data for a single transfer has been completely read.
When a zero-length packet is received while the FIFO buffer is empty, the USB module determines that all data for a single transfer has been completely read when the FRDY flag in the FIFO port control register is 1 and the DTLN[8:0] flags are 0. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding port control register through software.
With these settings, the USB does not detect a BRDY interrupt for the pipe in the transmitting direction.
The BRDY interrupt status of the pertinent pipe can be set to 0 by writing 0 to the corresponding BRDYSTS.PIPEnBRDY flag through software. In this case, 1s should be written to the PIPEnBRDY flags for the other pipes.
In this mode, the PIPECFG.BFRE bit setting should not be modified until all data for a single transfer has been processed. When it is necessary to modify the PIPECFG.BFRE bit before completion of processing, all FIFO buffers for the pertinent pipe should be cleared using the PIPEnCTR.ACLRM bit.

(3) When the SOFCFG.BRDYM Bit = 1 and the PIPECFG.BFRE Bit = 0

With these settings, the BRDYSTS.PIPEnBRDY values are linked to the BSTS flag setting for each pipe. In other words, the BRDY interrupt status flags (PIPEnBRDY) are set to 1 or 0 by the USB depending on the FIFO buffer status.

(a) For the pipe in the transmitting direction:

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for write access, and are set to 0 when it is not ready.
However, the BRDY interrupt is not generated even if the DCP in the transmitting direction is ready for write access.

(b) For the pipe in the receiving direction:

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for read access, and are set to 0 when all data have been read (not ready for read access).
When a zero-length packet is received while the FIFO buffer is empty, the pertinent bit is set to 1 and the BRDY interrupt is continuously generated until BCLR = 1 is written through software.
With this setting, the PIPEnBRDY flag cannot be set to 0 through software.
When the SOFCFG.BRDYM bit is set to 1, all PIPECFG.BFRE bits (for all pipes) should be set to 0.

Figure 38.11 shows the timing of BRDY interrupt generation.

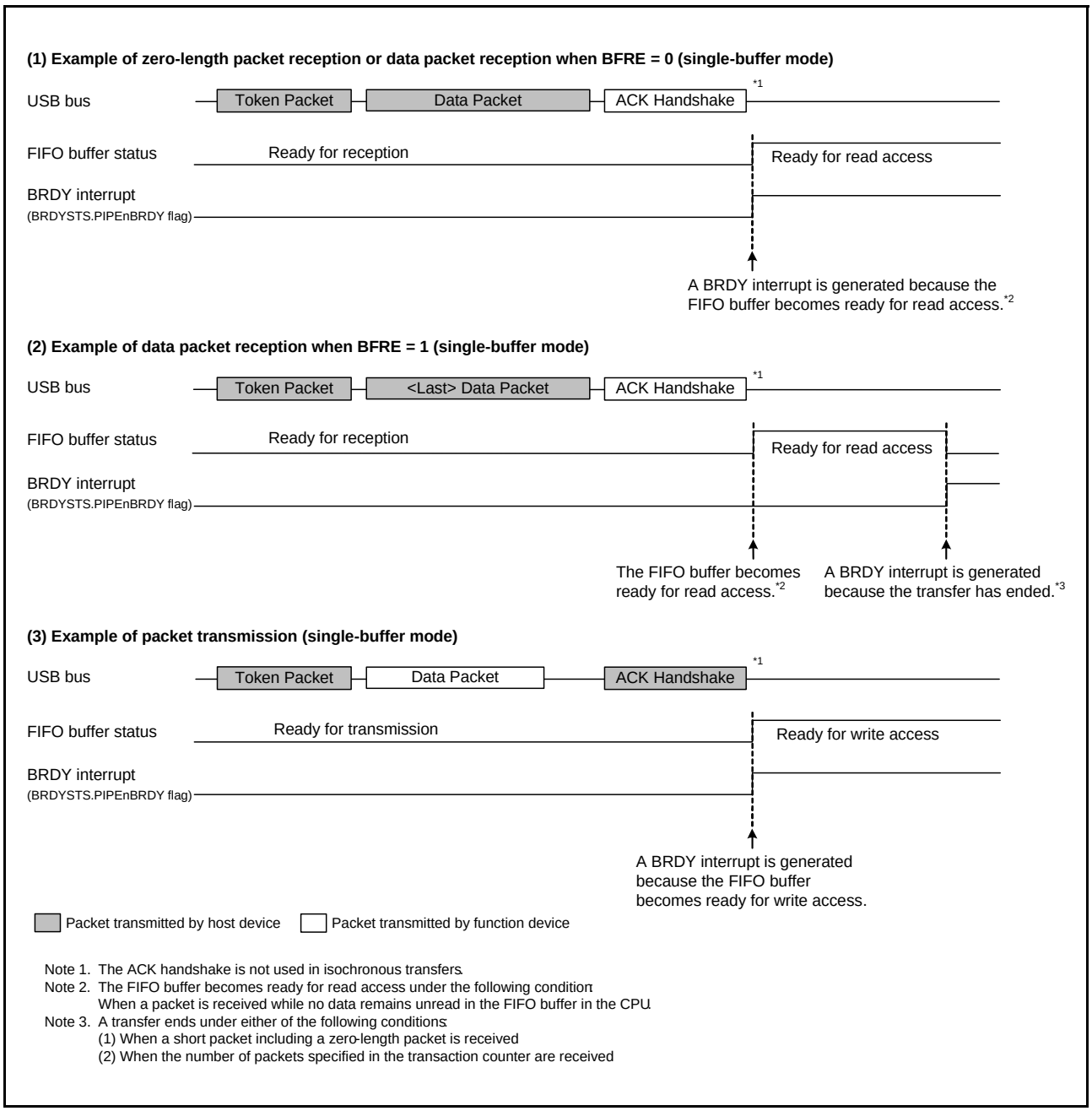


Figure 38.11 Timing of BRDY Interrupt Generation

The condition that USB clears the INTSTS0.BRDY flag depends on the SOFCFG.BRDYM bit setting. Table 38.15 shows the condition for clearing the BRDY flag.

Table 38.15 Condition for Clearing BRDY Flag

BRDYM Bit	Condition for Clearing BRDY Flag
0	The USB sets the BRDY flag to 0 when all bits in the BRDYSTS register have been set to 0 by software.
1	The USB sets the BRDY flag to 0 when the BSTS flags for all pipes have become 0.

38.3.3.2 NRDY Interrupt

On generating an internal NRDY interrupt request for the pipe whose PID[1:0] bits are set to 01b (BUF) by software, the USB sets the corresponding NRDYSTS.PIPEnNRDY flag to 1. If the corresponding bit in the NRDYENB register has been set to 1 by software, the USB sets the INTSTS0.NRDY flag to 1 and generates a USB interrupt.
The following describes the conditions on which the USB generates the internal NRDY interrupt request for a given pipe.
Note that the internal NRDY interrupt request is not generated during setup transaction execution when the host controller is selected. During setup transactions when the host controller is selected, the SACK or SIGN interrupt is detected.
The internal NRDY interrupt request is not generated during status stage execution of the control transfer when the function controller is selected.

(1) When Host Controller is Selected

(a) For the pipe in the transmitting direction:

On any of the following conditions, the USB detects an NRDY interrupt.

- For the pipe for isochronous transfers, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer.
  In this case, the USB transmits a zero-length packet following the OUT token and sets the bit corresponding to the NRDYSTS.PIPEnNRDY flag and the FRMNUM.OVRN flag to 1.
- During communications other than setup transactions using the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USB sets the bit corresponding to the PIPEnNRDY flag to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 00b (NAK).
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device.
  In this case, the USB sets the bit corresponding to the PIPEnNRDY flag to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL).

(b) For the pipe in the receiving direction:

- For the pipe for isochronous transfers, when the time to issue an IN token comes while there is no space available in the FIFO buffer.
  In this case, the USB discards the received data for the IN token and sets the PIPEnNRDY flag corresponding to the pipe and the OVRN flag to 1.
  When a packet error is detected in the received data for the IN token, the USB also sets the FRMNUM.CRCE flag to 1.
- For the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device for the IN token issued by the USB (when timeout is detected before detection of the DATA packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 00b (NAK).
- For the pipe for isochronous transfers, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the DATA packet from the peripheral device) or an error is detected in the packet from the peripheral device.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe to 1. (The setting of the PID[1:0] bits of the pipe is not modified.)

- For the pipe for isochronous transfers, when a CRC error or a bit stuffing error is detected in the received data packet.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe and the CRCE flag to 1.
- When the STALL handshake is received.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL).

(2) When Function Controller is Selected

(a) For the pipe in the transmitting direction:

- When an IN token is received while there is no data to be transmitted in the FIFO buffer.
  In this case, the USB generates a NRDY interrupt request at the reception of the IN token and sets the NRDYSTS.PIPEnNRDY flag to 1.
  For the pipe for the isochronous transfers in which an interrupt is generated, the USB transmits a zero-length packet and sets the FRMNUM.OVRN flag to 1.

(b) For the pipe in the receiving direction:

- When an OUT token is received while there is no space available in the FIFO buffer.
  For the pipe for the isochronous transfers in which an interrupt is generated, the USB generates a NRDY interrupt request at the reception of the OUT token and sets the PIPEnNRDY flag to 1 and OVRN flag to 1.
  For the pipe for the transfers other than isochronous transfers in which an interrupt is generated, the USB generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the PIPEnNRDY flag to 1.
  However, during re-transmission (due to data PID mismatch), the NRDY interrupt request is not generated. In addition, if an error occurs in the DATA packet, the NRDY interrupt request is not generated.
- For the pipe for isochronous transfers, when a token is not received successfully within an interval frame.
  In this case, the USB generates a NRDY interrupt request when SOF is received, and sets the PIPEnNRDY flag to 1.

Figure 38.12 shows the timing of NRDY interrupt generation when the function controller is selected.

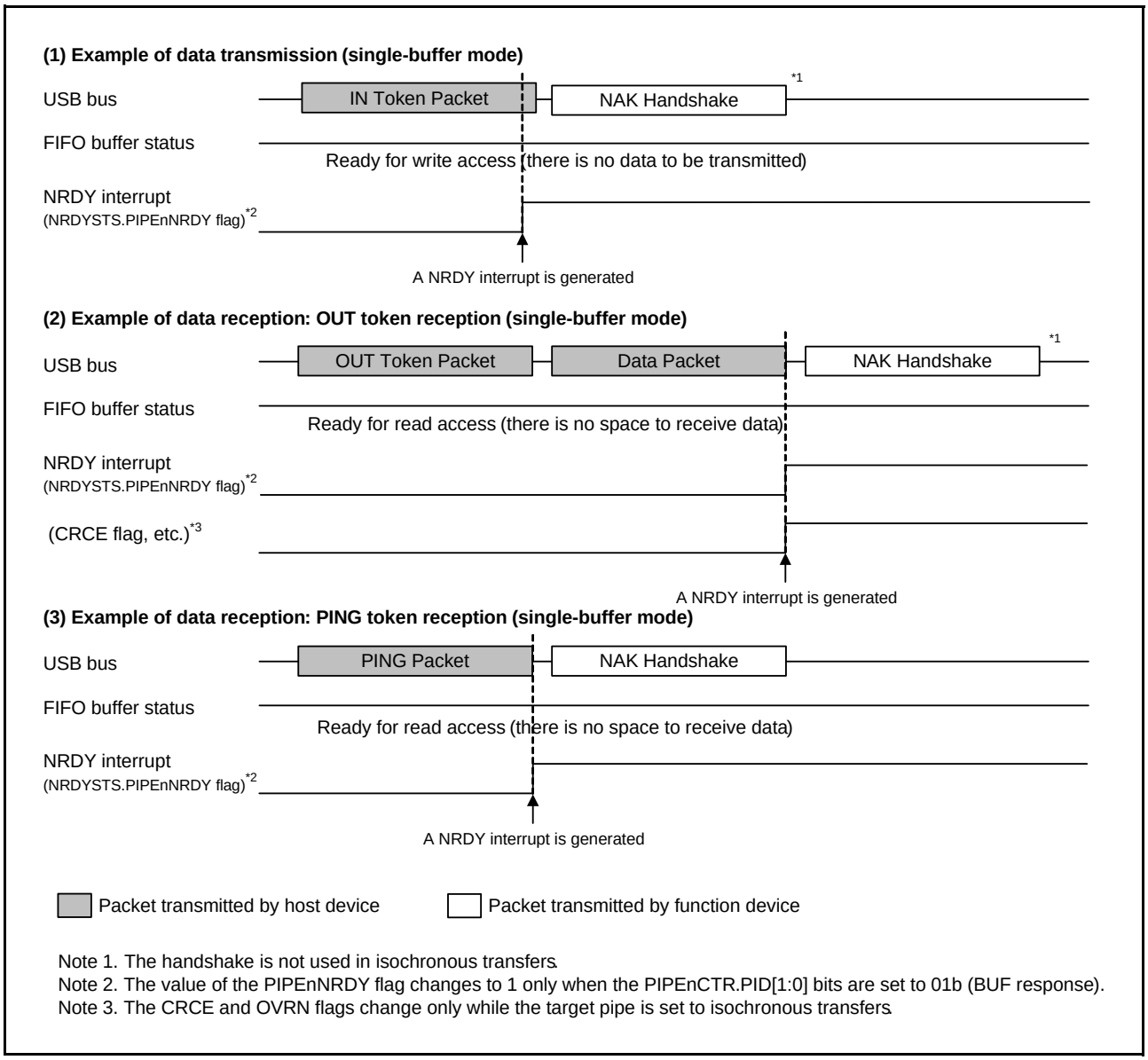


Figure 38.12 Timing of NRDY Interrupt Generation (When Function Controller is Selected)

38.3.3.3 BEMP Interrupt

On detecting a BEMP interrupt for the pipe whose PID[1:0] bits are set to 01b (BUF) by software, the USB sets the corresponding BEMPSTS.PIPEnBEMP flag to 1. If the corresponding bit in the BEMPENB register has been set to 1 by software, the USB sets the INTSTS0.BEMP flag to 1 and generates a USB interrupt.

The following describes the conditions on which the USB generates an internal BEMP interrupt request.

(1) For the pipe in the transmitting direction:

When the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission).
In single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for the pipe other than DCP. However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When the CPU or DMAC/DTC has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode.
- When the buffer is cleared (emptied) by setting the PIPEnCTR.ACLRM or the BCLR bit in the port control register to 1.
- When IN transfer (zero-length packet transmission) is performed during the control transfer status stage while the function controller is selected.

(2) For the pipe in the receiving direction:

When the successfully-received data packet size exceeds the specified maximum packet size.
In this case, the USB generates a BEMP interrupt request, sets the corresponding BEMPSTS.PIPEnBEMP flag to 1, discards the received data, and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL). Here, the USB returns no response when used as the host controller, and returns STALL response when used as the function controller.
However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When a CRC error or a bit stuffing error is detected in the received data.
- When a setup transaction is being performed,
  Writing 0 to the BEMPSTS.PIPEnBEMP flag clears the status.
  Writing 1 to the BEMPSTS.PIPEnBEMP flag has no effect.

Figure 38.13 shows the timing of BEMP interrupt generation when the function controller is selected.

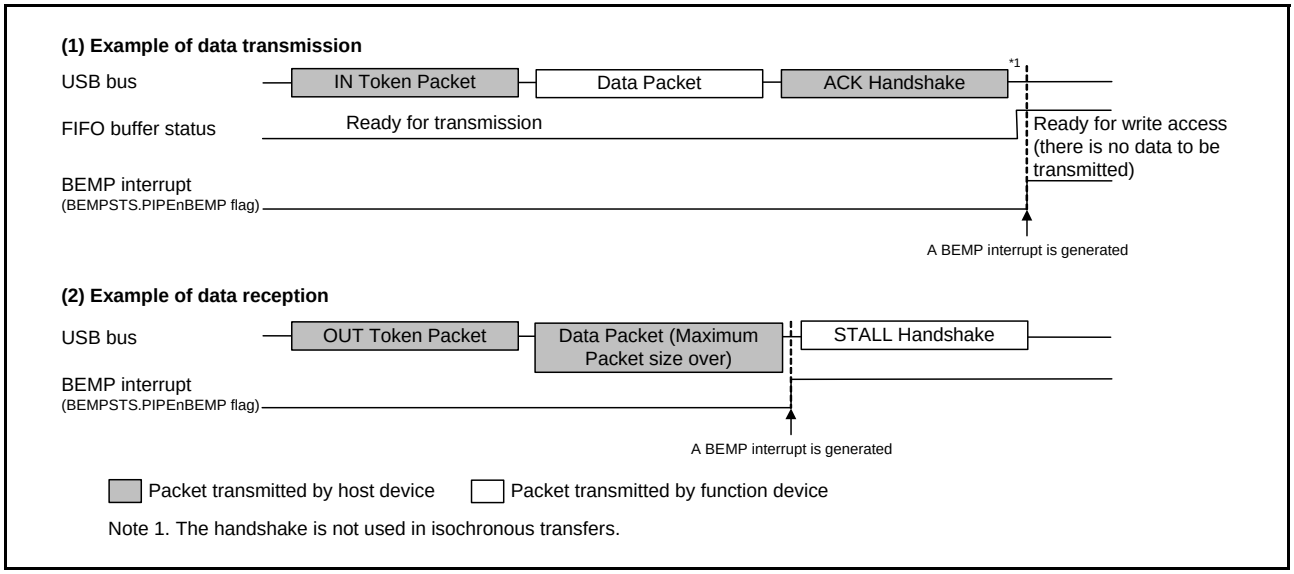


Figure 38.13 Timing of BEMP Interrupt Generation (When Function Controller is Selected)

38.3.3.4 Device State Transition Interrupt

Figure 38.14 is a diagram of device state transitions in the USB. The USB controls device state and generates device state transition interrupts. However, recovery from the suspended state (resume signal detection) is detected by means of the resume interrupt. The device state transition interrupts can be enabled or disabled individually using INTENB0. The device state to which a transition was made can be confirmed using the INTSTS0.DVSQ[2:0] flags.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

Device state can be controlled only when the function controller is selected. The device state transition interrupts can also be generated only when the function controller is selected.

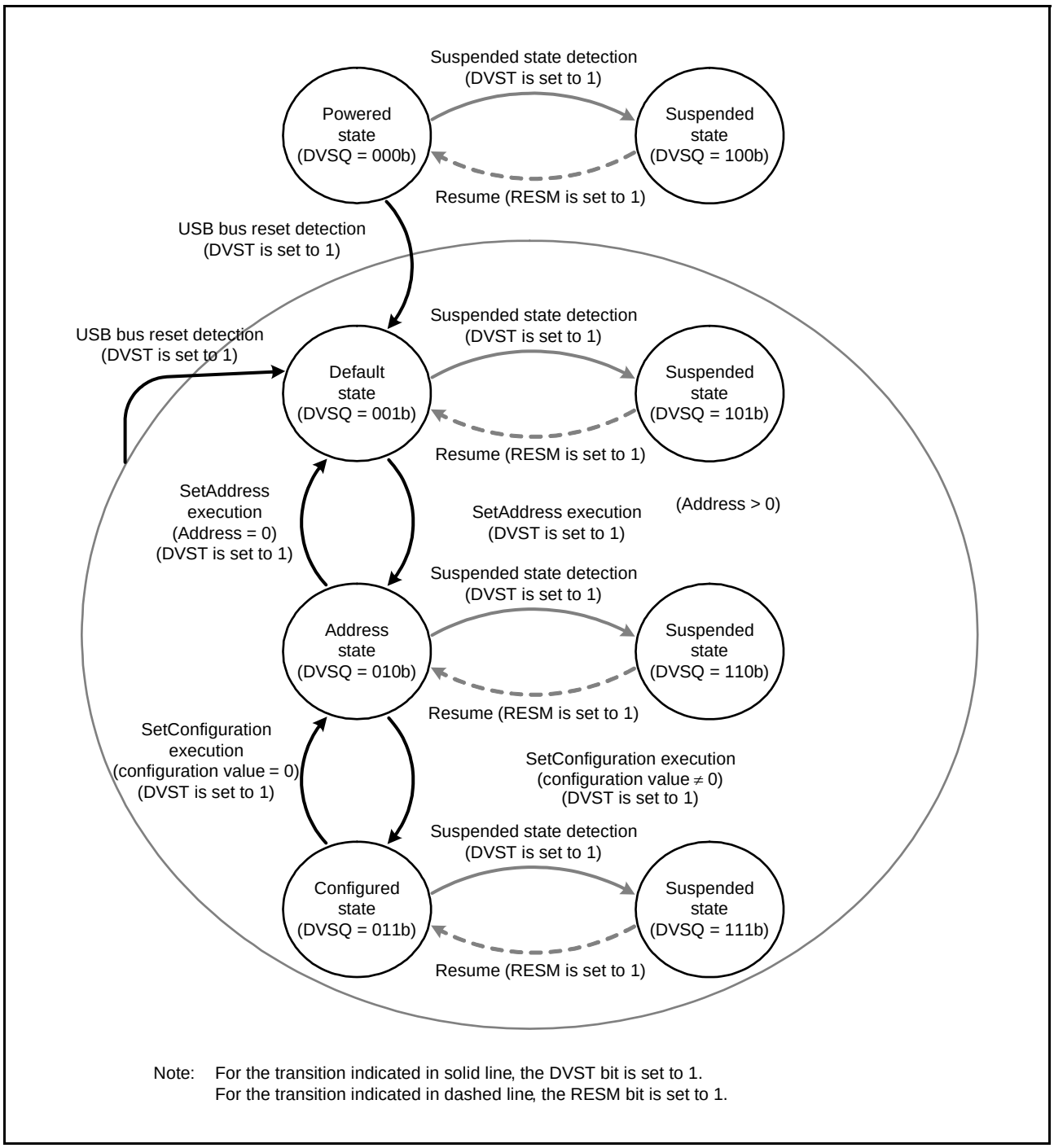


Figure 38.14 Device State Transitions

38.3.3.5 Control Transfer Stage Transition Interrupt

Figure 38.15 is a diagram of control transfer stage transitions in the USB. The USB controls the control transfer sequence and generates control transfer stage transition interrupts. The control transfer stage transition interrupts can be enabled or disabled individually using INTENB0. The transfer stage to which a transition was made can be confirmed using the INTSTS0.CTSQ[2:0] flags.

Control transfer stage transition interrupts are generated only when the function controller is selected.

The control transfer sequence errors are listed below. If an error occurs, the DCPCTR.PID[1:0] bits are set to 1xb (STALL response).

During control read transfer:

- An OUT token is received while no data has been transferred for the IN token at the data stage.
- An IN token is received at the status stage.
- A data packet with PID = DATA0 is received at the status stage.

During control write transfer:

- An IN token is received while no ACK response has been returned for the OUT token at the data stage.
- A data packet with PID = DATA0 is received for the first data packet at the data stage.
- An OUT token is received at the status stage

During no-data control transfers:

- An OUT token is received at the status stage.

At the control write transfer data stage, if the number of receive data exceeds the wLength value of the USB request, it cannot be recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (INTSTS0.CTRT = 1), CTSQ[2:0] = 110b value is retained until the CTRT flag = 0 is written from the system (the interrupt status is cleared). Therefore, while CTSQ[2:0] = 110b is being held, the CTRT interrupt that ends the setup stage will not be generated even if a new USB request is received. (The USB retains the setup stage end, and after the interrupt status has been cleared by software, a setup stage end interrupt is generated.)

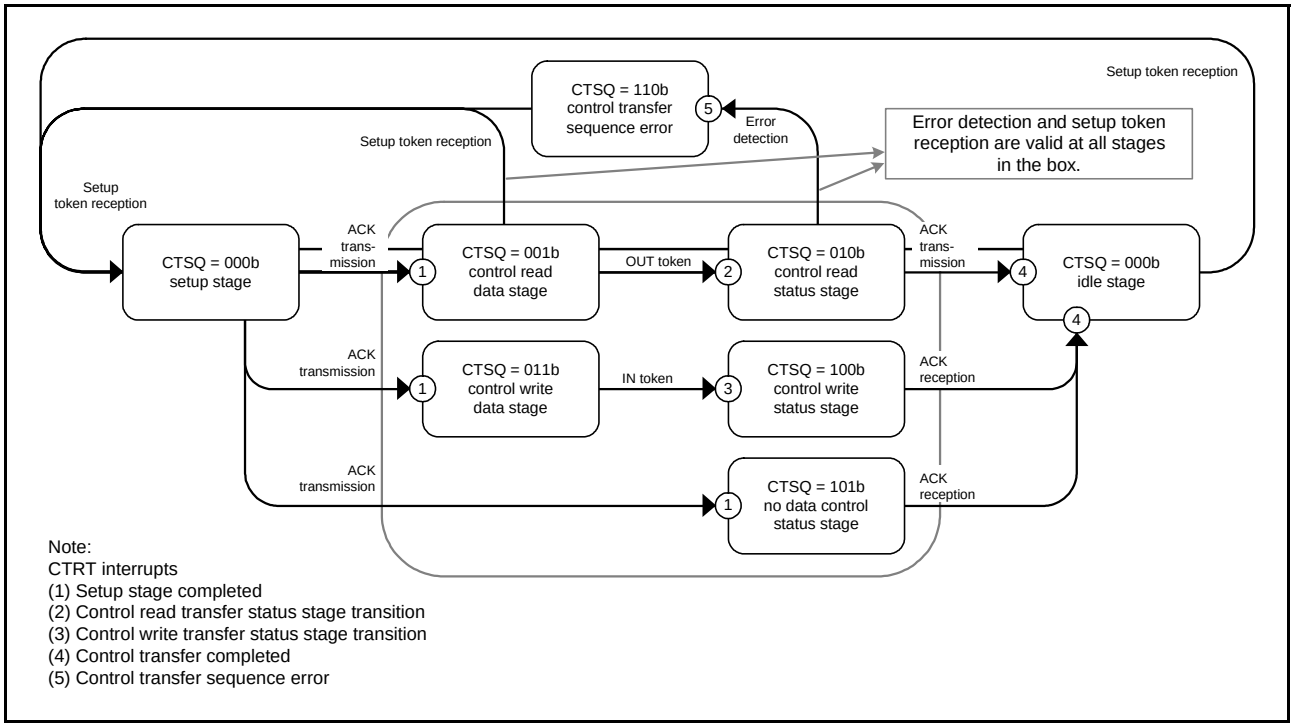


Figure 38.15 Control Transfer Stage Transitions

38.3.3.6 Frame Update Interrupt

With the host controller selected, an interrupt is generated at the timing when the frame number is updated. With the function controller selected, an SOFR interrupt is generated when the frame number is updated.
When the function controller is selected, the USB updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

38.3.3.7 VBUS Interrupt

When the USB0_VBUS pin level changes, a VBUS interrupt is generated. The level of the USB0_VBUS pin can be checked with the INTSTS0.VBSTS flag. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. However, if the system is activated with the host controller connected, the first VBUS interrupt is not generated because there is no change in the USB0_VBUS pin level.

38.3.3.8 Resume Interrupt

When the function controller is selected, a resume interrupt is generated when the device state is the suspended state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the suspended state is detected by means of the resume interrupt.
When the host controller is selected, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

38.3.3.9 OVRCR Interrupt

An OVRCR interrupt is generated when the USB0_OVRCURA or USB0_OVRCURB pin level has changed. The levels of the USB0_OVRCURA and USB0_OVRCURB pins can be checked with the SYSSTS0.OVCMON[1:0] bits. The external power supply IC can check whether overcurrent has been detected using the OVRCR interrupt.
For OTG connection, whether a change has been detected in the VBUS comparator can be checked using the OVRCR interrupt.

38.3.3.10 BCHG Interrupt

A BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether the peripheral device is connected and can also be used to detect a remote wakeup when the host controller is selected. The BCHG interrupt is generated regardless of whether the host controller or function controller is selected.

38.3.3.11 DTCH Interrupt

A DTCH interrupt is generated when disconnection of the USB bus is detected while the host controller is selected. The USB detects bus disconnection based on USB Specification 2.0.
After detecting a DTCH interrupt, the USB controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and make a transition to the wait state for bus connection to the pertinent port (wait state for ATTCH interrupt generation).

- Modifies the DVSTCTR0.UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the idle state.

38.3.3.12 SACK Interrupt

A SACK interrupt is generated when an ACK response for the transmitted setup packet has been received from the peripheral device with the host controller selected. The SACK interrupt can be used to confirm that the setup transaction has been completed successfully.

38.3.3.13 SIGN Interrupt

A SIGN interrupt is generated when an ACK response for the transmitted setup packet has not been correctly received from the peripheral device three consecutive times with the host controller selected. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

38.3.3.14 ATTCH Interrupt

An ATTCH interrupt is generated when J-state or K-state of the full-speed signal level is detected on the USB port for 2.5 µs with the host controller selected. To be more specific, an ATTCH interrupt is detected on any of the following conditions.

- When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 µs.
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 µs.

38.3.3.15 EOFERR Interrupt

An EOFERR interrupt is generated when it is detected that communication is not completed at the EOF2 timing prescribed in USB Specification 2.0.
After detecting an EOFERR interrupt, the USB controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and perform re-enumeration of the pertinent port.

- Modifies the DVSTCTR0.UACT bit for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

38.3.4 Pipe Control

Table 38.16 lists the pipe settings for the USB. With USB data transfer, data transfer is carried out using the pipe that the software has associated with the endpoint. The USB has ten pipes that are used for data transfer.
Appropriate settings should be made for each of the pipes according to the specifications of the system.

Table 38.16 Pipe Settings

Register Name	Bit Name	Setting	Remarks
DCPCFG PIPECFG	TYPE[1:0]	Specifies the transfer type	PIPE1 to PIPE9: Can be set
	BFRE	Selects the BRDY interrupt mode	PIPE1 to PIPE5: Can be set
	DBLB	Selects double buffer mode	PIPE1 to PIPE5: Can be set
	DIR	Selects transfer direction	IN or OUT can be set
	EPNUM[3:0]	Endpoint number	PIPE1 to PIPE9: Can be set A value other than 0000b should be set when the pipe is used.
	SHTNAK	Selects disabled state for pipe when transfer ends	PIPE1 and PIPE2: Can be set (only when bulk transfer has been selected) PIPE3 to PIPE5: Can be set
DCPMAXP PIPEMAXP	DEVSEL[3:0]	Selects a device	Referenced only when the host controller is selected.
	MXPS[8:0]	Maximum packet size	Compliant with USB Specification 2.0.
PIPEPERI	IFIS	Buffer flush	PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected) PIPE3 to PIPE9: Cannot be set
	IITV[2:0]	Interval counter	PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected) PIPE3 to PIPE5: Cannot be set PIPE6 to PIPE9: Can be set (only when the host controller has been selected)
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit.
	INBUFM	IN buffer monitor	Available only for PIPE1 to PIPE5.
	SUREQ	SETUP request	Can be set only for the DCP. Can be controlled only when the host controller has been selected.
	SUREQCLR	SUREQ clear	Can be set only for the DCP. Can be controlled only when the host controller has been selected.
	ATREPM	Auto response mode	PIPE1 to PIPE5: Can be set Can be set only when the function controller has been selected.
	ACLRM	Auto buffer clear	PIPE1 to PIPE9: Can be set
	SQCLR	Sequence clear	Clears the data toggle bit.
	SQSET	Sequence set	Sets the data toggle bit.
	SQMON	Sequence monitor	Monitors the data toggle bit.
	PBUSY	Pipe busy status	
	PID[1:0]	Response PID	Refer to section 38.3.4.6, Response PID.
PIPEnTRE	TRENB	Transaction counter enable	PIPE1 to PIPE5: Can be set
	TRCLR	Current transaction counter clear	PIPE1 to PIPE5: Can be set
PIPEnTRN	TRNCNT[15:0]	Transaction counter	PIPE1 to PIPE5: Can be set

38.3.4.1 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be modified only when USB communication is prohibited (PID[1:0] = 00b (NAK)).

The following shows the registers and bits that should not be modified when USB communication is enabled (PID[1:0] = 01b (BUF)).

- Bits in the DCPCFG and DCPMAXP registers
- The SQCLR and SQSET bits in the DCPCTR register
- Bits in registers PIPECFG, PIPEMAXP, and PIPEPERI
- The ATREPM, ACLRM, SQCLR, and SQSET bits in the PIPEnCTR register
- Bits in the PIPEnTRE and PIPEnTRN registers

In order to modify the above bits in the USB communication enabled (PID[1:0] = 01b (BUF)) state, follow the procedure shown below:

1. A request to modify bits in the pipe control register occurs.
2. Modify the PID[1:0] bits corresponding to the pipe to 00b (NAK).
3. Wait until the corresponding PBUSY flag is set to 0.
4. Modify the bits in the pipe control register.

The following bits in the pipe control registers can be modified only when the pertinent pipe information has not been set by the CURPIPE[3:0] bits in registers CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Registers that should not be set when the CURPIPE[3:0] bits are set:

- Bits in the DCPCFG and DCPMAXP register
- Bits in registers PIPECFG, PIPEMAXP and PIPEPERI

In order to modify pipe information, the CURPIPE[3:0] bits in the port select registers should be set to a pipe other than the pipe to be modified. For the DCP, the buffer should be cleared using the BCLR bit in the port control register after the pipe information is modified.

38.3.4.2 Transfer Types

The PIPECFG.TYPE[1:0] bits are used to specify the transfer type for each pipe. The transfer types that can be set for the pipes are as follows.

- DCP: No setting is necessary (fixed at control transfer).
- PIPE1 and PIPE2: These should be set to bulk transfer or isochronous transfer.
- PIPE3 to PIPE5: These should be set to bulk transfer.
- PIPE6 to PIPE9: These should be set to interrupt transfer.

38.3.4.3 Endpoint Number

The PIPECFG.EPNUM[3:0] bits are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to endpoint 15.

- DCP: No setting is necessary (fixed at endpoint 0).
- PIPE1 to PIPE9: The endpoint numbers from 1 to 15 should be selected and set.
  These should be set so that the combination of the PIPECFG.DIR bit and EPNUM[3:0] bits is unique.

38.3.4.4 Maximum Packet Size Setting

The DCPMAXP.MXPS[6:0] bits and the PIPEMAXP.MXPS[8:0] bits are used to specify the maximum packet size for each pipe. DCP and PIPE1 to PIPE5 can be set to any of the maximum pipe sizes defined by USB Specification 2.0. For PIPE6 to PIPE9, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID[1:0] = 01b (BUF)).

- DCP: Set 8, 16, 32, or 64.
- PIPE1 to PIPE5: Set 8, 16, 32, or 64 when using bulk transfer.
- PIPE1 and PIPE2: Set a value between 1 and 256 when using isochronous transfer.
- PIPE6 to PIPE9: Set a value between 1 and 64.

38.3.4.5 Transaction Counter (For PIPE1 to PIPE5 in Reading Direction)

When the specified number of transactions has been completed in the data packet receiving direction, the USB recognizes that the transfer has ended. Two transaction counters are provided: one is the PIPEnTRN register that specifies the number of transactions to be executed and the other is the current counter that internally counts the number of executed transactions. With the PIPECFG.SHTNAK bit set to 1, when the current counter value matches the specified number of transactions, the corresponding PIPEnCTR.PID[1:0] bits are set to 00b (NAK) and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the PIPEnTRE.TRCLR bit. The information read from PIPEnTRN differs depending on the setting of the PIPEnTRE.TRENB bit.

- The TRENB bit = 0: The specified transaction counter value can be read.
- The TRENB bit = 1: The current counter value indicating the internally counted number of executed transactions can be read.

When operating the TRCLR bit, the following should be noted.

- If the transactions are being counted and PID[1:0] = 01b (BUF), the current counter cannot be cleared.
- If there is any data left in the buffer, the current counter cannot be cleared.

38.3.4.6 Response PID

The PID[1:0] bits in the DCPCTR and PIPEnCTR registers are used to set the response PID for each pipe.
The following shows the USB operation with various response PID settings:

(1) Response PID settings when the host controller is selected:

The response PID is used to specify the execution of transactions.

- NAK setting: Using pipes is disabled. No transaction is executed.
- BUF setting: Transactions are executed based on the status of the buffer memory.
  For OUT direction: If there are transmit data in the buffer memory, an OUT token is issued.
  For IN direction: If there is an area to receive data in the buffer memory, an IN token is issued.
- STALL setting: Using pipes is disabled. No transaction is executed.

Note: Setup transactions for the DCP are set with the DCPCTR.SUREQ bit.

(2) Response PID settings when the function controller is selected:

The response PID is used to specify the response to transactions from the host.

- NAK setting: The NAK response is returned in response to the generated transaction.
- BUF setting: Responses are made to transactions according to the status of the buffer memory.
- STALL setting: The STALL response is returned in response to the generated transaction.

Note: For setup transactions, an ACK response is returned regardless of the PID[1:0] bits setting, and the USB request is stored in the register.

The USB may write to the PID[1:0] bits, depending on the results of the transaction as described below.

(3) When the host controller has been selected and the response PID is set by hardware:

- NAK setting: In the following cases, PID[1:0] = 00b (NAK) is set and issuing of tokens is automatically stopped:
  When a transfer other than isochronous transfer has been performed and an NRDY interrupt is generated.
  (For details, refer to section 38.3.3.2, NRDY Interrupt.)
  - If a short packet is received when the PIPECFG.SHTNAK bit has been set to 1 for bulk transfer.
  - If the transaction counting ends when the SHTNAK bit has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by the USB.
- STALL setting: In the following cases, PID[1:0] = 1xb (STALL) is set and issuing of tokens is automatically stopped:
  When STALL is received in response to the transmitted token.
  When the size of the receive data packet exceeds the maximum packet size.

(4) When the function controller has been selected and the response PID is set by hardware:

- NAK setting: In the following cases, PID[1:0] = 00b (NAK) is set and NAK is returned in response to transactions:
  When the SETUP token is received normally (DCP only).
  If the transaction counting ends or a short packet is received when the PIPECFG.SHTNAK bit has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by the USB.
- STALL setting: In the following cases, PID[1:0] = 1xb (STALL) is set and STALL is returned in response to transactions:
  When a maximum packet size exceeded error is detected in the received data packet.
  When a control transfer sequence error has been detected (DCP only).

38.3.4.7 Data PID Sequence Bit

The USB automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON flag in the DCPCTR and PIPEnCTR registers. When data is transmitted, the sequence bit switches at the timing of ACK handshake reception. When data is received, the sequence bit switches at the timing of ACK handshake transmission. The DCPCTR.SQCLR bit and the PIPEnCTR.SQSET bit can be used to change the data PID sequence bit.

When the function controller has been selected and control transfer is used, the USB automatically sets the sequence bit when a stage transition is made. DATA1 is returned when the setup stage is ended. The sequence bit is not referenced and PID = DATA1 is returned in a status stage. Therefore, software settings are not required. However, when the host controller has been selected and control transfer is used, the sequence bit should be set by software at a stage transition. For the ClearFeature request transmission or reception, the data PID sequence bit should be set by software regardless of whether the host controller or function controller is selected.

38.3.4.8 Response PID = NAK Function

The USB has a function that disables pipe operation (response PID = NAK) at the timing at which the final data packet of a transaction is received (the USB automatically distinguishes this based on reception of a short packet or the transaction counter) by setting the PIPECFG.SHTNAK bit to 1.

When the double buffer mode is being used for the buffer memory, using this function enables reception of data packets in transfer units. If pipe operation has been disabled, software should set the pipe to the enabled state again (response PID = BUF).

The response PID = NAK function can be used only when bulk transfers are used.

38.3.4.9 Auto Response Mode

With the pipes for bulk transfer (PIPE1 to PIPE5), when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode. During an OUT transfer (the PIPECFG.DIR bit = 0), OUT-NAK mode is entered, and during an IN transfer (the DIR bit = 1), null auto response mode is entered.

38.3.4.10 OUT-NAK Mode

With the pipes for bulk OUT transfer, NAK is returned in response to an OUT token and an NRDY interrupt is output when the PIPEnCTR.ATREPM bit is set to 1. To make a transition from normal mode to OUT-NAK mode, OUT-NAK mode should be specified in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, OUT-NAK mode becomes valid. However, if an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is returned to the host.

To make a transition from OUT-NAK mode to normal mode, OUT-NAK mode should be canceled in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). In normal mode, reception of OUT data is enabled.

38.3.4.11 Null Auto Response Mode

With the pipes for bulk IN transfer, zero-length packets are continuously transmitted when the PIPEnCTR.ATREPM bit is set to 1.

To make a transition from normal mode to null auto response mode, null auto response mode should be set in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, null auto response mode becomes valid. Before setting null auto response mode, the PIPEnCTR.INBUFM = 0 should be confirmed because the mode can be set only when the buffer is empty. If the INBUFM flag is 1, the buffer should be emptied with the PIPEnCTR.ACLRM bit. While a transition to null auto response mode is being made, data should not be written from the FIFO port.

To make a transition from null auto response mode to normal mode, pipe operation disabled state (response PID = NAK) should be retained for the period of zero-length packet transmission (about 10 µs) before canceling null auto response mode. In normal mode, data can be written from the FIFO port; therefore, packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

38.3.5 FIFO Buffer Memory

38.3.5.1 FIFO Buffer Memory

The USB has FIFO buffer memory for data transfer. The memory area used for each pipe is managed by the USB. The FIFO buffer memory has two states depending on whether the access right is assigned to the system (CPU side) or the USB (SIE side).

(1) Buffer Status

Table 38.17 and Table 38.18 show the buffer status in the USB. The buffer memory status can be confirmed using the DCPCTR.BSTS flag and the PIPEnCTR.INBUFM flag. The transfer direction for the buffer memory can be specified using either the PIPECFG.DIR bit or the CFIFOSEL.ISEL bit (when DCP is selected).

The INBUFM flag is valid for PIPE0 to PIPE5 in the transmitting direction.

When a transmitting pipe uses the double buffer configuration, software can read the BSTS flag to monitor the buffer memory status on the CPU side and the INBUFM flag to monitor the buffer memory status on the SIE side. When the BEMP interrupt may not show the buffer empty status because the write access to the FIFO port by the CPU or DMAC/DTC is slow, software can use the INBUFM flag to confirm the end of transmission.

Table 38.17 Buffer Status Indicated by the BSTS Flag

ISEL or DIR	BSTS	Buffer Memory Status
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is disabled.
0 (receiving direction)	1	There is received data, or a zero-length packet has been received. Reading from the FIFO port is allowed. Note that when a zero-length packet is received, reading is not possible and the buffer must be cleared.
1 (transmitting direction)	0	The transmission has not been completed. Writing to the FIFO port is disabled.
1 (transmitting direction)	1	The transmission has been completed. CPU write is allowed.

Table 38.18 Buffer Status Indicated by the INBUFM Flag

DIR	INBUFM	Buffer Memory Status
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	The transmission has been completed. There is no waiting data to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted.

38.3.5.2 FIFO Buffer Clearing

Table 38.19 shows the clearing of the FIFO buffer memory by the USB. The buffer memory can be cleared using the BCLR, DnFIFOSEL.DCLRM, and PIPEnCTR.ACLRM bit in the port control register.
Either a single or double buffer configuration can be selected for PIPE1 to PIPE5, using the PIPECFG.DBLB bit.

Table 38.19 List of Buffer Clearing Methods

FIFO Buffer Clearing Mode	Clearing Buffer Memory on CPU Side	Mode for Automatically Clearing Buffer Memory after Reading Specified Pipe Data	Auto Buffer Clear Mode for Discarding All Received Packets
Register used	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

(1) Auto Buffer Clear Mode Function

With the USB, all received data packets are discarded if the PIPEnCTR.ACLRM bit is set to 1. If a correct data packet has been received, the ACK response is returned to the host controller. The auto buffer clear mode function can be set only in the buffer memory reading direction.
If the ACLRM bit is set to 1 and then to 0, the buffer memory of the selected pipe can be cleared regardless of the access direction.
An access cycle of at least 100 ns is required for the internal hardware sequence processing time between ACLRM = 1 and ACLRM = 0.

38.3.5.3 FIFO Port Functions

Table 38.20 shows the settings for the FIFO port functions of the USB. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, the BVAL bit in the port control register should be set to end writing. To send a zero-length packet, the BCLR bit in the register should be used to clear the buffer and then the BVAL bit set in order to end writing.
In reading, reception of new packets is automatically enabled when all data has been read. Data cannot be read when a zero-length packet has been received (the DTLN[8:0] flags = 0), so the BCLR bit in the register should be used to clear the buffer. The length of the receive data can be confirmed using the DTLN[8:0] flags in the port control register.

Table 38.20 FIFO Port Function Settings

Register Name	Bit Name	Description
CFIFOSEL, DnFIFOSEL (n = 0, 1)	RCNT	Selects DTLN read mode.
	REW	Buffer memory rewind (re-read, rewrite).
	DCLRM	Automatically clears receive data for a specified pipe after the data has been read (only for DnFIFO).
	DREQE	Enables DMA/DTC transfers (only for DnFIFO).
	MBW	FIFO port access bit width.
	BIGEND	Selects FIFO port endian.
	ISEL	FIFO port access direction (only for DCP).
	CURPIPE	Selects the current pipe.
CFIFOCTR, DnFIFOCTR (n = 0, 1)	BVAL	Ends writing to the buffer memory.
	BCLR	Clears the buffer memory on the CPU side.
	DTLN	Checks the length of receive data.

(1) FIFO Port Selection

Table 38.21 shows the pipes that can be selected with the various FIFO ports. The pipe to be accessed should be selected using the CURPIPE[3:0] bits in the port select register. After the pipe is selected, whether the written value can be correctly read from the CURPIPE[3:0] bits should be checked. (If the previous pipe number is read, it indicates that the pipe modification is being executed by the USB controller.) Then, the FRDY flag in a port control register = 1 is checked.
In addition, the bus width to be accessed should be selected using the MBW bit in the port select register. The buffer memory access direction conforms to the PIPECFG.DIR bit. Only for the DCP, the ISEL bit in the port select register determines the direction.

Table 38.21 FIFO Port Access Categorized by Pipe

Pipe	Access Method	Port that can be Used
DCP	CPU access	CFIFO port register
PIPE1 to PIPE9	CPU access	CFIFO port register D0FIFO/D1FIFO port register
	DMAC/DTC access	D0FIFO/D1FIFO port register

(2) REW Bit

It is possible to temporarily stop access to the pipe currently being accessed, access a different pipe, and then continue processing for the current pipe again. The REW bit in the port select register is used for this processing.
If a pipe is selected through the CURPIPE[3:0] bits in the port select register with the REW bit set to 1, the pointer used for reading from and writing to the buffer memory is reset, and reading or writing can be carried out from the first byte.
If a pipe is selected with 0 set for the REW bit, data can be read and written in continuation from the previous selection, without the pointer being reset.
To access the FIFO port, the FRDY flag in the port control register = 1 should be checked after selecting a pipe.

38.3.5.4 DMA Transfers (D0FIFO and D1FIFO Ports)

(1) Overview of DMA Transfers

For PIPE1 to PIPE9, the FIFO port can be accessed using the DMAC. When accessing the buffer for the pipe targeted for DMA transfer is enabled, a DMA transfer request is issued.
The unit of transfer to the FIFO port should be selected using the DnFIFOSEL.MBW bit and the pipe targeted for the DMA transfer should be selected using the DnFIFOSEL.CURPIPE[3:0] bits. The selected pipe should not be changed during the DMA transfer.

(2) DnFIFO Auto Clear Mode (D0FIFO and D1FIFO Port Reading Direction)

If 1 is set in the DnFIFOSEL.DCLRM bit, the USB automatically clears the buffer memory of the selected pipe when reading of data from the buffer memory has been completed.
Table 38.22 shows the packet reception and buffer memory clearing processing by software for each of the various settings. As shown in Table 38.22, the buffer clearing conditions depend on the value set in the PIPECFG.BFRE bit. Using the DnFIFOSEL.DCLRM bit eliminates the need for the buffer to be cleared by software in any situation that requires buffer clearing. This enables DMA transfers without involving software.
The DnFIFO auto clear mode can be set only in the buffer memory reading direction.

Table 38.22 Packet Reception and Buffer Memory Clearing Processing by Software

Buffer Status When Packet is Received	Register Setting			
	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	Clearing is not necessary	Clearing is not necessary	Clearing is not necessary	Clearing is not necessary
Zero-length packet reception	Clearing is necessary	Clearing is necessary	Clearing is not necessary	Clearing is not necessary
Normal short packet reception	Clearing is not necessary	Clearing is necessary	Clearing is not necessary	Clearing is not necessary
Transaction count end	Clearing is not necessary	Clearing is necessary	Clearing is not necessary	Clearing is not necessary

38.3.6 Control Transfers Using DCP

In the data stage of control transfers, data is transferred using the default control pipe (DCP).
The DCP buffer memory is a 64-byte single buffer and is a fixed area that is shared for both control reading and control writing. The buffer memory can be accessed only through the CFIFO port.

38.3.6.1 Control Transfers When the Host Controller is Selected

(1) Setup Stage

Registers USBREQ, USBVAL, USBINDX, and USBLENG are the registers that are used to transmit a USB request for setup transactions. Writing setup packet data to the registers and writing 1 to the DCPCTR.SUREQ bit transmits the specified data for setup transactions. Upon completion of the transaction, the SUREQ bit is set to 0. The above USB request registers should not be modified while SUREQ = 1.
After the attached state of the connected function device is detected, the first setup transaction for the device should be issued by using the sequence described above with the DCPMAXP.DEVSEL[3:0] bits set to 0 and the DEVADD0.USBSPD[1:0] bits set appropriately.
After the connected function device is shifted to the Address state, setup transactions should be issued by using the sequence described above with the assigned USB address set in the DEVSEL[3:0] bits and the bits in the DEVADDn register corresponding to the specified USB address set appropriately. For example, when PIPEMAXP.DEVSEL[3:0] = 0010b, make appropriate settings in the DEVADD2 register; when PIPEMAXP.DEVSEL[3:0] = 0101b, make appropriate settings in the DEVADD5 register.
When the setup transaction data has been sent, an interrupt request is generated according to the response received from the peripheral device (SIGN or SACK flag in the INTSTS1 register), by means of which the result of the setup transactions can be confirmed.
A data packet of DATA0 (USB request) is transmitted as the data packet for a setup transaction regardless of the setting of the DCPCTR.SQMON flag.

(2) Data Stage

Data is transferred using the DCP buffer memory.
The access direction of the DCP buffer memory should be specified using the CFIFOSEL.ISEL bit. The transfer direction should be specified using the DCPCFG.DIR bit.
For the first data packet of the data stage, the data PID should be transferred as DATA1. Set data PID = DATA1 in the DCPCTR.SQSET bit and the PID[1:0] bits = 01b (BUF). Completion of data transfer is detected using the BRDY or BEMP interrupt.
For control write transfers, when the number of data bytes to be sent is an integer multiple of the maximum packet size, software should control so as to send a zero-length packet at the end.

(3) Status Stage

Zero-length packet data is transferred in the direction opposite to that in the data stage. As in the data stage, data is transferred using the DCP buffer memory. Transactions are done in the same manner as the data stage.
For the data packets of the status stage, the data PID should be set to DATA1 using the DCPCTR.SQSET bit.
For reception of a zero-length packet, the received data length should be confirmed using the CFIFOCTR.DTLN[8:0] flags after a BRDY interrupt is generated, and the buffer memory should then be cleared using the CFIFOCTR.BCLR bit.

38.3.6.2 Control Transfers When the Function Controller is Selected

(1) Setup Stage

The USB sends an ACK response for a correct setup packet targeted to the USB. The operation of the USB in the setup stage is described below.
When receiving a new setup packet, the USB sets the following bits.

- Set the INTSTS0.VALID flag to 1.
- Set the DCPCTR.PID[1:0] bits to 00b (NAK).
- Set the DCPCTR.CCPL bit to 0.

When receiving a data packet right after the setup packet, the USB stores the USB request parameters in registers USBREQ, USBVAL, USBINDX, and USBLENG.

Response processing with respect to the control transfer should be carried out after setting the VALID flag = 0. In the VALID flag = 1 state, PID[1:0] = 01b (BUF) cannot be set, and the data stage cannot be terminated.
Using the function of the VALID flag, the USB can suspend the current request processing when receiving a new USB request during a control transfer, and can send a response to the newest request.
In addition, the USB automatically detects the direction bit (bit 8 of bmRequestType) and the request data length (wLength) of the received USB request, distinguishes between control read transfer, control write transfer, and no-data control transfer, and controls stage transitions. For a wrong sequence, the sequence error of the control transfer stage transition interrupt is generated, and the software is notified of occurrence of the error. For the stage control of the USB, refer to Figure 38.15.

(2) Data Stage

Data transfers corresponding to received USB requests should be done using the DCP. Before accessing the DCP buffer memory, the access direction should be specified using the CFIFOSEL.ISEL bit.
If the transfer data is larger than the size of the DCP buffer memory, the data transfer should be carried out using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

(3) Status Stage

Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to 01b (BUF).
After the above settings have been made, the USB automatically executes the status stage in accordance with the data transfer direction determined at the setup stage. The specific procedure is as follows.

- For control read transfers
  A zero-length packet is received from the USB host and an ACK response is sent.
- For control write transfers and no-data control transfers
  A zero-length packet is transmitted and an ACK response is received from the USB host.

(4) Control Transfer Auto Response Function

The USB automatically responds to a correct SET_ADDRESS request. If any of the following errors occurs in the SET_ADDRESS request, a response from the software is necessary.

- bmRequestType is not 00h: Any transfer other than a control write transfer
- wIndex is not 00h: Request error
- wIndex is not 00h: Any transfer other than a no-data control transfer
- wValue is larger than 7Fh: Request error
- INTSTS0.DVSQ[2:0] are 011b (Configured state): Control transfer of a device state error

For all requests other than the SET_ADDRESS request, a response is required from the corresponding software.

38.3.7 Bulk Transfers (PIPE1 to PIPE5)

The buffer memory usage (single/double buffer setting) can be selected for bulk transfers. The USB provides the following functions for bulk transfers.

- BRDY interrupt function (PIPECFG.BFRE bit: refer to section 38.3.3.1, (2) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 1
- Transaction count function
  (PIPEnTRE.TRENB, TRCLR, and PIPEnTRN.TRNCNT[15:0] bits: refer to section 38.3.4.5, Transaction Counter (For PIPE1 to PIPE5 in Reading Direction)
- Response PID = NAK function (PIPECFG.SHTNAK bit: refer to section 38.3.4.8, Response PID = NAK Function
- Auto response mode (PIPEnCTR.ATREPM bit: refer to section 38.3.4.9, Auto Response Mode)

38.3.8 Interrupt Transfers (PIPE6 to PIPE9)

When the function controller is selected, the USB carries out interrupt transfers in accordance with the timing controlled by the host controller.
When the host controller is selected, the timing of issuing a token can be specified using the interval counter.

38.3.8.1 Interval Counter during Interrupt Transfers When the Host Controller is Selected

For interrupt transfers, intervals between transactions are set in the PIPEPERI.IITV[2:0] bits. The USB controller issues interrupt transfer tokens based on the specified intervals.

(1) Counter Initialization

The interval counter is initialized when the MCU is reset or when the PIPEnCTR.ACLRM bit is set to 1. Note that the PIPEPERI.IITV[2:0] bits are not initialized when the ACLRM bit is used for initialization.

Note that the interval counter is not initialized in the following case.

- USB bus reset or USB suspended
  The IITV[2:0] bits are not initialized. Setting 1 to the DVSTCTR0.UACT bit starts counting from the value before entering the USB bus reset state or USB suspended state.

(2) Operation When Transmission/Reception is Impossible at Token Issuance Timing

The USB cannot issue tokens even at token issuance timing in the following cases. In such a case, the USB attempts transactions at the subsequent interval.

- When the PID[1:0] bits are set to 00b (NAK) or 1xb (STALL).
- When the buffer memory is full at the token sending timing in the receiving (IN) direction.
- When there is no data to be sent in the buffer memory at the token sending timing in the transmitting (OUT) direction.

38.3.9 Isochronous Transfers (PIPE1 and PIPE2)

The USB has the following functions for isochronous transfers.

- Notification of isochronous transfer error information
- Interval counter (specified by the PIPEPERI.IITV[2:0] bits)
- Isochronous IN transfer data setup control (IDLY function)
- Isochronous IN transfer buffer flush function (specified by the PIPEPERI.IFIS bit)

38.3.9.1 Error Detection in Isochronous Transfers

The USB has a function for detecting the error information described below, so that when errors occur in isochronous transfers, they can be controlled by software. Table 38.23 and Table 38.24 show the priority in which errors are confirmed and the interrupts generated corresponding to errors.

(a) PID errors

- If the PID of the received packet is illegal.

(b) CRC errors and bit stuffing errors

- If an error occurs in the CRC of the received packet or the bit stuffing is illegal.

(c) Maximum packet size exceeded

- The data of the received packet is larger than the specified maximum packet size.

(d) Overrun and underrun errors

- When the host controller is selected
  When the buffer memory is full at the token sending timing in the IN (receiving) direction.
  When there is no data to be sent in the buffer memory at the token sending timing in the OUT (transmitting) direction.
- When the function controller is selected
  When there is no data to be sent in the buffer memory at the token receiving timing in the IN (transmitting) direction.
  When the buffer memory is full at the token receiving timing in the OUT (receiving) direction.

(e) Interval errors

An interval error is generated on any of the following conditions when the function controller is selected.

- During an isochronous IN transfer, an IN token could not be received in the interval frame.
- During an isochronous OUT transfer, an OUT token could not be received in the interval frame.

Table 38.23 Error Detection When a Token is Received

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated in both cases when the host controller is selected and the function controller is selected (ignored as a corrupted packet).
2	CRC errors and bit stuffing errors	No interrupts generated in both cases when the host controller is selected and the function controller is selected (ignored as a corrupted packet).
3	Overrun and underrun errors	An NRDY interrupt is generated to set the FRMNUM.OVRN flag to 1 in both cases when the host controller is selected and function controller is selected. When the function controller is selected, a zero-length packet is transmitted in response to IN token. However, no data packets are received in response to OUT token.
4	Interval errors	An NRDY interrupt is generated when the function controller is selected. It is not generated when the host controller is selected.

Table 38.24 Error Detection When a Data Packet is Received

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated (ignored as a corrupted packet).
2	CRC errors and bit stuffing errors	An NRDY interrupt is generated to set the FRMNUM.CRCE to 1 bit in both cases when the host controller is selected and the function controller is selected.
3	Maximum packet size exceeded errors	A BEMP interrupt is generated to set the PID[1:0] bits to 1xb (STALL) in both cases when the host controller is selected and the function controller is selected.

38.3.9.2 Data PID

When the function controller is selected, the USB operates as follows in response to the received PID.
IN direction

- DATA0: Sent as data packet PID
- DATA1: Not sent
- DATA2: Not sent
- mDATA: Not sent

OUT direction

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets are ignored
- mDATA: Packets are ignored

38.3.9.3 Interval Counter

The isochronous transfer interval can be set using the PIPEPERI.IITV[2:0] bits. The interval counter enables the functions shown in Table 38.25 when the function controller is selected. When the host controller is selected, the USB generates the token issuance timing. When the host controller is selected, the interval counter operation is the same as that in the interrupt transfer.

Table 38.25 Interval Counter Function When the Function Controller is Selected

Transfer Direction	Function	Conditions for Detection
IN	Flushes transmit buffer	When an IN token cannot be successfully received in the interval frame during an isochronous IN transfer
OUT	Notifies that a token not being received	When an OUT token cannot be successfully received in the interval frame during an isochronous OUT transfer

The interval count is carried out when an SOF is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF is damaged. The frame interval that can be set is the $2^{IITV[2:0]}$ frames.

(1) Counter Initialization When the Function Controller is Selected

The interval counter is initialized when the MCU is reset or when the PIPEnCTR.ACLRM bit is set to 1. Note that the PIPEPERI.IITV[2:0] bits are not initialized when the ACLRM bit is used for initialization.

After the interval counter has been initialized, counting is started under either of the following conditions 1 and 2 when a packet has been transferred successfully.

1. An SOF is received after transmission of data in response to an IN token while the PID[1:0] bits are 01b (BUF).
2. An SOF is received after reception of data of an OUT token while the PID[1:0] bits are 01b (BUF).

Note that the interval counter is not initialized under the following conditions.

- When the PID[1:0] bits are set to 00b (NAK) or 1xb (STALL)
  The interval timer does not stop. The USB attempts transactions at the subsequent interval.
- When the USB bus is reset or USB is suspended
  The IITV[2:0] bits are not initialized. When an SOF has been received, counting is restarted from the value prior to the reception of the SOF.

(2) Interval Counting and Transfer Control When the Host Controller is Selected

The USB controls the interval between token issuance operations based on the PIPEPERI.IITV[2:0] bit settings. Specifically, the USB issues a token for a selected pipe once every $2^{IITV[2:0]}$ frames.
The USB starts counting the token issuance interval at the frame following the frame in which the PID[1:0] bits have been set to 01b (BUF) by software.

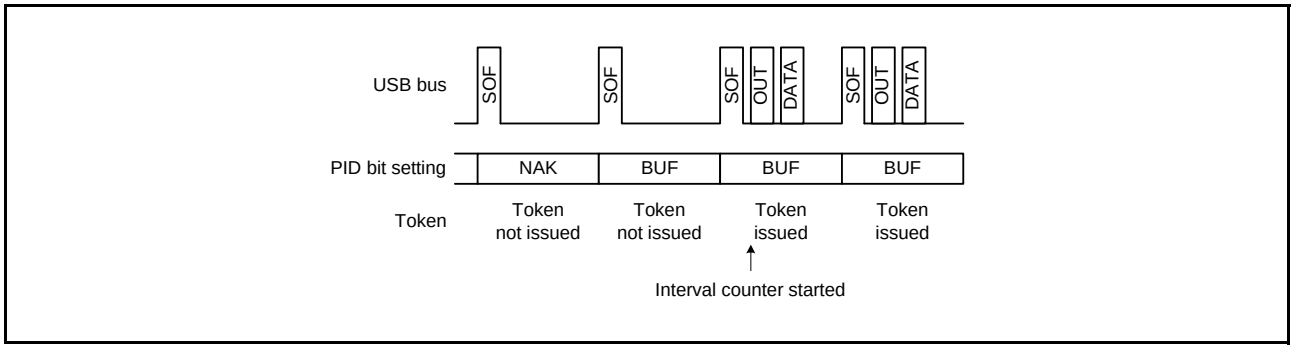


Figure 38.16 Token Issuance When IITV[2:0] = 000b

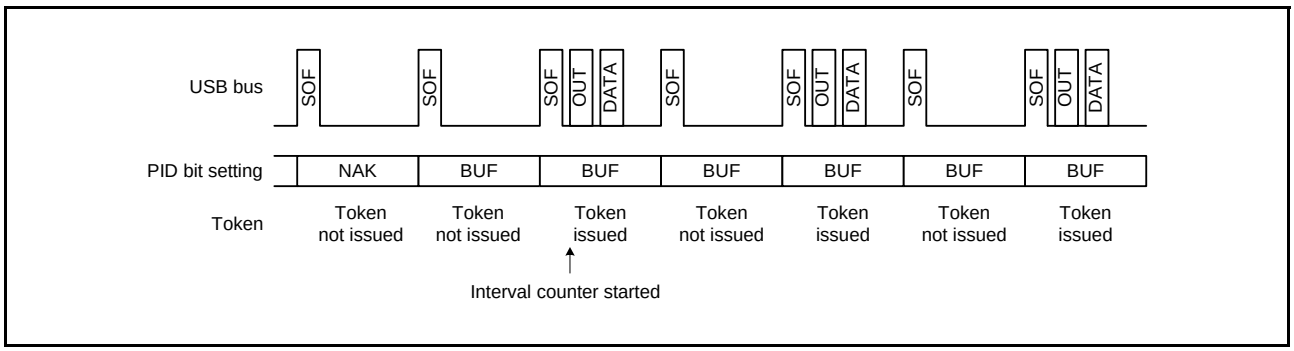


Figure 38.17 Token Issuance When IITV[2:0] = 001b

When the selected pipe is set for isochronous transfers, the USB carries out the following operation in addition to controlling the token issuance interval. The USB issues a token even when the NRDY interrupt generation condition is satisfied.

(a) When the selected pipe is for isochronous IN transfers

The USB generates an NRDY interrupt when the USB issues an IN token but does not receive a packet successfully from a peripheral device (no response or packet error).
The USB sets the FRMNUM.OVRN flag to 1 generating an NRDY interrupt when the time to issue an IN token comes while the USB cannot receive data because the FIFO buffer is full (due to the fact that the CPU or DMAC/DTC is too slow to read data from the FIFO buffer).

(b) When the selected pipe is for isochronous OUT transfers

The USB sets the OVRN flag to 1 generating an NRDY interrupt and transmitting a zero-length packet when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer (because the CPU or DMAC/DTC is too slow to write data to the FIFO buffer).
The token issuance interval is reset on any of the following conditions.

- When the USB is reset through a reset pin
  (The IITV[2:0] bits are also set to 0).
- When the PIPEnCTR.ACLRM bit has been set to 1 by software.

(3) Interval Counting and Transfer Control When the Function Controller is Selected

(a) When the selected pipe is for isochronous OUT transfers

The USB generates an NRDY interrupt when the USB fails to receive a data packet within the interval set by the PIPEPERI.IITV[2:0] bits.
The USB also generates an NRDY interrupt when the USB fails to receive data because of a CRC error or other errors contained in the data packet or because of the FIFO buffer being full.
The NRDY interrupt is generated at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the interrupt to be generated at the timing to receive the SOF packet.
However, when the IITV[2:0] bits are set to a value other than 0, the USB generates an NRDY interrupt on receiving an SOF packet for every interval after starting interval counting operation.
When the PID[1:0] bits are set to 00b (NAK) by software after starting the interval timer, the USB does not generate an NRDY interrupt on receiving an SOF packet.
The timing to start interval counting depends on the setting of IITV[2:0] bits as shown below.

- When the IITV[2:0] = 000b: The interval counting starts when software has set the PID[1:0] bits for the selected pipe to 01b (BUF).

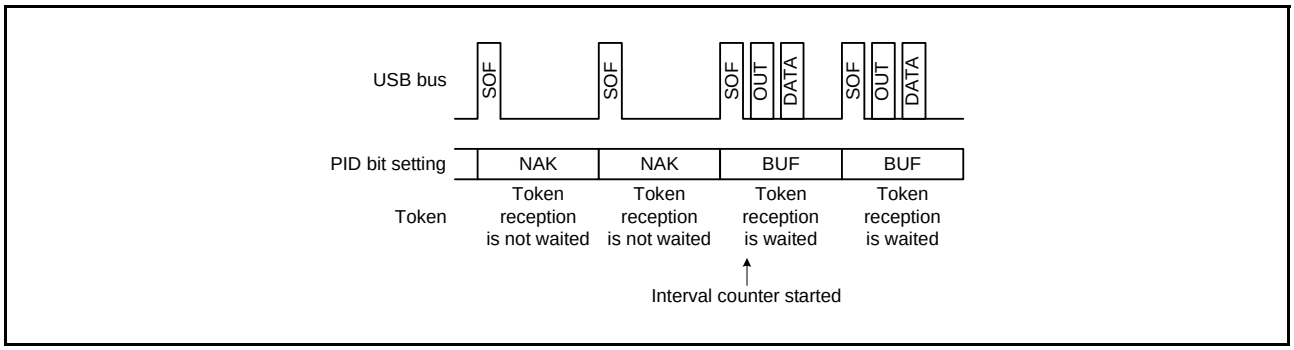


Figure 38.18 Relationship between Frames and Expected Token Reception When IITV[2:0] = 000b

- When the IITV[2:0] ≠ 000b: The interval counting starts on completion of successful reception of the first data packet after the PID[1:0] bits for the selected pipe have been modified to 01b (BUF).

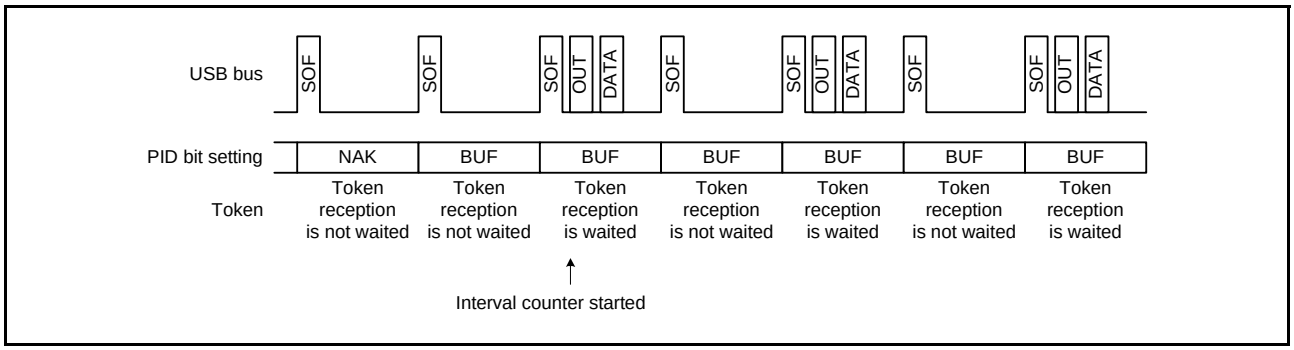


Figure 38.19 Relationship between Frames and Expected Token Reception When IITV[2:0] ≠ 000b

(b) When the selected pipe is for isochronous IN transfers

The PIPEPERI.IFIS bit should be 1 for this use. When IFIS = 0, the USB transmits a data packet in response to the received IN token irrespective of the setting of the PIPEPERI.IITV[2:0] bits.
When IFIS = 1, the USB clears the FIFO buffer when the USB fails to receive an IN token in the frame at the interval set by the IITV[2:0] bits while there is data to be transmitted in the FIFO buffer.
The USB also clears the FIFO buffer when the USB fails to receive an IN token successfully because of a bus error such as a CRC error contained in the IN token.
The FIFO buffer is cleared at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the FIFO buffer to be cleared at the timing to receive the SOF packet.

The timing to start interval counting depends on the setting of the IITV[2:0] bits (similar to the timing during OUT transfers).
The interval is counted on any of the following conditions in function controller mode.

- When a hardware-reset is applied to the USB (here, the IITV[2:0] bits are also set to 000b).
- When the PIPEnCTR.ACLRM bit is set to 1 by software.
- When the USB detects a USB bus reset.

(4) Setup of Data to be Transmitted Using Isochronous Transfer When the Function Controller is Selected

With isochronous data transmission using the USB in the function controller, after data has been written to the buffer memory, a data packet can be transmitted with the next frame after the frame in which an SOF packet is detected. This function is called the isochronous transfer transmission data setup function, and it makes it possible to designate the frame from which transmission began.

In a double buffer configuration, even after the writing of data to both buffers has been completed, transmission will be enabled for only one buffer to which data writing was completed first. Accordingly, even if multiple IN tokens are received, only one packet of data is transmitted from a single buffer.

When an IN token is received, if the buffer memory is in the transmission enabled state, the USB transmits data as a normal response. If the buffer memory is not in the transmission enabled state, however, a zero-length packet is sent and an underrun error occurs.

Figure 38.20 shows an example of transmission using the isochronous transfer transmission data setup function with the USB when IITV[2:0] = 000b (every frame) has been set.

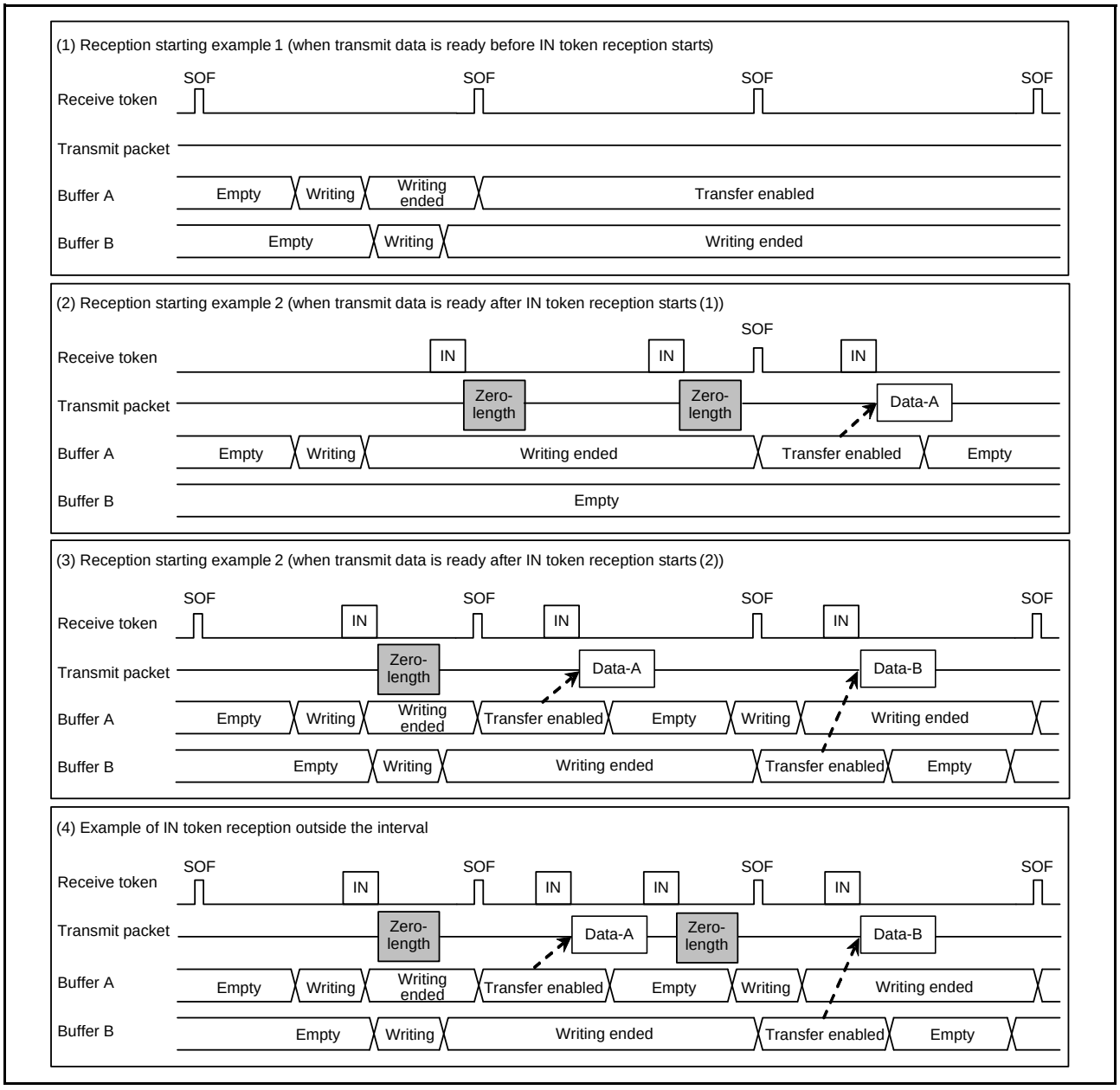


Figure 38.20 Example of Data Setup Function Operation

(5) Isochronous Transfer Transmission Buffer Flush When the Function Controller is Selected

If an SOF packet of the next frame is received without receiving an IN token in an interval frame during isochronous data transmission, the USB operates as if an IN token had been corrupted, and clears the buffer for which transmission is enabled, putting that buffer in the writing enabled state.
If a double buffer configuration is used and writing to both buffers has been completed, the buffer memory that was cleared is assumed as the data having been sent in the interval frame, and transmission is enabled for the buffer memory that is not cleared with SOF packet reception.
The timing of the buffer flush function depends on the setting of the PIPEPERI.IITV[2:0] bits.

- When the IITV[2:0] = 000b
  The buffer flush operation starts from the next frame after the pipe becomes valid.
- When the IITV[2:0] ≠ 000b
  The buffer flush operation is carried out after the first successful transaction.

Figure 38.21 shows an example of the buffer flush function in the USB. When an unanticipated token is received before the interval frame, the USB sends the write data or a zero-length packet as an underrun error according to the data setup state.

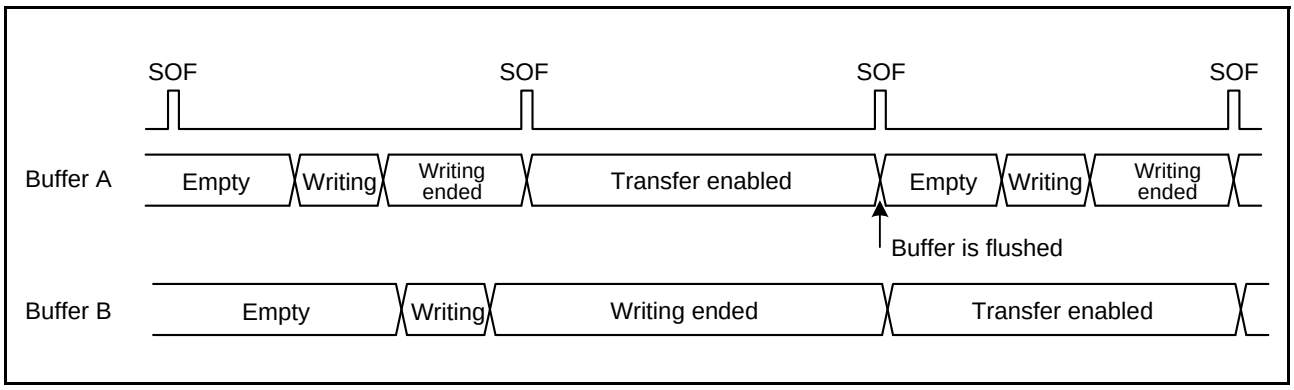


Figure 38.21 Example of Buffer Flush Operation

Figure 38.22 shows an example of interval error occurrence in the USB. There are five types of interval errors, as shown below. The interval error is generated at the timing indicated by ① in the figure, and the buffer flush function is activated.

If an interval error occurs during an IN transfers, the buffer flush function is activated; if it occurs during an OUT transfer, an NRDY interrupt is generated.

The FRMNUM.OVRN flag should be used to distinguish between NRDY interrupts such as received packet errors and overrun errors.

In response to tokens that are shaded in the figure, responses are sent according to the buffer memory status.

IN direction

- If the buffer is in the transmission enabled state, the data is transferred as a normal response.
- If the buffer is in the transmission disabled state, a zero-length packet is sent and an underrun error occurs.

OUT direction

- If the buffer is in the reception enabled state, the data is received as a normal response.
- If the buffer is in the reception disabled state, the data is discarded and an overrun error occurs.

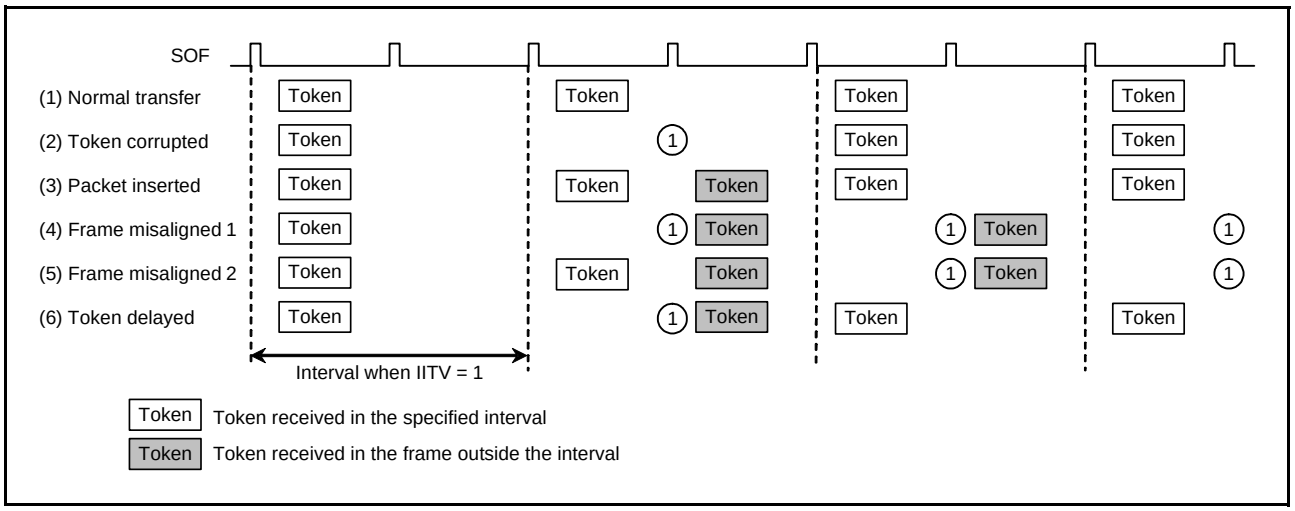


Figure 38.22 Example of Interval Error Occurrence When IITV[2:0] = 001b

38.3.10 SOF Interpolation Function

When the function controller is selected and if data could not be received at intervals of 1 ms because an SOF packet was corrupted or missing, the USB interpolates the SOF. The SOF interpolation operation begins when the USBE and SCKE bits in the SYSCFG register have been set to 1 and an SOF packet is received. The interpolation function is initialized under the following conditions.

- MCU reset
- USB bus reset
- Suspended state detected

The SOF interpolation operates as follows.

- The interpolation function is not activated until an SOF packet is received.
- After the first SOF packet is received, interpolation is carried out by counting 1 ms with an internal clock of 48 MHz.
- After the second and subsequent SOF packets are received, interpolation is carried out at the previous reception interval.
- Interpolation is not carried out in the suspended state or while a USB bus reset is being received.

The USB supports the following functions based on the SOF packet reception. These functions also operate normally with SOF interpolation, if the SOF packet was missing.

- Updating of the frame number
- SOFR interrupt timing
- Isochronous transfer interval count

If an SOF packet is missing during full-speed operation, the FRMNUM.FRNM[10:0] flags are not updated.

38.3.11 Pipe Schedule

38.3.11.1 Conditions for Generating a Transaction

When the host controller is selected and the DVSTCTR0.UACT bit has been set to 1, the USB generates a transaction under the conditions shown in Table 38.26.

Table 38.26 Conditions for Generating a Transaction

	Conditions for Generation				
Transaction	**DIR**	**PID[1:0]**	**IITV[0]**	**Buffer State**	**SUREQ**
Setup	—*1	—*1	—*1	—*1	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	Invalid	Receive area exists	—*1
	OUT	BUF	Invalid	Transmit data exists	—*1
Interrupt transfer	IN	BUF	Valid	Receive area exists	—*1
	OUT	BUF	Valid	Transmit data exists	—*1
Isochronous transfer	IN	BUF	Valid	*2	—*1
	OUT	BUF	Valid	*3	—*1

Note 1. Symbols (—) in the table indicate that the condition is unrelated to the generating of tokens. "Valid" indicates that, for interrupt transfers and isochronous transfers, a transaction is generated only in transfer frames that are based on the interval counter. "Invalid" indicates that a transaction is generated regardless of the interval counter.
Note 2. This indicates that a transaction is generated regardless of whether there is a receive area. If there is no receive area, however, the received data is discarded.
Note 3. This indicates that a transaction is generated regardless of whether there is any data to be transmitted. If there is no data to be transmitted, however, a zero-length packet is transmitted.

38.3.11.2 Transfer Schedule

This section describes the transfer scheduling within a frame of the USB. After the USB sends an SOF, the transfer is carried out in the sequence described below.

1. Execution of periodic transfers
   A pipe is searched in the order of PIPE1 → PIPE2 → PIPE6 → PIPE7 → PIPE8 → PIPE9, and then, if there is a pipe for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.
2. Setup transactions for control transfers
   The DCP is checked, and if a setup transaction is possible, it is sent.
3. Execution of bulk transfers, control transfer data stages, and control transfer status stages
   A pipe is searched in the order of DCP → PIPE1 → PIPE2 → PIPE3 → PIPE4 → PIPE5, and then, if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.
   When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

38.3.11.3 Enabling USB Communication

Setting the DVSTCTR0.UACT bit to 1 initiates SOF transmission and transaction generation is enabled.
Setting the UACT bit to 0 stops SOF transmission and a suspend state is entered. If the setting of the UACT bit is changed from 1 to 0, processing stops after the next SOF is sent.

38.4 Usage Notes

38.4.1 Setting the Module Stop Function

Operation of the USB module can be disabled or enabled using module stop control register B (MSTPCRB). The setting after a reset is for operation of the USB module to be stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

39. USB2.0 Full-Speed Host/Function Module (USBA)

39.1 Overview

This MCU incorporates a USB2.0 full-speed host/function module (USBA). The USBA is a USB controller that is equipped to operate as a host controller or a function controller. As a host controller, the USBA supports full-speed transfer and low-speed transfer as defined in the Universal Serial Bus Specification Revision 2.0 (hereinafter referred to as USB 2.0). As a function controller, the USBA supports full-speed transfer as defined in USB 2.0. The USBA has an internal USB transceiver and supports all of the transfer types defined in USB 2.0. The USBA has FIFO buffers for data transfer, providing a maximum of 10 pipes. Any endpoint numbers can be assigned to pipes 1 to 9, based on the peripheral devices or user system for communication.

This chapter includes the following initialisms:

- BESL: Best Effort Service Latency
- HIRD: Host Initiated Resume Duration
- UTMI: USB Transceiver Macrocell Interface
- PHY: USB physical layer transceiver

Table 39.1 USBA Specifications

Item	Specifications
Features	• Incorporates a USB device controller (UDC) and transceiver for USB 2.0 The host controller and function controller operations are provided and the On-The-Go (OTG) function is supported. • Host controller and function controller operations are switched by software.
	In host controller operation: • Supports full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) • Automatic scheduling for Start-of-Frame (SOF) packets and other packet transmissions • Programmable intervals for isochronous and interrupt transfers • Communications with multiple peripheral devices connected via a single hub
	In function controller operation: • Supports full-speed transfer (12 Mbps) • Control transfer stage control function • Device state control function • Auto response function for SetAddress() request • SOF recovery function
Communication data transfer types	• Control transfer • Bulk transfer • Interrupt transfer • Isochronous transfer
Internal bus interface	• Connected to internal peripheral bus 3
Pipe configuration	• FIFO buffer of up to 8.5 Kbytes for USB communications is available. • Up to 10 pipes can be selected (including the default control pipe). • Programmable pipe configurations • Endpoint numbers can be assigned flexibly to pipes 1 to 9.
	• Transfer conditions that can be set for each pipe: Pipe 0 (default control pipe): Control transfer, 64-byte fixed single buffer Pipes 1 and 2: Bulk transfer/isochronous transfer in continuous transfer mode. Programmable buffer size up to 2 Kbytes. Selectable single buffer or double buffer mode. Pipes 3 to 5: Bulk transfer in continuous transfer mode. Programmable buffer size up to 2 Kbytes. Selectable single buffer or double buffer mode. Pipes 6 to 9: Interrupt transfer, 64-byte fixed single buffers
Other features	• Transfer ending function using transaction count • Function that changes the BRDY interrupt event notification timing • Function that automatically clears the FIFO buffer after the data for the pipe specified at the D0FIFO port or D1FIFO port has been read • NAK setting function for response packet ID (PID) generated by end of transfer • Internal pull-up and pull-down resistors for D+ and D– • Supports the USB 2.0 Engineering Change Notice for Link Power Management (LPM). A new sleep state (referred to as the L1 state) is available. • Supports Battery Charging Specification Revision 1.2 • To reduce power consumption, a classic-only mode where operation is only in accord with the USB 1.1 standard is selectable.

Figure 39.1 shows a block diagram of the USBA.

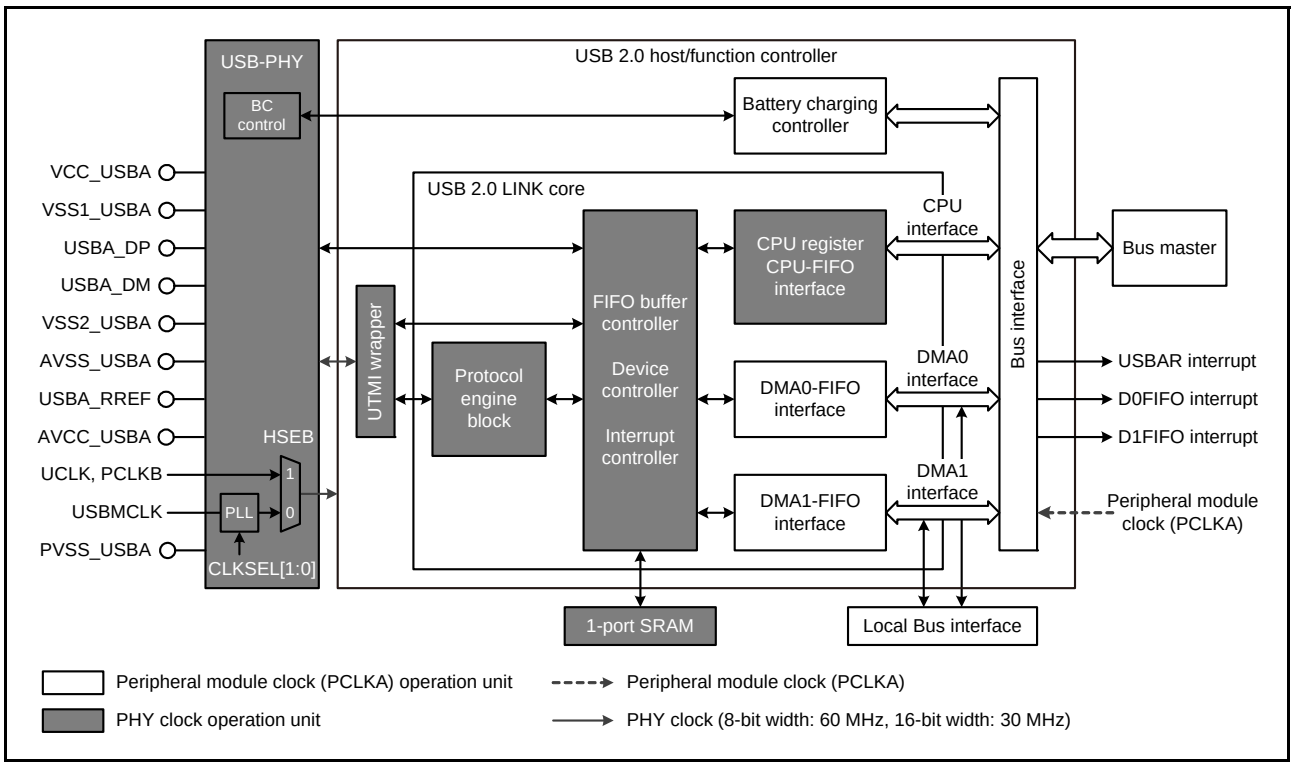


Figure 39.1 USBA Block Diagram

Table 39.2 lists the I/O pins of the USBA.

Table 39.2 Pin Configuration

Pin Symbol	I/O	Function
VCC_USBA	Input	Power supply pin for the USBA
VSS1_USBA VSS2_USBA	Input	Ground pin for the USBA
AVCC_USBA	Input	Analog power supply pin for the USBA
AVSS_USBA	Input	Analog ground pin for the USBA. Must be shorted to the PVSS_USBA pin.
PVSS_USBA	Input	PLL circuit ground pin for the USBA. Must be shorted to the AVSS_USBA pin.
USBA_RREF	I/O	Reference current source pin for the USBA. This pin should be connected to the AVSS_USBA pin through a resistor of 2.2 kΩ (±1%).
USBA_DP	I/O	I/O pin for the D+ data line of the USB bus
USBA_DM	I/O	I/O pin for the D– data line of the USB bus
USBA_EXICEN	Output	This pin should be connected to the OTG power supply IC.
USBA_ID	Input	This pin should be connected to the OTG power supply IC.
USBA_VBUSEN	Output	VBUS power enable pin for USB
USBA_OVRCURA USBA_OVRCURB	Input	Overcurrent pin for USB
USBA_VBUS	Input	USB cable connection monitor input pin

39.2 Register Descriptions

39.2.1 System Configuration Control Register (SYSCFG)

Address: 000D 0400h

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	—	CNEN
Value after reset:	x	x	x	x	x	x	x	0
	b7	b6	b5	b4	b3	b2	b1	b0
	—	DCFM	DRPD	DPRPU	—	—	—	USBE
Value after reset:	0	0	1	0	x	x	x	0

Bit	Symbol	Bit Name	Description	R/W
b0	USBE	USB Operation Enable	0: USB operation is disabled. 1: USB operation is enabled.	R/W
b3 to b1	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b4	DPRPU	D+ Line Resistor Control	0: Pulling up the line is disabled. 1: Pulling up the line is enabled.	R/W
b5	DRPD	D+/D– Line Resistor Control	0: Pulling down the line is disabled. 1: Pulling down the line is enabled.	R/W
b6	DCFM	Controller Operation Select	0: Function controller operation is selected. 1: Host controller operation is selected.	R/W
b7	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b8	CNEN	Single End Receiver Enable	0: Single end receiver operation is disabled. 1: Single end receiver operation is enabled.	R/W
b15 to b9	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

The SYSCFG register can be write accessed even when the LPSTS.SUSPENDM bit is 0 (PHY clock is stopped). Note, however, that written values are only reflected in the SYSCFG register after the LPSTS.SUSPENDM bit is set to 1 (PHY clock is supplied to the LINK).

USBE Bit (USB Operation Enable)

The USBE bit enables and disables USBA operation. Use the PHYSET.CLKSEL[1:0] bits to specify the input clock and verify that the PLLSTA.PLLLOCK flag is 1 before setting the USBE bit to 1. When operating in classic-only mode, set the PHYSET.HSEB bit to 1 before setting the USBE bit to 1. At that time, the UCLK must be set to 48 MHz and PCLKB must be set to 60 MHz. Refer to section 39.3.3 for details on setting clocks.

In host controller operation, set the DRPD bit to 1, eliminate chattering of the SYSSTS0.LNST[1:0] flags, and confirm that the USB bus state is stable before setting the USBE bit to 1.

When modifying the USBE bit from 1 to 0, some bits are initialized. Refer to Table 39.3 for a list of the bits initialized from setting the USBE bit to 0.

Table 39.3 Bits Initialized by Writing 0 to the SYSCFG.USBE Bit

Selected Function	Bits	Remarks
Function controller operation (when the DCFM bit is 0)	SYSSTS0.LNST[1:0]	The value is retained in host controller operation.
	DVSTCTR0.RHST[2:0]	
	INTSTS0.DVSQ[2:0]	The value is retained in host controller operation.
	USBADDR.USBADDR[6:0]	The value is retained in host controller operation.
	USBREQ.BREQUEST[7:0] and BMREQUESTTYPE[7:0]	The value is retained in host controller operation.
	USBVAL.WVALUE[15:0]	The value is retained in host controller operation.
	USBINDX.WINDEX[15:0]	The value is retained in host controller operation.
	USBLENG.WLENGTH[15:0]	The value is retained in host controller operation.
Host controller operation (when the DCFM bit is 1)	DVSTCTR0.RHST[2:0]	
	FRMNUM.FRNM[10:0]	The value is retained in function controller operation.

DPRPU Bit (D+ Line Resistor Control)

In function controller operation, the DPRPU bit enables and disables pulling up the D+ line.
When the DPRPU bit is set to 1 in function controller operation, the USBA pulls up the D+ line and notifies the USB host of the attached state. Modifying the DPRPU bit from 1 to 0 stops the USBA from pulling up the D+ line, thus it appears that the USBA is detached from the USB host.
Set the DPRPU bit to 1 in function controller operation, and set it to 0 in host controller operation.

DRPD Bit (D+/D– Line Resistor Control)

In host controller operation, the DRPD bit enables and disables pulling down the D+ and D– lines.
Set the DRPD bit to 1 in host controller operation. Set the DRPD bit to 0 when OTG is not used in function controller operation.

DCFM Bit (Controller Operation Select)

The DCFM bit selects the function of the USBA. Modify this bit when both the DPRPU and DRPD bits are 0.

CNEN Bit (Single End Receiver Enable)

Setting the CNEN bit to 1 enables single-ended receiver operation and enables the SYSSTS0.LNST[1:0] flags to monitor the states of the D+ and D– lines.
When the USBA is detached, the CNEN bit is used to prevent through-current damage which may occur from a single-ended receiver floating.
In host controller operation, verify that the PHY clock is being supplied before setting this bit to 1. In function controller operation, set this bit to 1 when the VBUS is detected due to a VBUS interrupt, and set the bit to 0 when the VBUS is removed.

39.2.2 CPU Bus Wait Register (BUSWAIT)

Address: 000D 0402h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	BWAIT[3:0]			
Value after reset:	x	x	0	0	1	1	1	1	x	x	0	0	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	BWAIT[3:0]	CPU Bus Access Wait Select	b3 b0 0 0 0 0: 0 waits (2 access cycles) 0 0 0 1: 1 wait (3 access cycles) 0 0 1 0: 2 waits (4 access cycles) 0 0 1 1: 3 waits (5 access cycles) 0 1 0 0: 4 waits (6 access cycles) 0 1 0 1: 5 waits (7 access cycles) 0 1 1 0: 6 waits (8 access cycles) 0 1 1 1: 7 waits (9 access cycles) 1 0 0 0: 8 waits (10 access cycles) 1 0 0 1: 9 waits (11 access cycles) 1 0 1 0: 10 waits (12 access cycles) 1 0 1 1: 11 waits (13 access cycles) 1 1 0 0: 12 waits (14 access cycles) 1 1 0 1: 13 waits (15 access cycles) 1 1 1 0: 14 waits (16 access cycles) 1 1 1 1: 15 waits (17 access cycles) (initial value)	R/W
b5, b4	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b7, b6	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b11 to b8	—	Reserved	These bits are 1 when read. Set them to 1 when writing.	R/W
b13, b12	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b15, b14	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

BWAIT[3:0] Bits (CPU Bus Access Wait Select)

The BWAIT[3:0] bits specify the wait period to access registers associated with the USBA.

Among the registers associated with the USBA, when consecutively accessing registers allocated to address 000D 0404h and higher, the access cycle time must be at least 67 ns. Though the initial value of the BWAIT[3:0] bits is 1111b (17 access cycles), satisfy this condition by setting the best value for wait period in accordance with the frequency of the CPU clock. This setting is the same as the waits in accesses to the FIFO port register. The maximum speed for accessing the FIFO port registers is as follows:

MBW[1:0] bits are 00b (8-bit width): 15 Mbytes/s

MBW[1:0] bits are 01b (16-bit width): 30 Mbytes/s

MBW[1:0] bits are 10b (32-bit width): 60 Mbytes/s

39.2.3 System Configuration Status Register (SYSSTS0)

Address: 000D 0404h

	b15	b14	b13	b12	b11	b10	b9	b8
	OVCMON[1:0]		—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x

	b7	b6	b5	b4	b3	b2	b1	b0
	—	HTACT	SOFEA	—	—	IDMON	LNST[1:0]	
Value after reset:	x	0	0	x	x	x	x	x

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LNST[1:0]	USB Data Line Status Monitor Flags	These flags indicate the status of the USB data line. See Table 39.4 for details.	R
b2	IDMON	USBA_ID Pin Monitor Flag	0: Low is input to the USBA_ID pin. 1: High is input to the USBA_ID pin.	R
b4, b3	—	Reserved	These bits are undefined when read.	R
b5	SOFEA	SOF Active Monitor Flag While in Host Controller Operation	0: SOF output is stopped. 1: SOF output is continued.	R
b6	HTACT	Host Sequencer Status Monitor Flag	0: Host sequencer is stopped. 1: Host sequencer is operating.	R
b13 to b7	—	Reserved	These bits are undefined when read.	R
b15, b14	OVCMON[1:0]	USBA_OVRCURA/USBA_OVRCURB Pin Monitor Flags	The OVCMON[1] flag indicates the status of the USBA_OVRCURA pin. The OVCMON[0] flag indicates the status of the USBA_OVRCURB pin.	R

LNST[1:0] Flags (USB Data Line Status Monitor Flags)

These flags indicate the state of the USB data lines (D+ line and D– line). See Table 39.4 for details. In function controller operation, set the SYSCFG.CNEN and USBE bits to 1 before reading the LNST[1:0] flags. In host controller operation, set the SYSCFG.DRPD bit to 1 before reading the LNST[1:0] flags.
However, for hardware-based contact checking using the battery charging function in function controller operation, set the SYSCFG.DRPD, SYSCFG.CNEN, and BCCTRL.IDPSRCE bits to 1 before reading the LNST[1:0] flags. Refer to section 39.3.15, Battery Charging Detection Processing for details.

Table 39.4 Status of USB Data Bus Lines

LNST[1:0] Flags	During Low-Speed Operation (Only in Host Controller Operation)	During Full-Speed Operation
00b	SE0	SE0
01b	K state	J state
10b	J state	K state
11b	SE1	SE1

SOFEA Flag (SOF Active Monitor Flag While in Host Controller Operation)

When the USB bus enters the Suspend state by setting the DVSTCTR0.UACT bit to 0 in host controller operation, the SOFEA flag is used to check whether the output of the last SOF is completed. If the SYSCFG.USBE bit is set to 0 to stop the USBA module and the LPSTS.SUSPENDM bit is set to 0 to stop supply of the clock signal while communications are in progress in host controller operation, make sure the HTACT and SOFEA flags are set to 0.

HTACT Flag (Host Sequencer Status Monitor Flag)

The HTACT flag is set to 0 when the host sequencer of the USBA is completely stopped. In host controller operation, make sure the HTACT flag is set to 0 when setting the DVSTCTR0.UACT bit to 0 to place the USBA interface in the USB bus Suspend state or setting the LPSTS.SUSPENDM bit to 0 to stop the clock. In addition, when communicating in host controller operation, make sure the HTACT flag and SOFEA flag are both 0 when setting the SYSCFG.USBE bit to 0 to stop the USBA, or when setting the LPSTS.SUSPENDM bit to 0 to stop the clock.

OVCMON[1:0] Flags (USBA_OVRCURA/USBA_OVRCURB Pin Monitor Flags)

The OVCMON[1] flag indicates the status of the USBA_OVRCURA pin, and the OVCMON[0] flag indicates the status of the USBA_OVRCURB pin.

39.2.4 PLL Status Register (PLLSTA)

Address: 000D 0406h

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PLLLOCK
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Bit Name	Description	R/W
b0	PLLLOCK	PLL Lock Flag	0: PLL is not locked. 1: PLL is locked.	R
b15 to b1	—	Reserved	These bits are undefined when read.	R

PLLLOCK Flag (PLL Lock Flag)

Indicates whether the UTMI-PHY internal PLL is locked or not. When classic-only mode is not used, make sure the PLL is locked before starting USB communication.

39.2.5 Device State Control Register 0 (DVSTCTR0)

Address: 000D 0408h

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	HNPBTOA	EXICEN	VBUSEN	WKUP
Value after reset:	x	x	x	x	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
Value after reset:	0	0	0	0	x	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RHST[2:0]	USB Reset Handshake Status Flags	• In host controller operation b2 b0 0 0 0: Communication speed not determined (powered state or disconnected) 0 0 1: Low-speed connection 0 1 0: Full-speed connection 0 1 1: Reserved 1 0 0: USB reset handshake in progress 1 0 1: USB reset handshake in progress 1 1 0: USB reset handshake in progress 1 1 1: USB reset handshake in progress • In function controller operation b2 b0 0 0 0: Communication speed not determined (powered state or disconnected) 0 0 1: Reserved 0 1 0: Full-speed connection 0 1 1: Reserved 1 0 0: Reserved 1 0 1: Reserved 1 1 0: Reserved 1 1 1: Reserved	R
b3	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b4	UACT	USB Bus Operation Enable for Host Controller Operation	0: Downstream port is disabled (SOF transmission is disabled). 1: Downstream port is enabled (SOF transmission is enabled).	R/W
b5	RESUME	Resume Signal Output for Host Controller Operation	0: Resume signal is not output. 1: Resume signal is output.	R/W
b6	USBRST	USB Bus Reset Output for Host Controller Operation	0: USB bus reset signal is not output. 1: USB bus reset signal is output.	R/W
b7	RWUPE	Remote Wakeup Detection Enable for Host Controller Operation	0: Downstream port remote wakeup detection is disabled 1: Downstream port remote wakeup detection is enabled	R/W
b8	WKUP	Remote Wakeup Output for Function Controller Operation	0: Remote wakeup signal is not output. 1: Remote wakeup signal is output.	R/W
b9	VBUSEN	USBA_VBUSEN Output Pin Control	0: The external USBA_VBUSEN pin outputs the low level. 1: The external USBA_VBUSEN pin outputs the high level.	R/W
b10	EXICEN	USBA_EXICEN Output Pin Control	0: The external USBA_EXICEN pin outputs the low level. 1: The external USBA_EXICEN pin outputs the high level.	R/W
b11	HNPBTOA	Host Negotiation Protocol (HNP) Control	0: HNP not enabled 1: HNP is enabled	R/W
b15 to b12	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

RHST[2:0] Flags (USB Reset Handshake Status Flags)

These flags indicate that a reset handshake is in progress.

1. In host controller operation
   When the USBRST bit is set to 1, the RHST[2:0] flags become 100b and then become 110b or 101b. When the USBRST bit is set to 0, the USBA stops the USB bus reset signal, and the USBA fixes the RHST[2:0] flag value.
2. In function controller operation
   When the USBA detects a USB bus reset, the RHST[2:0] flags become 010b. After the USBA detects the USB bus reset, when the RHST[2:0] flags are confirmed to be 010b, the device state transition interrupt is generated.

UACT Bit (USB Bus Operation Enable for Host Controller Operation)

When the UACT bit is set to 1 in host controller operation, the USBA enables the USB bus for the USB port.
The USBA starts the output of an SOF packet within one frame period after the UACT bit is set to 1.
If the UACT bit is set to 0, the USBA enters the Idle state after an SOF packet is output.
The USBA sets the UACT bit to 0 on either of the following conditions.

- A DTCH interrupt is detected
- An EOFERR interrupt is detected

Set the UACT bit to 1 either when the USB bus reset processing is complete (USBRST bit is set to 0), or when the USBA has exited the Suspend state (RESUME bit is set to 0).
When the HL1CTRL1.L1REQ bit is set to 1 to issue an LPM token and an ACK response is received, the USBA sets the UACT bit to 0. In addition, after exiting the L1 state, the USBA sets the UACT bit to 1.
In function controller operation, set the UACT bit to 0.

RESUME Bit (Resume Signal Output for Host Controller Operation)

This bit controls the output of the resume signal in host controller operation.
If the RESUME bit set to 1 while in the Suspend state, the USBA outputs the resume signal to the USB port. In addition, if the RWUPE bit is 1 and a remote wakeup signal is detected while in the Suspend state, the USBA sets the RESUME bit to 1, and USBA outputs the resume signal to the USB port.
The USBA continues outputting the resume signal until the RESUME bit is set to 0 by software. Ensure that the period the RESUME bit is 1 (resume period) complies with the time defined in USB 2.0. In addition, set the UACT bit to 1 at the same time the resume processing is complete (RESUME bit is set to 0).
If the RESUME bit to 1 during the L1 state, the USBA outputs the resume signal to the USB port. The resume period is the period indicated by the HL1CTRL2.HIRD[3:0] bits. The USBA sets the RESUME bit to 0 at the end of the resume period.
Only set the RESUME bit is 1 while in the Suspend state or L1 state. In addition, set the RESUME bit to 0 in function controller operation.

USBRST Bit (USB Bus Reset Output for Host Controller Operation)

In host controller operation, this bit controls the output of the USB bus reset signal.
In host controller operation, if the USBRST bit is set to 1, the USBA performs a USB bus reset on the USB port.
The USBA continues outputting the bus reset signal until the USBRST bit is set to 0 by software. Ensure that the period the USBRST bit is 1 (USB bus reset period) complies with the time defined in USB 2.0.
When setting the USBRST bit to 1 either during communication (UACT bit is 1), or during the resume processing (RESUME bit is 1), the USBA prevents the USB but reset processing from starting until both the UACT and RESUME bits become 0. Set the UACT bit to 1 at the same time the USB bus reset processing is complete (USBRST bit is set to 0).
In function controller operation, set the USBRST bit to 0.

RWUPE Bit (Remote Wakeup Detection Enable for Host Controller Operation)

In host controller operation, this bit enables and disables remote wakeup signals (resume signals) from peripheral devices connected to downstream ports.
While the RWUPE bit set to 1, the USBA performs resume processing (resume signal is output) when a remote wakeup signal (K state for 2.5 µs) is detected from a peripheral device connected to a downstream port.
While the RWUPE bit is 0, even if the USBA detects a remote wakeup signal from a peripheral device connected to the USB port, the remote wakeup signal is ignored.
When the RWUPE bit is set to 1, do not stop the PHY clock even in the Suspend state (keep the LPSTS.SUSPENDM bit to 1). Also, executing a USB bus reset (setting the USBRST bit to 1) while in the Suspend state is prohibited in USB 2.0.
In addition, the RWUPE bit is used to enable and disable detection of the remote wakeup signal while in the L1 state.
In function controller operation, set the RWUPE bit to 0.

WKUP Bit (Remote Wakeup Output for Function Controller Operation)

In function controller operation, this bit is used to output the remote wakeup signal (resume signal) to the USB port.
The USBA controls the output time of the remote wakeup signal. When the WKUP bit is set to 1, the USBA sets this bit to 0 after outputting a 10-ms remote wakeup signal.
According to USB 2.0, the USBA must be in the USB bus Idle state for at least 5 ms before the remote wakeup signal is transmitted. For this reason, even if the WKUP bit is set to 1 immediately after the USB bus is detected as being in the suspended state, the USBA waits 2 ms before outputting the remote wakeup signal.
Only set the WKUP bit to 1 when the device state is in the suspended state (INTSTS0.DVSQ[2:0] flags are 1xxb) and the remote wakeup signaling is enabled by the USB host.
Also, when the WKUP bit is set to 1, do not stop the PHY clock even if the USBA is in the suspended state (keep the LPSTS.SUSPENDM bit to 1).
If the WKUP bit is set to 1 while the USBA is in the L1 state, the USBA outputs the remote wakeup signal for 50 µs and then sets the WKUP bit to 0.
In host controller operation, set the WKUP bit to 0.

HNPBTOA Bit (Host Negotiation Protocol (HNP) Control)

When the USBA is used as an OTG device B, set this bit to switch from function controller operation to host controller operation.
If the HNPBTOA bit is 1, the USBA remains in the suspended state until HNP processing is complete even if the SYSCFG.DPRPU bit is set to 0 or the SYSCFG.DCFM bit is set to 1. At this time, a resume interrupt is not generated even if a falling edge on the D+ line is detected.
After setting the HNPBTOA bit to 1, when attachment of a host is detected (due to pulling up by the other party) or if a timeout occurs during HNP processing, set the HNPBTOA bit to 0 to end HNP processing.

39.2.6 FIFO Port Registers (CFIFO, D0FIFO, D1FIFO)

(1) FIFO port register values when the MBW[1:0] bits are 10b (32-bit width)

Addresses CFIFO 000D 0414h, D0FIFO 000D 0418h, D1FIFO 000D 041Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol																																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	—	FIFO Port	From this port, receive data is read from the FIFO buffer and transmit data is written to the FIFO buffer.	R/W

(2) FIFO port register values when the MBW[1:0] bits are 01b (16-bit width) and the BIGEND bit is 0 (Little endian)

Addresses CFIFO 000D 0416h, D0FIFO 000D 041Ah, D1FIFO 000E 041Eh

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	H[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	H[15:0]	FIFO Port	From this port, receive data is read from the FIFO buffer and transmit data is written to the FIFO buffer.	R/W

(3) FIFO port register values when the MBW[1:0] bits are 01b (16-bit width) and the BIGEND bit is 1 (Big endian)

Addresses CFIFO 000D 0414h, D0FIFO 000D 0418h, D1FIFO 000D 041Ch

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	L[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	L[15:0]	FIFO Port	From this port, receive data is read from the FIFO buffer and transmit data is written to the FIFO buffer.	R/W

(4) FIFO port register values when the MBW[1:0] bits are 00b (8-bit width) and the BIGEND bit is 0 (Little endian)

Addresses CFIFO 000D 0417h, D0FIFO 000D 041Bh, D1FIFO 000F 041Fh

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	HH[7:0]							
Value after reset	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	HH[7:0]	FIFO Port	From this port, receive data is read from the FIFO buffer and transmit data is written to the FIFO buffer.	R/W

(5) FIFO port register values when the MBW[1:0] bits are 00b (8-bit width) and the BIGEND bit is 1 (Big Endian)

Addresses CFIFO 000D 0414h, D0FIFO 000D 0418h, D1FIFO 000D 041Ch

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	LL[7:0]							
Value after reset	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	LL[7:0]	FIFO Port	From this port, receive data is read from the FIFO buffer and transmit data is written to the FIFO buffer.	R/W

There are three FIFO ports: CFIFO, D0FIFO, and D1FIFO. Each FIFO port is configured of a FIFO port register, FIFO port select register, and a FIFO port control register. The FIFO port registers (CFIFO, D0FIFO, and D1FIFO) read data from and write data to the FIFO buffers. The FIFO port select registers (CFIFOSEL, D0FIFOSEL, and D1FIFOSEL) are used to select the pipe assigned to the FIFO port. The FIFO port control registers are CFIFOCTR, D0FIFOCTR, and D1FIFOCTR.

The following are features and precautions pertaining to the FIFO ports:

- Access the FIFO buffer for the default control pipe (control transfer) through the CFIFO port.
- Accessing a FIFO buffer using DMA/DTC transfer should be performed through the D0FIFO or D1FIFO port.
- The D0FIFO and D1FIFO ports can also be accessed by the CPU.
- When using functions specific to a FIFO port such as a DMA/DTC transfer function, the pipe number (current pipe) set in the CURPIPE[3:0] bits of the FIFO port select register cannot be changed.
- Registers configuring one FIFO port do not affect other FIFO ports.
- The same pipe should not be assigned to two or more FIFO ports.
- Access rights for a FIFO buffer can be on the CPU side or on the serial interface engine (SIE) side. When the access right for a FIFO buffer is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

FIFO Port Bits

Using the FIFO port bits, the received data can be read from the FIFO buffer and the transmit data can be written to the FIFO buffer. The FIFO port registers can only be accessed when the FRDY flag in the corresponding FIFO port control register is 1. Valid bits in the FIFO port registers depend on the MBW[1:0] and BIGEND bit values in the corresponding FIFO port select register. When the MBW[1:0] bits are 01b or 10b, data allocation changes in relation to the MDE.MDE[2:0] bit value and the CFIFOSEL.BIGEND, D0FIFOSEL.BIGEND, and D1FIFOSEL.BIGEND bits. Table 39.5 and Table 39.6 show the various data allocation.

Table 39.5 Data Allocation in a 32-Bit Access (MBW[1:0] Bits are 10b)

MDE[2:0] Bits	BIGEND Bit	b31 to b24	b23 to b16	b15 to b8	b7 to b0	Remarks
000b (big endian)	0 (little endian)	Address N+3	Address N+2	Address N+1	Address N+0	Data is transmitted sequentially from address N+3. Data received is stored sequentially from address N+3.
	1 (big endian)	Address N+0	Address N+1	Address N+2	Address N+3	Data is transmitted sequentially from address N+0. Data received is stored sequentially from address N+0.
111b (little endian)	0 (little endian)	Address N+3	Address N+2	Address N+1	Address N+0	Data is transmitted sequentially from address N+0. Data received is stored sequentially from address N+0.
	1 (big endian)	Address N+0	Address N+1	Address N+2	Address N+3	Data is transmitted sequentially from address N+3. Data received is stored sequentially from address N+3.

Table 39.6 Data Allocation in a 16-Bit Access (MBW[1:0] Bits are 01b)

MDE[2:0] Bits	BIGEND Bit	b15 to b8	b7 to b0	Remarks
000b (big endian)	0 (little endian)	Address N+1	Address N+0	Access the H[15:0] bits. Data is transmitted sequentially from address N+1. Data received is stored sequentially from address N+1.
	1 (big endian)	Address N+0	Address N+1	Access the L[15:0] bits. Data is transmitted sequentially from address N+0. Data received is stored sequentially from address N+0.
111b (little endian)	0 (little endian)	Address N+1	Address N+0	Access the H[15:0] bits. Data is transmitted sequentially from address N+0. Data received is stored sequentially from address N+0.
	1 (big endian)	Address N+0	Address N+1	Access the L[15:0] bits. Data is transmitted sequentially from address N+1. Data received is stored sequentially from address N+1.

39.2.7 CFIFO Port Select Register (CFIFOSEL)

Address: 000D 0420h

	b15	b14	b13	b12	b11	b10	b9	b8
	RCNT	REW	—	—	MBW[1:0]		—	BIGEND
Value after reset:	0	0	x	x	0	0	x	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	ISEL	—	CURPIPE[3:0]			
Value after reset:	x	x	0	x	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE[3:0]	FIFO Port Access Pipe Select	b3 b0 0 0 0 0: Pipe 0 (default control pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Only set the values listed above.	R/W
b4	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b5	ISEL	FIFO Port Access Direction When Default Control Pipe is Selected	0: Reading from the FIFO buffer is selected 1: Writing to the FIFO buffer is selected	R/W
b7, b6	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b8	BIGEND	FIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b11, b10	MBW[1:0]	CFIFO Port Access Bit Width	b11 b10 0 0: 8-bit width 0 1: 16-bit width 1 0: 32-bit width 1 1: Do not set this value.	R/W
b13, b12	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b14	REW	Buffer Pointer Rewind	Set this bit to 1 to rewind the buffer pointer. Writing 0 has no effect.	R/W
b15	RCNT	Read Count Mode	0: The CFIFOCTR.DTLN[11:0] flags are set to 000h when all of the receive data has been read from the CFIFO register. 1: The CFIFOCTR.DTLN[11:0] flags are decremented each time the receive data is read from the CFIFO register.	R/W

The same pipe number should not be set to the CURPIPE[3:0] bits in registers CFIFOSEL, D0FIFOSEL, and D1FIFOSEL. In addition, do not change the pipe number while DMA or DTC transfer is enabled.

CURPIPE[3:0] Bits (FIFO Port Access Pipe Select)

The CURPIPE[3:0] bits specify the current pipe number for which data is read or written through the CFIFO port. When modifying the CURPIPE[3:0] bits, first set them to 0000b, and then set the desired pipe number. After setting the desired pipe number, read the CURPIPE[3:0] bits, confirm that the value set and the value read match, and then proceed to the next processing.

Do not set the same pipe number in the CURPIPE[3:0] bits in registers CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

If the CURPIPE[3:0] bits are modified while accessing the FIFO buffer, access until that point is retained, and after the CURPIPE[3:0] bits are modified back to the original setting, access can be resumed.

ISEL Bit (FIFO Port Access Direction When Default Control Pipe is Selected)

If the ISEL bit is modified while the default control pipe is the current pipe, read the bit to confirm it has been modified. Set the ISEL bit and the CURPIPE[3:0] bits simultaneously.

BIGEND Bit (FIFO Port Endian Control)

Set this bit to specify the byte endian of the CFIFO port.

MBW[1:0] Bits (CFIFO Port Access Bit Width)

Set this bit to specify the bit width for accessing the CFIFO port.
When the transfer direction of the current pipe is receiving, if the MBW[1:0] bits are set to start reading data from the FIFO buffer, do not modify the MBW[1:0] bits until all the data has been read. When modifying the MBW[1:0] bits, set the CURPIPE[3:0] bits and MBW[1:0] bits simultaneously after setting the CURPIPE[3:0] bits to 0000b.
When the transfer direction of the current pipe is transmitting, while data is being written to the FIFO buffer, the bit width cannot be changed from 8-bit width to 16- or 32-bit width, and 16-bit width cannot be changed to 32-bit width. Also, even when 16-bit or 32-bit width is selected, an odd number of bytes can be written by byte access.

REW Bit (Buffer Pointer Rewind)

When the transfer direction of the current pipe is receiving, if the REW bit is set to 1 while reading data from the FIFO buffer, the FIFO buffer can be reread from the first data. In double buffer mode, the buffer that was being read can be reread from the first data.
Do not modify the CURPIPE[3:0] bits when setting the REW bit to 1. Also, confirm that the FRDY flag is 1 before setting the REW bit to 1.
For pipes in the receiving transfer direction, to rewrite to the FIFO buffer from the first data, set the BCLR bit.

RCNT Bit (Read Count Mode)

When the RCNT bit is set to 0, when all data has been read from the FIFO buffer (in double buffer mode, when a single plane is done being read) for the pipe specified by the CURPIPE[3:0] bits, the USBA sets the CFIFOCTR.DTLN[11:0] flags to 000h.
When the RCNT bit set to 1, each time data is read from the FIFO buffer for the pipe specified by the CURPIPE[3:0] bits, the USBA decrements the value of the CFIFOCTR.DTLN[11:0] flags.

39.2.8 D0FIFO Port Select Register (D0FIFOSEL)
D1FIFO Port Select Register (D1FIFOSEL)

Addresses: D0FIFOSEL 000D 0428h, D1FIFOSEL 000D 042Ch

	b15	b14	b13	b12	b11	b10	b9	b8
	RCNT	REW	DCLRM	DREQE	MBW[1:0]		—	BIGEND
Value after reset:	0	0	0	0	0	0	x	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CURPIPE[3:0]			
Value after reset:	x	x	x	x	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE[3:0]	FIFO Port Access Pipe Select	b3 b0 0 0 0 0: No pipe selected 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Only set the values listed above.	R/W
b7 to b4	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b8	BIGEND	FIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b11, b10	MBW[1:0]	FIFO Port Access Bit Width	b11 b10 0 0: 8-bit width 0 1: 16-bit width 1 0: 32-bit width 1 1: Do not set this value.	R/W
b12	DREQE	DMA/DTC Transfer Request Enable	0: DMA/DTC transfer request is disabled. 1: DMA/DTC transfer request is enabled.	R/W
b13	DCLRM	Automatic FIFO Buffer Clear Mode after Current Pipe is Read	0: Automatic buffer clear mode is disabled. 1: Automatic buffer clear mode is enabled.	R/W
b14	REW	Buffer Pointer Rewind	Set this bit to 1 to rewind the buffer pointer. Writing 0 has no effect.	R/W
b15	RCNT	Read Count Mode	0: The DTLN[11:0] flags in the D0FIFOCTR/D1FIFOCTR register are set to 000h when all of the receive data has been read from D0FIFO/D1FIFO (after reading a single plane in double buffer mode). 1: The DTLN[11:0] flags in the D0FIFOCTR/D1FIFOCTR register are decremented each time the receive data is read from D0FIFO/D1FIFO.	R/W

The same pipe number should not be set to the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. In addition, a pipe is not assigned if the setting of the D0FIFOSEL/D1FIFOSEL.CURPIPE[3:0] bits is 0000b. Do not change the pipe number while DMA or DTC transfer is enabled.

CURPIPE[3:0] Bits (FIFO Port Access Pipe Select)

The CURPIPE[3:0] bits specify the current pipe number for which data is read or written through the D0FIFO or D1FIFO port.

When modifying the CURPIPE[3:0] bits, write a value to the CURPIPE[3:0] bits, then read the bits to confirm that the value written and the value read match written value agree, and then proceed to the next processing.

Do not set the same pipe number in the CURPIPE[3:0] bits in registers CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

If the CURPIPE[3:0] bits are modified while accessing the FIFO buffer, access until that point is retained, and after the CURPIPE[3:0] bits are modified back to the original setting, access can be resumed.

BIGEND Bit (FIFO Port Endian Control)

Set this bit to specify the byte endian of the D0FIFO and D1FIFO port.

MBW[1:0] Bits (FIFO Port Access Bit Width)

This bit specifies the bit width for accessing the D0FIFO or D1FIFO port.
When the current pipe is in the receiving direction, once reading data from the FIFO buffer is started after setting the MBW[1:0] bits, these bits should not be modified until all the data has been read. When modifying the MBW[1:0] bits, set the CURPIPE[3:0] bits and the MBW[1:0] bits simultaneously after setting the CURPIPE[3:0] bits to 0000b.
When the current pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit/32-bit width or 16-bit width to 32-bit width while data is being written to the FIFO buffer. An odd number of bytes can also be written through byte-access control even when 16-bit or 32-bit width is selected.

DREQE Bit (DMA/DTC Transfer Request Enable)

This bit enables or disables the DMA/DTC transfer request. To enable the DMA/DTC transfer request, set this bit to 1 after setting the CURPIPE[3:0] bits. To change the setting of the CURPIPE[3:0] bits, do it after setting this bit to 0.

DCLRM Bit (Automatic FIFO Buffer Clear Mode after Current Pipe is Read)

This bit enables or disables the automatic FIFO buffer clear after data in the current pipe is read. When the DCLRM bit is set to 1 on receiving a zero-length packet while the FIFO buffer assigned to the current pipe is empty, or when reading the received short packet is completed while the PIPECFG.BFRE bit is 1, the USBA sets the BCLR bit in a FIFO port control register to 1. Set the DCLRM bit to 0 when the SOFCFG.BRDYM bit 1.

REW Bit (Buffer Pointer Rewind)

When the current pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data. When double-buffering is in use and reading is already in progress, reading either FIFO buffer from the first entry is possible. Do not set the REW bit to 1 simultaneously with modifying the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY flag is 1. To rewrite to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

RCNT Bit (Read Count Mode)

With the RCNT bit set to 0, the USBA sets the DTLN[11:0] flags in the D0FIFOCTR/D1FIFOCTR register to 000h upon finishing reading all the received data in the FIFO buffer assigned to the pipe specified in the CURPIPE[3:0] bits (after reading a single plane in double buffer mode). With the RCNT bit set to 1, the USBA decrements the value indicated by the DTLN[11:0] flags each time it reads data received from the FIFO buffer assigned to the pipe specified in the CURPIPE[3:0] bits. Set the RCNT bit to 0 when accessing D0FIFO/D1FIFO with the PIPECFG.BFRE bit set to 1.

39.2.9 CFIFO Port Control Register (CFIFOCTR)
D0FIFO Port Control Register (D0FIFOCTR)
D1FIFO Port Control Register (D1FIFOCTR)

Addresses: CFIFOCTR 000D 0422h, D0FIFOCTR 000D 042Ah, D1FIFOCTR 000D 042Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BVAL	BCLR	FRDY	—	DTLN[11:0]											
Value after reset:	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	DTLN[11:0]	Receive Data Length Flags	These flags indicate the length of the receive data. The meaning of the value of these flags differs with the setting of the RCNT bit in the FIFO port select register. For details, see the more detailed description of the DTLN[11:0] flags below.	R
b12	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b13	FRDY	FIFO Port Ready Flag	0: FIFO port access is disabled 1: FIFO port access is enabled	R
b14	BCLR	CPU Buffer Clear	This bit should be set to 1 to clear the FIFO buffer on the CPU side for the pipe specified by the CUREPIPE[3:0] bits. Writing 0 has no effect. This bit is 0 when read.	R/W
b15	BVAL	FIFO Buffer Valid	This bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe specified by the CURPIPE[3:0] bits. Writing 0 has no effect.	R/W

The CFIFOCTR, D0FIFOCTR, and D1FIFOCTR registers respectively correspond to the CFIFO, D0FIFO, and D1FIFO buffers.

DTLN[11:0] Flags (Receive Data Length Flags)

These flags indicate the length of the receive data. While data is being read from the FIFO buffer, the value of the DTLN[11:0] flags differs according to the setting of the RCNT bit in the FIFO port select registers as described below.

- When the RCNT bit is 0
  The USBA sets the DTLN[11:0] flags to indicate the length of the receive data until the CPU or DMAC/DTC has read all the received data in the FIFO buffer (until a single plane has been read in double buffer mode).
  When the PIPECFG.BFRE bit for the current pipe is set to 1, the USBA retains the length of the receive data until the BCLR bit is set to 1 even after all the data has been read.
- When the RCNT bit is 1
  The USBA decrements the value indicated by the DTLN[11:0] flags each time the CPU or DMAC/DTC reads the receive data from the FIFO buffer. (The value is decremented by one when the MBW[1:0] bits are 00b, by two when the MBW[1:0] bits are 01b, and by four when the MBW[1:0] bits are 10b.)
  The DTLN[11:0] flags become 000h when all the data has been read from the FIFO buffer. However, in double buffer mode, if data reception is completed in one FIFO buffer plane before all the data has been read from the other plane, the DTLN[11:0] flags indicate the length of the receive data in the latter plane when all the data has been read from the former plane.
  In addition, updating the DTLN[11:0] flags can take up to 150 ns. After reading the FIFO buffer, wait at least 150 ns before reading the DTLN[11:0] flags.

FRDY Flag (FIFO Port Ready Flag)

This flag indicates whether the FIFO port can be accessed by the CPU or DMAC/DTC.
In the cases below, the FRDY flag also becomes 1, but data cannot be read via the FIFO port because there is no data to be read. In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

- A zero-length packet is received when the FIFO buffer assigned to the current pipe is empty.
- When the PIPECFG.BFRE bit for the current pipe is 1, a short packet is received and the data read operation is complete.

BCLR Bit (CPU Buffer Clear)

Set the BCLR bit to 1 to clear the FIFO buffer on the CPU side of the current pipe.
When double buffer mode is set for the FIFO buffer assigned to the current pipe, the USBA only clears one plane of the FIFO buffer even when both planes are readable.
When the current pipe is the default control pipe, setting the BCLR bit to 1 clears the FIFO buffer regardless of whether the FIFO buffer access rights are on the CPU side or the SIE side. To clear the FIFO buffer when the SIE has the access right, set the DCPCTR.PID[1:0] bits to 00b (NAK response) and then set the BCLR bit to 1.
When the current pipe is a pipe other than the default control pipe, set the BCLR bit to 1 when the FRDY flag in a FIFO port control register is 1.

BVAL Bit (FIFO Buffer Valid)

When the transfer direction of the pipe specified by the CURPIPE[3:0] bits is transmitting, set the BVAL bit to 1 in the cases listed below. When the BVAL bit is set to 1, the USBA switches the FIFO buffer access right from the CPU side to the SIE side, enabling transmission.

- To transmit a short packet, set this bit to 1 after transmit data has been written.
- To transmit a zero-length packet, set this bit to 1 before transmit data is written.
- For pipes in continuous transfer mode, set this bit to 1 after writing data that is an integral multiple of the maximum packet size and less than the buffer size.

When the BVAL bit and BCLR bit are set to 1 simultaneously, the USBA clears the data already written and enables the transmission of a zero-length packet. For pipes in discontinuous transfer mode, when the maximum packet size of data is written, the USBA sets the BVAL bit to 1, and switches the FIFO buffer access right from the CPU side to the SIE side, enabling transmission.
Set the BVAL bit to 1 while the FRDY flag is 1.
When the transfer direction of the current pipe is receiving, do not set the BVAL bit to 1.

39.2.10 Interrupt Enable Register 0 (INTENB0)

Address: 000D 0430h

	b15	b14	b13	b12	b11	b10	b9	b8
	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE
Value after reset:	0	0	0	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b8	BRDYE	Buffer Ready Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled	R/W
b9	NRDYE	Buffer Not Ready Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled	R/W
b10	BEMPE	Buffer Empty Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled	R/W
b11	CTRE	Control Transfer Stage Transition Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled	R/W
b12	DVSE	Device State Transition Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled	R/W
b13	SOFE	Frame Number Refresh Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled	R/W
b14	RSME	Resume Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled	R/W
b15	VBSE	VBUS Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled	R/W

Note: The CTRE bit, DVSE bit, and RSME bit can only be set to 1 in function controller operation. Set these bits to 0 in host controller operation.

When a status flag in the INTSTS0 register becomes 1, if the corresponding interrupt request enable bit in the INTENB0 register is 1, the USBA requests the USBAR interrupt.
Regardless of the INTENB0 register setting, if the conditions for each status are satisfied, the status flags corresponding to the INTSTS0 register become 1.
When the status flags in the INTSTS0 register are 1, if a corresponding interrupt request enable bit in the INTENB0 register is rewritten from 0 to 1, the USBAR interrupt request is generated.

39.2.11 Interrupt Enable Register 1 (INTENB1)

Address: 000D 0432h

	b15	b14	b13	b12	b11	b10	b9	b8
	OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	L1RSMENDE	LPMENDE
Value after reset:	0	0	x	0	0	x	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	EOFERRE	SIGNE	SACKE	—	—	—	PDDETINTE
Value after reset:	x	0	0	0	x	x	x	0

Bit	Symbol	Bit Name	Description	R/W
b0	PDDETINTE	Portable Device Detection Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled	R/W
b3 to b1	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b4	SACKE	Setup Transaction Normal Response Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b5	SIGNE	Setup Transaction Error Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b6	EOFERRE	End-of-Frame Error Detection Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b7	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b8	LPMENDE	LPM Transaction End Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b9	L1RSMENDE	L1 Resume End Interrupt Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b10	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b11	ATTCHE	Device Connection Detection Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b12	DTCHE	Device Disconnection Detection Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b13	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b14	BCHGE	Bus Change Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W
b15	OVRCRE	Overcurrent Change Interrupt Request Enable	0: Interrupt request output disabled 1: Interrupt request output enabled *1	R/W

Note 1. This bit can only be set to 1 in host controller operation. Set this bit to 0 in function controller operation.

When a status flag in the INTSTS1 register becomes 1, if the corresponding interrupt request enable bit in the INTENB1 register is 1, the USBA requests the USBAR interrupt.
Regardless of the INTENB1 register setting, if the conditions for each status are satisfied, the status flags corresponding to the INTSTS1 register become 1.
When the status flags in the INTSTS1 register are 1, if a corresponding interrupt request enable bit in the INTENB1 register is rewritten from 0 to 1, the USBAR interrupt request is generated.

39.2.12 BRDY Interrupt Enable Register (BRDYENB)

Address: 000D 0436h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPEBRDYE[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	PIPEBRDYE[9:0]	Pipe 9 to Pipe 0 BRDY Interrupt Request Enable *1	0: Interrupt request output disabled 1: Interrupt request output enabled	R/W
b15 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. Each bit corresponds to a pipe number (e.g. pipe 9 corresponds to the PIPEBRDYE[9] bit, pipe 8 corresponds to the PIPEBRDYE[8] bit, etc.).

When a flag in the BRDYSTS register becomes 1, if the corresponding BRDYENB.PIPEBRDYE[9:0] bit is 1, the INTSTS0.BRDY flag becomes 1.
In this case, if the INTENB0.BRDYE bit is 1, the USBA requests the USBAR interrupt.
When any of the BRDYSTS.PIPEBRDY[9:0] flags is 1, if the corresponding interrupt request enable bit in the BRDYENB register is rewritten from 0 to 1, the INTSTS0.BRDY flag becomes 1.

39.2.13 NRDY Interrupt Enable Register (NRDYENB)

Address: 000D 0438h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPENRDYE[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	PIPENRDYE[9:0]	Pipe 9 to Pipe 0 NRDY Interrupt Enable *1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. Each bit corresponds to a pipe number (e.g. pipe 9 corresponds to the PIPENRDYE[9] bit, pipe 8 corresponds to the PIPENRDYE[8] bit, etc.).

When a flag in the NRDYSTS register becomes 1, if the corresponding NRDYENB.PIPENRDYE[9:0] bit is 1, the INTSTS0.NRDY flag becomes 1.
In this case, if the INTENB0.NRDYE bit is 1, the USBA requests the USBAR interrupt.
When any of the NRDYSTS.PIPENRDY[9:0] flags is 1, if the corresponding interrupt request enable bit in the NRDYENB register is rewritten from 0 to 1, the INTSTS0.NRDY flag becomes 1.

39.2.14 BEMP Interrupt Enable Register (BEMPENB)

Address: 000D 043Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPEBEMPE[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	PIPEBEMPE[9:0]	Pipe 9 to Pipe 0 BEMP Interrupt Enable *1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. Each bit corresponds to a pipe number (e.g. pipe 9 corresponds to the PIPEBEMPE[9] bit, pipe 8 corresponds to the PIPEBEMPE[8] bit, etc.).

When a flag in the BEMPSTS register becomes 1, if the corresponding BEMPENB.PIPEBEMPE[9:0] bit is 1, the INTSTS0.BEMP flag becomes 1.
In this case, if the INTENB0.BEMPE bit is 1, the USBA requests the USBAR interrupt.
When any of the BEMPSTS.PIPEBEMP[9:0] flags is 1, if the corresponding interrupt request enable bit in the BEMPENB register is rewritten from 0 to 1, the INTSTS0.BEMP flag becomes 1.

39.2.15 SOF Output Configuration Register (SOFCFG)

Address: 000D 043Ch

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	—	TRNENSEL
Value after reset:	x	x	x	x	x	x	x	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	BRDYM	INTL	EDGESTS	—	—	—	—
Value after reset:	x	0	0	0	0	0	x	x

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b4	EDGESTS	Interrupt Edge Processing Status Flag *1	This flag is 1 when the USBA outputs an edge interrupt request signal.	R
b5	INTL	Interrupt Signal Output Method Select *2	0: Do not set this value (edge output) 1: Level output	R/W
b6	BRDYM	PIPEBRDY Interrupt Status Clear Timing *3	0: The PIPEBRDY[9:0] flags are set to 0 by software. 1: The PIPEBRDY[9:0] flags are set to 0 by reading the FIFO buffer or writing to the FIFO buffer.	R/W
b7	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b8	TRNENSEL	Transaction-Enabled Time Select *4	0: Not low-speed compatible 1: Low-speed compatible	R/W
b15 to b9	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

Note 1. Confirm that the EDGESTS flag is 0 before stopping the clock supplied to the USBA.
Note 2. Set this bit to 1 before operating the USBA.
Note 3. When setting the BRDYM bit to 1, set the INTL bit to 1.
Note 4. The TRNENSEL bit setting is only valid in host controller operation.

EDGESTS Flag (Interrupt Edge Processing Status Flag)

This flag is enabled when the INTL bit is 0 (edge output).
When multiple interrupt requests are generated, if one of the interrupt requests is cleared, the USBA temporarily negates the interrupt request signal, after waiting 666.6 μs, the interrupt request signal is reasserted. While the interrupt request signal is negated, the EDGESTS flag becomes 1.
Confirm the EDGESTS flag is 0 when stopping the PHY clock.

BRDYM Bit (PIPEBRDY Interrupt Status Clear Timing)

This bit specifies the timing to clear the BRDY interrupt status (BRDYSTS.PIPEBRDY[9:0] flags) for each pipe.

TRNENSEL Bit (Transaction-Enabled Time Select)

When the USB port is in use for full-speed or low-speed communications, this bit specifies the timing with which the USBA issues tokens in a frame (transaction-enabled time).
Set the TRNENSEL bit to 1 when a low-speed device is connected.
The TRNENSEL bit is only valid in host controller operation.
Set this bit to 0 in function controller operation.

39.2.16 PHY Setting Register (PHYSET)

Address: 000D 043Eh

	b15	b14	b13	b12	b11	b10	b9	b8
	HSEB	—	—	—	—	—	REPSEL[1:0]	
Value after reset:	0	x	x	x	0	x	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CLKSEL[1:0]		CDPEN	—	PLLRESET	DIRPD
Value after reset:	x	x	1	1	0	x	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	DIRPD	Power-Down Control	0: Power-down control not performed 1: Power-down control performed	R/W
b1	PLLRESET	PLL Reset Control *1	0: Disable PLL reset control for UTMI-PHY 1: Enable PLL reset control for UTMI-PHY	R/W
b2	—	Reserved	This bit is read as 0. Set it to 0 when writing.	R/W
b3	CDPEN	Charging Downstream Port Enable	0: Disable the charging downstream port 1: Enable the charging downstream port	R/W
b5, b4	CLKSEL[1:0]	Input System Clock Frequency	b5 b4 0 0: Do not set this value. 0 1: Do not set this value. 1 0: 20 MHz 1 1: 24 MHz	R/W
b7, b6	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b9, b8	REPSEL[1:0]	Terminating Resistance Adjustment	b9 b8 0 0: Disable the terminating resistance adjustment 0 1: Enable the terminating resistance adjustment 1 0: Do not set this value. 1 1: Do not set this value.	R/W
b14 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b15	HSEB	Classic-Only Mode *2	0: Classic-only mode is disabled. 1: Classic-only mode is enabled.	R/W

Note 1. Since the PLLRESET bit is 1 after a reset, changing the setting after release from the reset state is not required because of PLL reset control. Setting the PLLRESET bit to 1 after setting it to 0 may cause the USBA operate erroneously.
Note 2. Do not set the HSEB bit to 0 after setting it to 1. Doing so may cause the USBA operate erroneously.

CLKSEL[1:0] Bits (Input System Clock Frequency)

These bits select the frequency of the PLL clock source (USBMCLK) necessary for generating the USBA PHY clock. The USBA PHY clock is generated by the PLL embedded in the USB-PHY. The CLKSEL[1:0] bits set the frequency of the clock source (USBMCLK) supplied to the PLL. The USBMCLK is input from the EXTAL pin, but the clock specification must follow the clock standard in the USB 2.0. In classic-only mode, the PLL in the USB-PHY is stopped, so writing to the CLKSEL[1:0] bits is invalid (refer to the HSEB bit description for details).
Refer to section 39.3.3, Supplying Clocks for details on setting clocks.

REPSEL[1:0] Bits (Terminating Resistance Adjustment)

These bits automatically adjust the terminating resistance. Before operating the USBA, set the REPSEL[1:0] bits to 01b to enable the automatic adjustment function for the terminating resistance. Note that the automatic adjustment function for the terminating resistance is stopped when the HSEB bit is 1 (classic-only mode is enabled).

HSEB Bit (Classic-Only Mode)

This bit selects the method to supply the PHY clock.
When the HSEB bit is set to 0, the clock generated by the PLL embedded in the USB-PHY is used as the PHY clock.
When the HSEB bit is set to 1, the clock generated by the clock generator is used as the PHY clock. When the HSEB bit is set to 1, the UCLK must be set to 48 MHz, and PCLKB must be set to 60 MHz. Refer to section 9, Clock Generation Circuit for details on supplying the clock. Refer to section 39.3.3 for details on setting clocks.

39.2.17 Interrupt Status Register 0 (INTSTS0)

Address: 000D 0440h

	b15	b14	b13	b12	b11	b10	b9	b8
	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY
Value after reset:	0	0	0	0/1 *1	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	VBSTS		DVSQ[2:0]		VALID		CTSQ[2:0]	
Value after reset:	x	0	0	0/1 *1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CTSQ[2:0]	Control Transfer Stage Flag *2	b2 b0 0 0 0: Idle state or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error 1 1 1: Reserved	R
b3	VALID	USB Request Reception Flag *2	0: Setup packet is not received 1: Setup packet is received	R/(W)
b6 to b4	DVSQ[2:0]	Device State Flag *2	b6 b4 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state 1 0 0: Suspended state 1 0 1: Suspended state 1 1 0: Suspended state 1 1 1: Suspended state	R
b7	VBSTS	VBUS Input Status Flag	0: The USBA_VBUS pin is low 1: The USBA_VBUS pin is high	R
b8	BRDY	BRDY Interrupt Status Flag	0: BRDY interrupt not requested 1: BRDY interrupt requested	R
b9	NRDY	NRDY Interrupt Status Flag	0: NRDY interrupt not requested 1: NRDY interrupt requested	R
b10	BEMP	BEMP Interrupt Status Flag	0: BEMP interrupt not requested 1: BEMP interrupt requested	R
b11	CTRT	Control Transfer Stage Transition Interrupt Status Flag *3	0: Control transfer stage transition interrupt not requested 1: Control transfer stage transition interrupt requested	R/(W) *4
b12	DVST	Device State Transition Interrupt Status Flag *3	0: Device state transition interrupt not requested 1: Device state transition interrupt requested	R/(W) *4
b13	SOFR	Frame Number Refresh Interrupt Status Flag	0: Frame number refresh interrupt not requested 1: Frame number refresh interrupt requested	R/(W) *4
b14	RESM	Resume Interrupt Status Flag *3	0: Resume interrupt not requested 1: Resume interrupt requested	R/(W) *4
b15	VBINT	VBUS Interrupt Status Flag *5	0: VBUS interrupt is not requested on detecting a change in the USBA_VBUS pin. 1: VBUS interrupt is requested on detecting a change in the USBA_VBUS pin.	R/(W) *4

Note 1. The value is 0 when the MCU is reset, and 1 when a USB bus reset occurs.
Note 2. The CTSQ[2:0], VALID, and DVSQ[2:0] flags are only valid in function controller operation.
Note 3. The CTRT, DVST, and RESM flags are only changed in function controller operation. Set the corresponding interrupt enable bits to 0 (disabled) in host controller operation.
Note 4. To clear the CTRT, DVST, SOFR, RESM, or VBINT flags, write 0 only to the flags to be cleared, and write 1 to the other flags. Do not write 0 to the status flags indicating 0.
Note 5. The USBA detects a change in the status indicated by the VBINT flag even while the clock supply is stopped (while the LPSTS.SUSPENDM bit is 0), and requests the USBAR interrupt when the corresponding interrupt enable bit is 1. Clear these flags after supplying the PHY clock.

BRDY Flag (BRDY Interrupt Status Flag)

This flag indicates the state of the BRDY interrupt. Refer to section 39.2.12 for conditions of the BRDY flag becoming 1. The BRDY flag becomes 0 when 0 is written to the BRDYSTS.PIPEBRDY[9:0] flags for all pipes for which the BRDYENB.PIPEBRDYE[9:0] bits are 1 (interrupt request output enabled). The BRDY flag does not become 0 even when set to 0 by software.

NRDY Flag (NRDY Interrupt Status Flag)

This flag indicates the state of the NRDY interrupt. Refer to section 39.2.13 for conditions of the NRDY flag becoming 1. The NRDY flag becomes 0 when 0 is written to the NRDYSTS.PIPENRDY[9:0] flags for all pipes for which the NRDYENB.PIPENRDYE[9:0] bits are 1 (interrupt request output enabled). The NRDY flag does not become 0 even when set to 0 by software.

BEMP Flag (BEMP Interrupt Status Flag)

This flag indicates the state of the BEMP interrupt. Refer to section 39.2.14 for conditions of the BEMP flag becoming 1. The BEMP flag becomes 0 when 0 is written to the BEMPSTS.PIPEBEMP[9:0] flags for all pipes for which the BEMPENB.PIPEBEMPE[9:0] bits are 1 (interrupt request output enabled). The BEMP flag does not become 0 even when set to 0 by software.

CTRT Flag (Control Transfer Stage Transition Interrupt Status Flag)

In function controller operation, the USBA updates the value of the CTSQ[2:0] flags and sets the CTRT flag to 1 on detecting a transition in the control transfer stage. When a control transfer stage transition interrupt is generated, set the CTRT flag to 0 before the USBA detects the next control transfer stage transition. Values read from the CTRT flag are invalid in host controller operation.

DVST Flag (Device State Transition Interrupt Status Flag)

In function controller operation, the USBA updates the value of the DVSQ[2:0] flags and sets the DVST flag to 1 when a change in the device state is detected. When a device state transition interrupt is generated, set the DVST flag to 0 before the USBA detects the next device state transition. Values read from the DVST flag are invalid in host controller operation.

SOFR Flag (Frame Number Refresh Interrupt Status Flag)

- In host controller operation
  When the DVSTCTR0.UACT bit is 1, the SOFR flag becomes 1 (a frame number refresh interrupt is detected every millisecond) when the frame number is updated.
- In function controller operation
  The USBA sets the SOFR flag to 1 (a frame number refresh interrupt is detected every millisecond) when the frame number is updated. The USBA can detect a frame number refresh interrupt using the SOF recovery function even when a corrupted SOF packet is received from the USB host. Refer to section 39.3.13 for details on the SOF recovery function.

RESM Flag (Resume Interrupt Status Flag)

In function controller operation, the RESM flag becomes 1 when the USBA detects the resume signal in the suspended state (DVSQ[2:0] flags are 1xxb). Values read from the RESM flag are invalid in host controller operation.

VBINT Flag (VBUS Interrupt Status Flag)

The VBINT flag becomes 1 when a level change (high to low or low to high) is detected in the USBA_VBUS pin input signal. The USBA sets the VBSTS flag to indicate the USBA_VBUS pin input level. When the VBUS interrupt request is generated, eliminate chattering by reading the VBSTS flag several times by software and checking that the values read are the same.

39.2.18 Interrupt Status Register 1 (INTSTS1)

Address: 000D 0442h

	b15	b14	b13	b12	b11	b10	b9	b8
	OVRCR	BCHG	—	DTCH	ATTCH	—	L1RSMEND	LPMEND
Value after reset:	0	0	x	0	0	x	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	EOFERR	SIGN	SACK	—	—	—	PDDETINT
Value after reset:	x	0	0	0	x	x	x	0

Bit	Symbol	Bit Name	Description	R/W
b0	PDDETINT	Portable Device Detection Interrupt Status Flag *1	0: Portable Device detection interrupt not requested 1: Portable Device detection interrupt requested	R/(W) *2
b3 to b1	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b4	SACK	Setup Transaction Normal Response Interrupt Status Flag *3	0: SACK interrupt not requested 1: SACK interrupt requested	R/(W) *2
b5	SIGN	Setup Transaction Error Interrupt Status Flag *3	0: SIGN interrupt not requested 1: SIGN interrupt requested	R/(W) *2
b6	EOFERR	End-of-Frame Error Detection Interrupt Status Flag *3	0: EOFERR interrupt not requested 1: EOFERR interrupt requested	R/(W) *2
b7	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b8	LPMEND	LPM Transaction End Interrupt Status Flag *3	0: LPMEND interrupt not requested 1: LPMEND interrupt requested	R/(W) *2
b9	L1RSMEND	L1 Resume End Interrupt Status Flag *3	0: L1RSMEND interrupt not requested 1: L1RSMEND interrupt requested	R/(W) *2
b10	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b11	ATTCH	Device Attach Detection Interrupt Status Flag *3	0: ATTCH interrupt not requested 1: ATTCH interrupt requested	R/(W) *2
b12	DTCH	Device Detach Detection Interrupt Status Flag *3	0: DTCH interrupt not requested 1: DTCH interrupt requested	R/(W) *2
b13	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b14	BCHG	Bus Change Interrupt Status Flag *3	0: BCHG interrupt not requested 1: BCHG interrupt requested	R/(W) *2
b15	OVRCR	Overcurrent Change Interrupt Status Flag *1, *3	0: OVRCR interrupt not requested 1: OVRCR interrupt requested	R/(W) *2

Note 1. If a change in the status indicated by the PDDETINT or OVRCR flag is detected even while the clock is stopped (when the LPSTS.SUSPENDM bit is 0), and if the corresponding interrupt enable bit is 1, the USBAR interrupt is requested. Supply the PHY clock before setting these flags to 0. Interrupts indicated by flags other than the PDDETINT and OVRCR flags are not detected while the PHY clock is stopped (when the LPSTS.SUSPENDM bit is 0).

Note 2. To clear the flags in the INTSTS1 register, write 0 only to the flags to be cleared; write 1 to the other flags.

Note 3. Only enable interrupts other than the Portable Device detection interrupt in host controller operation.

PDDETINT Flag (Portable Device Detection Interrupt Status Flag)

In host controller operation, this flag indicates the status of the Portable Device detection interrupt.

The PDDETINT flag becomes 1 when the USBA detects a Portable Device. The USBA indicates the D+ line status in the BCCTRL.PDDETSTS flag. When the Portable Device detection interrupt is generated, eliminate chattering by reading the BCCTRL.PDDETSTS flag several times by software and checking that the values read are the same.

SACK Flag (Setup Transaction Normal Response Interrupt Status Flag)

In host controller operation, this flag indicates the status of the setup transaction normal response interrupt (SACK interrupt). When the USBA issues a setup transaction, if an ACK response is received from a peripheral device, the SACK flag becomes 1. At this time, if the INTENB1.SACKE bit is 1, the USBA requests the SACK interrupt.
In function controller operation, the SACK flag read value is invalid.

SIGN Flag (Setup Transaction Error Interrupt Status Flag)

In host controller operation, this flag indicates the status of the setup transaction error interrupt (SIGN interrupt). When the USBA issues a setup transaction, if an ACK response is not returned from a peripheral device three consecutive times, the SIGN flag becomes 1. At this time, if the INTENB1.SIGNE bit is 1, the USBA requests the SIGN interrupt. If any of the following conditions occurs three times consecutively, the USBA detects the setup transaction error

- A timeout is detected by the USBA when there is no response from a peripheral device
- An ACK packet is corrupted.
- A handshake other than ACK (i.e. NAK, NYET, or STALL) is received

In function controller operation, the SIGN flag read value is invalid.

EOFERR Flag (End-of-Frame Error Detection Interrupt Status Flag)

In host controller operation, this flag indicates the status of the EOF error detection interrupt (EOFERR interrupt). When communication is detected as not being complete at the EOF2 timing prescribed in USB 2.0, the EOFERR flag becomes 1. At this time, if the INTENB1.EOFERRE bit is 1, the USBA requests the EOFERR interrupt. After detecting the EOF error, regardless of the INTENB1.EOFERRE bit setting, the USBA sets the DVSTCTR0.UACT bit to 0 and enters the Idle state. controls hardware as described below. Use software to end all communication currently being performed on pipes, and repeat enumeration.
In function controller operation, the EOFERR flag read value is invalid.

LPMEND Flag (LPM Transaction End Interrupt Status Flag)

When the HL1CTRL1.L1REQ bit is set to 1, the USBA transmits an LPM token. When the LPM transaction is ended because a response from the peripheral device or a timeout is detected, the LPMEND flag becomes 1.
In function controller operation, the LPMEND flag read value is invalid.

L1RSMEND Flag (L1 Resume End Interrupt Status Flag)

When performing processing to exit the L1 state, the L1RSMEND flag becomes 1 when the processing is completed.
In function controller operation, the L1RSMEND flag read value is invalid.

ATTCH Flag (Device Attach Detection Interrupt Status Flag)

In host controller operation, this flag indicates the status of the device attach detection interrupt (ATTCH interrupt). When the USBA detects J state or K state of the full-speed or low-speed signal level for 2.5 µs on the port, the USBA detects that a device has been attached, and sets the ATTCH flag to 1. At this time, if the INTENB1.ATTCHE bit is 1, the USBA requests the USBAR interrupt.
The conditions for the USBA to detect a device being attached are as follows:

- The K state, SE0, or SE1 changes to the J state, and the J state continues for 2.5 µs
- The J state, SE0, or SE1 changes to the K state, and the K state continues for 2.5 µs

In function controller operation, the ATTCH flag read value is invalid.

DTCH Flag (Device Detach Detection Interrupt Status Flag)

In host controller operation, this flag indicates the status of the device detach detection interrupt (DTCH interrupt).
When a USB bus device disconnect is detected, the USBA detects device disconnection, and the DTCH flag becomes 1. At this time, if the INTENB1.DTCHE bit is set to 1, the USBA requests the USBAR interrupt.
The USBA detects bus disconnection based on USB 2.0.
After device disconnect is detected, regardless of the INTENB1.DTCHE bit setting, the USBA sets the DVSTCTR0.UACT bit to 0 and enters the Idle state. Use software to end all communication currently being performed on pipes, and transition to the wait for attach (ATTCH interrupt generated) state.
In function controller operation, the DTCH flag read value is invalid.

BCHG Flag (Bus Change Interrupt Status Flag)

This flag indicates the status of the bus change interrupt (BCHG interrupt).
When a status change to full-speed or low-speed occurs on the USB port (when the J state, K state, or SE0 changes to the J state, K state, or SE0), the USBA detects a change in the bus status, and the BCHG flag becomes 1. At this time, if the INTENB1.BCHGE bit is 1, the USBA requests the USBAR interrupt.
The SYSSTS0.LNST[1:0] flags indicate the current state of the USB port. When the BCHG interrupt is generated, in order to eliminate chattering, read the LNST[1:0] flags several times by software and confirm that the values read are the same.
In function controller operation, the BCHG flag read value is invalid.

OVRCR Flag (Overcurrent Change Interrupt Status Flag)

This flag indicates the status change interrupt (OVRCR interrupt) for the USBA_OVRCURA pin or USBA_OVRCURB pin.
If there is a change in the input of either the USBA_OVRCURA pin or USBA_OVRCURB pin (from high to low or low to high), the USBA detects a change in the overcurrent pin, and the OVRCR flag becomes 1. At this time, if the INTENB1.OVRCRE bit is 1, the USBA requests the USBAR interrupt.
The SYSSTS0.OVCMON[1:0] flags to indicate the current status of the USBA_OVRCURA and USBA_OVRCURB pins.
In host controller operation, if an overcurrent notification signal (from the power supply IC which supplies VBUS to the peripheral devices) is connected to the USBA_OVRCURA or USBA_OVRCURB pin, software can detect an occurrence of overcurrent.

39.2.19 BRDY Interrupt Status Register (BRDYSTS)

Address: 000D 0446h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPEBRDY[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	PIPEBRDY[9:0]	Pipe 9 to Pipe 0 BRDY Interrupt Status Flags *1	0: Interrupt not generated 1: Interrupt generated	R/(W) *2
b15 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. Each bit corresponds to a pipe number (e.g. pipe 9 corresponds to the PIPEBRDY[9] bit, pipe 8 corresponds to the PIPEBRDY[8] bit, etc.).

Note 2. When the SOFCFG.BRDYM bit is 0, to clear the status indicated by the PIPEBRDY[9:0] flags, write 0 only to the flags to be cleared; write 1 to the other flags.
When the SOFCFG.BRDYM bit is set to 0, clear the BRDY interrupt before accessing the FIFO buffer.

PIPEBRDY[9:0] Flags (Pipe 9 to Pipe 0 BRDY Interrupt Status Flags)

When the BRDY interrupt is detected, the USBA sets the bits corresponding to the PIPEBRDY[9:0] flags to 1. Refer to section 39.3.6.1 for details on the BRDY interrupt.

39.2.20 NRDY Interrupt Status Register (NRDYSTS)

Address: 000D 0448h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPENRDY[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	PIPENRDY[9:0]	Pipe 9 to Pipe 0 NRDY Interrupt Status Flags *1	0: Interrupt not generated 1: Interrupt generated	R/(W) *2
b15 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. Each bit corresponds to a pipe number (e.g. Pipe 9 corresponds to the PIPENRDY[9] bit, pipe 8 corresponds to the PIPENRDY[8] bit, etc.).

Note 2. To clear the status indicated by the PIPENRDY[9:0] flags, write 0 only to the flags to be cleared; write 1 to the other flags.

PIPENRDY[9:0] Flags (Pipe 9 to Pipe 0 NRDY Interrupt Status Flags)

If an NRDY interrupt source is detected while the PID[1:0] bits in a pipe control register are 01b (BUF response), the USBA module sets the bits corresponding to the PIPENRDY[9:0] flags to 1. Refer to section 39.3.6.2 for details on the NRDY interrupt.

39.2.21 BEMP Interrupt Status Register (BEMPSTS)

Address: 000D 044Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPEBEMP[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	PIPEBEMP[9:0]	Pipe 9 to Pipe 0 BEMP Interrupt Status Flags *1	0: Interrupt not generated 1: Interrupt generated (n = 0 to 9)	R/(W) *2
b15 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. Each bit corresponds to a pipe number (e.g. pipe 9 corresponds to the PIPEBEMP[9] bit, pipe 8 corresponds to the PIPEBEMP[8] bit, etc.).

Note 2. To clear the status indicated by the PIPEBEMP[9:0] flags, write 0 only to the flags to be cleared; write 1 to the other flags.

PIPEBEMP[9:0] Flags (Pipe 9 to Pipe 0 BEMP Interrupt Status Flags)

If a BEMP interrupt is detected while the PID[1:0] bits in a pipe control register are 01b (BUF response), the USBA module sets the bits corresponding to the PIPEBEMP[9:0] flags to 1. Refer to section 39.3.6.3 for details on the BEMP interrupt.

39.2.22 Frame Number Register (FRMNUM)

Address: 000D 044Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRN	CRCE	—	—	—	FRNM[10:0]										
Value after reset:	0	0	x	x	x	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b10 to b0	FRNM[10:0]	Frame Number Flags	Latest frame number	R
b13 to b11	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b14	CRCE	CRC Error Detection Status Flag	0: No error occurred 1: Error occurred	R/(W) *2
b15	OVRN	Overrun/Underrun Detection Status Flag *1	0: No error occurred 1: Error occurred	R/(W) *2

Note 1. The OVRN flag is used for debugging. Design the timing so that no overrun or underrun occurs in the system.
Note 2. When setting this flag to 0, write 0 only to the flags to be cleared; write 1 to the other flags.

FRNM[10:0] Flags (Frame Number Flags)

An SOF packet is issued or received every millisecond. Each time an SOF packet is issued or received, the USBA rewrites these flags to indicate the latest frame number.

CRCE Flag (CRC Error Detection Status Flag)

This flag becomes 1 when a CRC error or bit stuffing error occurs during isochronous transfer. Write 0 to the CRCE flag to clear it. When writing 0 to the CRCE flag, write 1 to the other flags in the FRMNUM register. When a CRC error is detected, the USBA generates an internal NRDY interrupt request.

OVRN Flag (Overrun/Underrun Detection Status Flag)

This flag becomes 1 when an overrun or underrun error occurs during isochronous transfer. Write 0 to the OVRN flag to clear it. When writing 0 to the OVRN flag, write 1 to the other flags in the FRMNUM register.

(1) In host controller operation

The USBA sets the OVRN flag to 1 when either of the following conditions is satisfied.

- For pipes set to isochronous transfer for transmitting, the timing to issue an OUT token comes before all the transmit data has been written to the FIFO buffer.
- For pipes set to isochronous transfer for receiving, the timing to issue an IN token comes while no FIFO buffer plane is empty.

(2) In function controller operation

The USBA sets the OVRN flag to 1 when any of the following conditions is satisfied.

- For pipes set to isochronous transfer for transmitting, the IN token is received before all the transmit data has been written to the FIFO buffer.
- For pipes set to isochronous transfer for receiving, the OUT token is received while no FIFO buffer plane is empty.

39.2.23 USB Address Register (USBADDR)

Address: 000D 0450h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	USBADDR[6:0]						
Value after reset:	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	USBADDR[6:0]	USB Address Flags	In function controller operation, these flags indicate the USB address assigned by the host when the USBA processed the SetAddress() request successfully.	R
b15 to b7	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

USBADDR[6:0] Flags (USB Address Flags)

In function controller operation, these flags indicate the USB address received in the USBADDR[6:0] flags when the USBA processed the SetAddress() request successfully. The USBADDR[6:0] flags indicate 00h when a USB bus reset is detected.

In host controller operation, the USBA does not use the USBADDR[6:0] flags.

39.2.24 USB Request Type Register (USBREQ)

Address: 000D 0454h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BREQUEST[7:0]								BMREQUESTTYPE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	BMREQUESTTYPE[7:0]	Request Type	USB request *bmRequestType* value	R/W *1
b15 to b8	BREQUEST[7:0]	Request	USB request *bRequest* value	R/W *1

Note 1. In function controller operation, these bits are readable but writing to them is invalid. In host controller operation, these bits are both readable and writable.

BMREQUESTTYPE[7:0] Bits (Request Type)

These bits store the *bmRequestType* value of USB requests.

- In host controller operation
  Set these bits to the value of the USB request data in setup transactions to be transmitted. Do not modify these bits when the DCPCTR.SUREQ bit is 1.
- In function controller operation
  These bits indicate the value of the USB request data in setup transactions to be received. Writing to these bits is invalid.

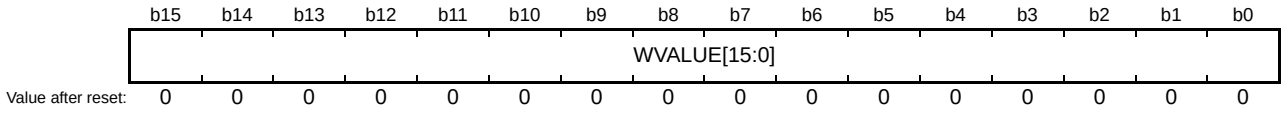
BREQUEST[7:0] Bits (Request)

These bits store the *bRequest* value of USB requests.

- In host controller operation
  Set these bits to the value of the USB request data in setup transactions to be transmitted. Do not modify these bits when the DCPCTR.SUREQ bit is 1.
- In function controller operation
  These bits indicate the value of the USB request data in setup transactions to be received. Writing to these bits is invalid.

39.2.25 USB Request Value Register (USBVAL)

Address: 000D 0456h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	WVALUE[15:0]	Value	Value of the USB request *wValue*	R/W *1

Note 1. In function controller operation, these bits are readable but writing to them is invalid. In host controller operation, these bits are both readable and writable.

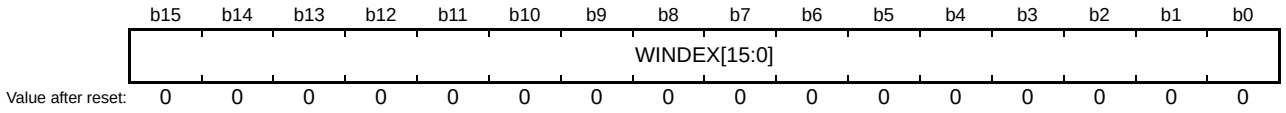
WVALUE[15:0] Bits (Value)

These bits store the *wValue* value of USB requests.

- In host controller operation
  Set the *wValue* value of USB requests in setup transactions to be transmitted. Do not modify these bits when the DCPCTR.SUREQ bit is 1.
- In function controller operation
  These bits indicate the *wValue* value of USB requests in setup transactions to be received. Writing to these bits is invalid.

39.2.26 USB Request Index Register (USBINDX)

Address: 000D 0458h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	WINDEX[15:0]	Index	Value of the USB request *wIndex*	R/W *1

Note 1. In function controller operation, these bits are readable but writing to them is invalid. In host controller operation, these bits are both readable and writable.

WINDEX[15:0] Bits (Index)

These bits store the *wIndex* value of USB requests.

- In host controller operation
  Set the *wIndex* value of USB requests in setup transactions to be transmitted. Do not modify these bits when the DCPCTR.SUREQ bit is 1.
- In function controller operation
  These bits indicate the *wIndex* value of USB requests in setup transactions to be received. Writing to these bits is invalid.

39.2.27 USB Request Length Register (USBLENG)

Address: 000D 045Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WLENGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	WLENGTH[15:0]	Length	Value of the USB request *wLength*	R/W *1

Note 1. In function controller operation, these bits are readable but writing to them is invalid. In host controller operation, these bits are both readable and writable.

WLENGTH[15:0] Bits (Length)

These bits store the *wLength* value of USB requests.

- In host controller operation
  Set the *wLength* value of USB requests in setup transactions to be transmitted. Do not modify these bits when the DCPCTR.SUREQ bit is 1.
- In function controller operation
  These bits indicate the *wLength* value of USB requests in setup transactions to be received. Writing to these bits is invalid.

39.2.28 Default Control Pipe Configuration Register (DCPCFG)

Address: 000D 045Ch

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	—	CNTMD
Value after reset:	x	x	x	x	x	x	x	0

	b7	b6	b5	b4	b3	b2	b1	b0
	SHTNAK	—	—	DIR	—	—	—	—
	0	x	x	0	x	x	x	x

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b4	DIR	Transfer Direction *1	0: Receive data 1: Transmit data	R/W
b6, b5	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer *1	0: The pipe is continued at the end of transfer. 1: The pipe is disabled at the end of transfer.	R/W
b8	CNTMD	Continuous Transfer Mode	0: Discontinuous transfer mode 1: Continuous transfer mode	R/W
b15 to b9	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

Note 1. Only change the settings of bits in the DCPCFG register when the DCPCTR.PID[1:0] bits are 00b (NAK response). Only change the DCPCTR.PID[1:0] bits from 01b (BUF response) to 00b (NAK response) after confirming that the value of the DCPCTR.PBUSY flag is 0. However, the PBUSY flag does not need to be checked by software if the USBA changes the PID[1:0] bits to 00b (NAK response).

DIR Bit (Transfer Direction)

In host controller operation, this bit sets the transfer direction of the data stage and status stage for control transfer.
In function controller operation, set this bit to 0.

SHTNAK Bit (Pipe Disabled at End of Transfer)

When the control transfer direction is receiving, this bit specifies whether the DCPCTR.PID[1:0] bits are changed from 00b (NAK response) when transfer is complete.
The SHTNAK bit is valid when the default control pipe transfer direction is receiving.
When the SHTNAK bit is 1, the DCPCTR.PID[1:0] bits become 00b (NAK response) when the USBA transfer is complete. The USBA determines that transfer is complete when short packet data (including the zero-length packet) is received successfully.

CNTMD Bit (Continuous Transfer Mode)

This bit specifies whether the default control pipe communicates in continuous transfer mode or not.

39.2.29 Default Control Pipe Maximum Packet Size Register (DCPMAXP)

Address: 000D 045Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEVSEL[3:0]				—	—	—	—	—	MXPS[6:0]						
Value after reset:	0	0	0	0	x	x	x	x	x	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	MXPS[6:0]	Maximum Packet Size *1	These bits specify the maximum data payload (maximum packet size) for the default control pipe.	R/W
b11 to b7	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b15 to b12	DEVSEL[3:0]	Device Select *2	b15 b12 0 0 0 0: Address 0000b 0 0 0 1: Address 0001b 0 0 1 0: Address 0010b 0 0 1 1: Address 0011b 0 1 0 0: Address 0100b 0 1 0 1: Address 0101b 0 1 1 0: Address 0110b 0 1 1 1: Address 0111b 1 0 0 0: Address 1000b 1 0 0 1: Address 1001b 1 0 1 0: Address 1010b Only set the values listed above.	R/W

Note 1. Only change the MXPS[6:0] bit setting when the DCPCTR.PID[1:0] bits are 00b (NAK response), and the CFIFOSEL.CURPIPE[3:0] bits are 0000b. When changing the setting after changing the DCPCTR.PID[1:0] bits from 01b (BUF response) to 00b (NAK response), confirm the DCPCTR.PBUSY flag is 0 before modifying these bits. However, when the USBA changes the DCPCTR.PID[1:0] bits from 01b (BUF response) to 00b (NAK response), the PBUSY flag does not need to be checked by software. In addition, after the MXPS[6:0] bit setting is changed, set the CFIFOSEL.BCLR bit to 1 to perform buffer clear processing.

Note 2. Only change the DEVSEL[3:0] bit setting when the DCPCTR.PID[1:0] bits are 00b (NAK response), and the DCPCTR.SUREQ bit is 0. Only change the DCPCTR.PID[1:0] bits of the default control pipe from 01b (BUF response) to 00b (NAK response) after confirming the value of the DCPCTR.PBUSY flag is 0. However, the PBUSY flag does not need to be checked by software if the USBA changes the DCPCTR.PID[1:0] bits to 00b (NAK response).

MXPS[6:0] Bits (Maximum Packet Size)

Set the default control pipe's maximum data payload (maximum packet size) to the MXPS[6:0] bits. The initial value is 40h (64 bytes).

The value set to the MXPS[6:0] bits must be compliant with USB 2.0.

When the MXPS[6:0] bits are set to 00h, do not write to the FIFO buffer and do not set the DCPCTR.PID[1:0] bits to 01b (BUF response).

DEVSEL[3:0] Bits (Device Select)

In host controller operation, these bits specify the peripheral device address of the communication target for control transfer.

Set the DEVADDm register corresponding to the value set in the DEVSEL[3:0] bits, and then set the DEVSEL[3:0] bits (m = 0 to A). For example, to set the DEVSEL[3:0] bits to 0010b, set the address in the DEVADD2 register.

In function controller operation, set the DEVSEL[3:0] bits to 0000b.

39.2.30 Default Control Pipe Control Register (DCPCTR)

Address: 000D 0460h

	b15	b14	b13	b12	b11	b10	b9	b8
	BSTS	SUREQ	—	—	SUREQCLR	—	—	SQCLR
Value after reset:	0	0	0	0	x	x	x	0

	b7	b6	b5	b4	b3	b2	b1	b0
	SQSET	SQMON	PBUSY	—	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	x	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the FIFO buffer state) 1 0: STALL response 1 1: STALL response	R/W
b2	CCPL	Control Transfer End Enable	0: Completion of control transfer is disabled. 1: Completion of control transfer is enabled.	R/W
b4, b3	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b5	PBUSY	Pipe Busy Flag	0: The default control pipe is not used for the USB bus. 1: The default control pipe is in use for the USB bus.	R
b6	SQMON	Sequence Toggle Bit Monitor Flag	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set *1	This bit allows setting for the expected value of the sequence toggle bit of the next transaction for data 1 in default control pipe transfer. Writing 0 has no effect. This bit is read as 0.	R/W
b8	SQCLR	Sequence Toggle Bit Clear *1	This bit allows setting for the expected value of the sequence toggle bit of the next transaction for data 0 in default control pipe transfer. Writing 0 has no effect. This bit is read as 0.	R/W
b10, b9	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b11	SUREQCLR	SUREQ Bit Clear	In host controller operation, the SUREQ bit becomes 0 when this bit is set to 1. Writing 0 has no effect. This bit is read as 0.	R/W
b12	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R
b13	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b14	SUREQ	SETUP Token Transmission	In host controller operation, setting this bit to 1 transmits the setup packet. Writing 0 has no effect.	R/W
b15	BSTS	Buffer Status Flag	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only write 1 to bits SQSET and SQCLR while the DCPCTR.PID[1:0] bits are 00b (NAK response). Only change the DCPCTR.PID[1:0] bits from 01b (BUF response) to 00b (NAK response) after confirming that the value of the DCPCTR.PBUSY flag is 0. However, the PBUSY flag does not need to be checked by software if the USBA changes the DCPCTR.PID[1:0] bits to 00b (NAK response).

PID[1:0] Bits (Response PID)

The DCPCTR.PID[1:0] bits control the USB response for control transfer.

(1) In host controller operation

Perform the following to modify the DCPCTR.PID[1:0] bits from 00b (NAK response) to 01b (BUF response).

- When transmitting
  Write all the transmit data to the FIFO buffer while the DVSTCTR0.UACT bit is 1 and the DCPCTR.PID[1:0] bits are 00b (NAK response), and then set the DCPCTR.PID[1:0] bits to 01b (BUF response). After the DCPCTR.PID[1:0] bits have been set to 01b (BUF response), the USBA executes the OUT transaction.
- When receiving
  Confirm that the FIFO buffer is empty (or empty the buffer) while the DVSTCTR0.UACT bit is 1 and the DCPCTR.PID[1:0] bits are 00b (NAK response), and then set the DCPCTR.PID[1:0] bits to 01b (BUF response). After the DCPCTR.PID[1:0] bits have been set to 01b (BUF response), the USBA executes the IN transaction.

The USBA modifies the value of the DCPCTR.PID[1:0] bits in the following cases.

- When the DCPCTR.PID[1:0] bits are set to 01b (BUF response) and the USBA has received data exceeding the maximum packet size, the USBA sets the DCPCTR.PID[1:0] bits to 11b (STALL response).
- When a reception error such as CRC error is detected three times in succession, the USBA sets the DCPCTR.PID[1:0] bits to 00b (NAK response).
- When a STALL handshake is received, the USBA sets the DCPCTR.PID[1:0] bits to 11b (STALL response).

(2) In function controller operation

The USBA modifies the value of the DCPCTR.PID[1:0] bits in the following cases.

- When a setup packet is received, the USBA modifies the DCPCTR.PID[1:0] bits to 00b (NAK response). At this time, the USBA sets the INTSTS0.VALID flag to 1, and the DCPCTR.PID[1:0] bits cannot be modified by software until the VALID flag is set to 0 by software.
- When the DCPCTR.PID[1:0] bits are set to 01b (BUF response) and the USBA receives data exceeding the maximum packet size, the USBA sets the DCPCTR.PID[1:0] bits to 11b (STALL response).
- When the USBA detects a control transfer sequence error, the DCPCTR.PID[1:0] bits are set to 10b or 11b (STALL response).
- When the USBA detects a USB bus reset, the DCPCTR.PID[1:0] bits are set to 00b (NAK response).

The USBA does not check the setting of the DCPCTR.PID[1:0] bits during the SetAddress() request processing (automatic processing).

CCPL Bit (Control Transfer End Enable)

In function controller operation, setting this bit to 1 enables completion of the status stage of the control transfer.
If the CCPL bit is set to 1 while the corresponding DCPCTR.PID[1:0] bits are 01b (BUF response), the USBA completes the status stage of the control transfer.
During a control read transfer, the USBA transmits an ACK handshake for the OUT transaction from the USB host.
During a control write or no-data control transfer, the USBA transmits a zero-length packet for the IN transaction from the USB host. However, when the SetAddress() request is detected, the USBA performs automatic responses from the SETUP stage until the status stage is completed regardless of the setting of the CCPL bit.
When a new setup packet is received, the USBA modifies the CCPL bit from 1 to 0.
When the INTSTS0.VALID flag is 1, the CCPL bit cannot be set to 1.
The CCPL bit is initialized by the USB bus reset.
In host controller operation, set the CCPL bit to 0.

PBUSY Flag (Pipe Busy Flag)

When the USBA starts a USB transaction using the default control pipe, the USBA changes the PBUSY flag from 0 to 1.
The USBA switches the PBUSY flag back to 0 at the end of the transaction.
After the DCPCTR.PID[1:0] bits are set to 00b (NAK response), the read value of the PBUSY flag indicates whether or not changes to pipe settings can proceed. Refer to section 39.3.7.1 for details.

SQMON Flag (Sequence Toggle Bit Monitor Flag)

This flag indicates the value of the sequence toggling bit for the next transaction in default control pipe transfer. When a transaction is processed successfully, the USBA toggles the SQMON flag. However, the SQMON flag is not toggled when data PID mismatch occurs during data reception. In function controller operation, the USBA sets the SQMON flag to 1 (setting the expected value for DATA1) when a setup packet is received successfully. In addition, in function controller operation, the USBA does not check the SQMON flag during IN or OUT transactions in the status stage, and the flag is not toggled even at the end of a successful processing.

SQSET Bit (Sequence Toggle Bit Set)

This bit sets the value of the sequence toggling bit to DATA1 for the next transaction in default control pipe transfer. Do not set bits SQCLR and SQSET to 1 at the same time.

SQCLR Bit (Sequence Toggle Bit Clear)

This bit sets the value of the sequence toggling bit to DATA0 for the next transaction in default control pipe transfer. Do not set bits SQCLR and SQSET to 1 at the same time.

SUREQCLR Bit (SUREQ Bit Clear)

In host controller operation, the SUREQ bit can be cleared by setting the SUREQCLR bit to 1. If transfer stops while the SUREQ bit is set to 1 in a SETUP transaction, set the SUREQCLR bit to 1 by software. This is not necessary at the end of a successful setup transaction because the USBA will automatically set the SUREQ bit to 0. Use the SUREQCLR bit to control the SUREQ bit when transfer stops while the DVSTCTR0.UACT bit is set to 0 or transfer is not proceeding because detaching of the port has been detected. Set the SUREQCLR bit to 0 in function controller operation.

SUREQ Bit (SETUP Token Transmission)

In host controller operation, a setup packet is transmitted when the SUREQ bit is set to 1. After the SETUP transaction processing is completed, the USBA generates the SACK interrupt or SIGN interrupt and sets the SUREQ bit to 0. When the SUREQCLR bit is set to 1, the SUREQ bit becomes 0.
Set the SUREQ bit to 1 after a USB request to be transmitted in the setup transaction is set in the DCPMAXP.DEVSEL[3:0] bits and registers USBREQ, USBVAL, USBINDX, and USBLENG. Before setting the SUREQ bit to 1, confirm that the DCPCTR.PID[1:0] bits are set to 00b (NAK response). In addition, after the SUREQ bit is set to 1, until the setup transaction ends (SUREQ bit is 1), do not modify the DCPMAXP.DEVSEL[3:0] bits or values of registers USBREQ, USBVAL, USBINDX, and USBLENG. Set the SUREQ bit to 1 only when issuing a SETUP token. In other cases, be sure to write 0 to this bit. Set the SUREQ bit to 0 in function controller operation.

BSTS Flag (Buffer Status Flag)

This flag indicates whether the default control pipe's FIFO buffer can be accessed.
The meaning of the BSTS flag varies depending on the setting of the CFIFOSEL.ISEL bit.

- When the ISEL bit is 0, the BSTS flag indicates whether receive data can be read.
- When the ISEL bit is 1, the BSTS flag indicates whether transmit data can be written.

39.2.31 Pipe Window Select Register (PIPESEL)

Address: 000D 0464h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PIPESEL[3:0]	Pipe Window Select	b3 b0 0 0 0 0: No pipe selected 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Only set the values listed above.	R/W
b15 to b4	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

Set pipes 1 to 9 using registers PIPESEL, PIPECFG, PIPEBUF, PIPEMAXP, PIPEPERI, PIPEnCTR (n = 1 to 9), PIPEnTRE (n = 1 to 5), and PIPEnTRN (n = 1 to 5). After selecting the pipe to be set by the PIPESEL register, each pipe setting is performed by the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers. In addition, the PIPEnCTR, PIPEnTRE, and PIPEnTRN registers can be set regardless of the pipe selection by the PIPESEL register.

PIPESEL[3:0] Bits (Pipe Window Select)

When the pipe number is specified in the PIPESEL[3:0] bits, the pipe information is indicated in the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers. Perform the pipe setting by modifying these registers.
When the PIPESEL[3:0] bits are set to 0000b, reading bits in the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers returns 0. Writing has no effect.

39.2.32 Pipe Configuration Register (PIPECFG)

Address: 000D 0468h

	b15	b14	b13	b12	b11	b10	b9	b8
	TYPE[1:0]		—	—	—	BFRE	DBLB	CNTMD
Value after reset:	0	0	x	x	x	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	SHTNAK	—	—	DIR	EPNUM[3:0]			
Value after reset:	0	x	x	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	EPNUM[3:0]	Endpoint Number *1	These bits specify the endpoint number for the selected pipe. 0000b indicates the pipe is not used.	R/W
b4	DIR	Transfer Direction *2, *3	0: Receive data 1: Transmit data	R/W
b6, b5	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer *1	0: The pipe is continued at the end of transfer. 1: The pipe is disabled at the end of transfer.	R/W
b8	CNTMD	Continuous Transfer Mode *2, *3	0: Discontinuous transfer mode 1: Continuous transfer mode	R/W
b9	DBLB	Double Buffer Mode *2, *3	0: Single buffer mode 1: Double buffer mode	R/W
b10	BFRE	BRDY Interrupt Operation Specification *2, *3	0: BRDY interrupt when data is transmitted or received 1: BRDY interrupt when all data has been read	R/W
b13 to b11	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b15, b14	TYPE[1:0]	Transfer Type *1	• For pipes 1 and 2 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Do not set this value. 1 1: Isochronous transfer • For pipes 3 to 5 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Do not set this value. 1 1: Do not set this value. • For pipes 6 to 9 b15 b14 0 0: Pipe not used 0 1: Do not set this value. 1 0: Interrupt transfer 1 1: Do not set this value.	R/W

Note 1. Only change the settings of the TYPE[1:0], SHTNAK, and EPNUM[3:0] bits while the PIPEnCTR.PBUSY flag is 0, and the PIPEnCTR.PID[1:0] bits are 00b (NAK response). Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY flag is 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBA changes the PID[1:0] bits to 00b (NAK response).

Note 2. Only change the setting of the BFRE, DBLB, CNTMD, and DIR bits while the PIPEnCTR.PBUSY flag is 0, and the PIPEnCTR.PID[1:0] bits are 00b (NAK response) with the pipe not specified in the CURPIPE[3:0] bits of a FIFO port select register. Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY flag is 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBA changes the PID[1:0] bits to 00b (NAK response).

Note 3. When the setting of the BFRE, DBLB, CNTMD, and DIR bits are modified after USB communications have been carried with the selected pipe, write 1 and 0 to the PIPEnCTR.ACLRM bit continuously by software and clear the FIFO buffer assigned to the selected pipe, in addition to the note described in Note 2.

EPNUM[3:0] Bits (Endpoint Number)

These bits specify the endpoint number for the selected pipe. Setting 0000b indicates the pipe not used. Set these bits so that the combination of the settings of the DIR and EPNUM[3:0] bits is different from the settings of other pipes. (The EPNUM[3:0] bits being set to 0000b can be set for all pipes.)

DIR Bit (Transfer Direction)

This bit specifies the transfer direction for the selected pipe. When the DIR bit is set to 0, the USBA uses the selected pipe to receive data. When the DIR bit is set to 1, the USBA uses the selected pipe to transmit data.

SHTNAK Bit (Pipe Disabled at End of Transfer)

This bit selects whether to modify the PIPEnCTR.PID[1:0] bits to 00b (NAK response) at the end of transfer when receiving direction is set for the selected pipe. This bit is valid when the default control pipe is in the receiving direction. When software processing sets this bit to 1 for a pipe in the receiving direction, the USBA modifies the PIPEnCTR.PID[1:0] bits corresponding to the selected pipe to 00b (NAK response) on determining the end of the transfer. The USBA determines that the transfer has ended from the following conditions.

- Short packet data (including a zero-length packet) is received successfully.
- The transaction counter is used and packets of the number specified by the transaction counter are received successfully.

CNTMD Bit (Continuous Transfer Mode)

This bit is valid when the selected pipe is pipe 1 to 5 of the bulk transfer type. According to the setting of this bit, the USBA determines the completion of transmission or reception for the FIFO buffer allocated to the selected pipe as shown in Table 39.7.

Table 39.7 Relationship Between the CNTMD Bit Setting and Methods for Determining Completion of FIFO Buffer Transmission/Reception

CNTMD Bit Setting	Method for Determining Readability and Transmittability
0	Conditions for FIFO buffer readable state when receiving direction is set (DIR bit is 0): The USBA has received one packet.
	Conditions for FIFO buffer transmittable state when transmitting direction is set (DIR bit is 1): When (1) or (2) is satisfied. (1) CPU (or DMAC/DTC) has written data of the maximum packet size to the FIFO buffer. (2) CPU (or DMAC/DTC) has written data of the short packet size (including 0 bytes) to the FIFO buffer and has set the BVAL bit in the FIFO port control register to 1.
1	Conditions for FIFO buffer readable state when receiving direction is set (DIR bit is 0): (1) The byte count of data received in the FIFO buffer allocated to the selected pipe is equal to the allocated byte count ((BUFSIZE + 1) × 64). (2) While there is no data stored in the FIFO buffer allocated to the selected pipe, the USBA has received a short packet except for a zero-length packet. (3) The USBA has received a zero-length packet when data is already contained in the FIFO buffer allocated to the selected pipe. (4) The number of packets specified by the transaction counter set for the selected pipe have been received.
	Conditions for FIFO buffer transmittable state when transmitting direction is set (DIR bit is 1): When (1) or (2) is satisfied. (1) CPU (or DMAC/DTC) has written data that is equal to the size of the FIFO buffer allocated to the selected pipe. (2) CPU (or DMAC/DTC) has written data of size smaller than the size of the FIFO buffer allocated to the selected pipe (including 0 bytes) and has set the BVAL bit in the FIFO port control register to 1.

DBLB Bit (Double Buffer Mode)

This bit specifies single buffer or double buffer for the FIFO buffer used by the selected pipe. This bit is valid when the selected pipe is pipe 1 to 5. When this bit is set to 1, the USBA allocates twice the FIFO buffer size specified by the PIPEBUF.BUFSIZE[4:0] bits for the selected pipe. The FIFO buffer size that the USBA allocates to the selected pipe is (PIPEBUF.BUFSIZE[4:0] + 1) × 64 × (DBLB + 1) [bytes].

BFRE Bit (BRDY Interrupt Operation Specification)

This bit specifies the timing for transmitting the BRDY interrupt. When this bit is set to 1 and the selected pipe is used in the receiving direction, the USBA detects the end of the transfer and issues a BRDY interrupt upon reading the packet. When a BRDY interrupt is generated with this setting, set the BCLR bit in a FIFO port control register to 1. Until the BCLR bit is set to 1, the FIFO buffer allocated to the selected pipe does not become the receivable state. When software processing sets this bit to 1 and the selected pipe is used in the transmitting direction, the USBA generates no BRDY interrupt. Refer to section 39.3.6.1 for details.

TYPE[1:0] Bits (Transfer Type)

These bits specify the transfer type of the pipe specified by the PIPESEL.PIPESEL[3:0] bits. Before modifying the PIPEnCTR.PID[1:0] bits of the selected pipe to 01b (BUF response) (before starting the USB communication with the selected pipe), set the TYPE[1:0] bits to a value other than 00b.

39.2.33 Pipe Buffer Register (PIPEBUF)

Address: 000D 046Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	BUFSIZE[4:0]					—	—	BUFNMB[7:0]							
Value after reset:	x	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description			R/W
b7 to b0	BUFNMB[7:0]	Buffer Number *1	These bits specify the start block of the FIFO buffer allocated to the selected pipe (04h to 87h).			R/W
b9, b8	—	Reserved	These bits are undefined when read. Set them to 0 when writing.			R/W
b14 to b10	BUFSIZE[4:0]	Buffer Size *1	00h: 64 bytes 01h: 128 bytes 02h: 192 bytes 03h: 256 bytes 04h: 320 bytes 05h: 384 bytes 06h: 448 bytes 07h: 512 bytes 08h: 576 bytes 09h: 640 bytes 0Ah: 704 bytes	0Bh: 768 bytes 0Ch: 832 bytes 0Dh: 896 bytes 0Eh: 960 bytes 0Fh: 1,024 bytes 10h: 1,088 bytes 11h: 1,152 bytes 12h: 1,216 bytes 13h: 1,280 bytes 14h: 1,344 bytes 15h: 1,408 bytes	16h: 1,472 bytes 17h: 1,536 bytes 18h: 1,600 bytes 19h: 1,664 bytes 1Ah: 1,728 bytes 1Bh: 1,792 bytes 1Ch: 1,856 bytes 1Dh: 1,920 bytes 1Eh: 1,984 bytes 1Fh: 2,048 bytes	R/W
b15	—	Reserved	This bit is undefined when read. Set it to 0 when writing.			R/W

Note 1. Only change the settings of bits in the PIPEBUF register while the PIPEnCTR.PID[1:0] bits are 00b (NAK response) and the CURPIPE[3:0] bits in the D0FIFOSEL/D1FIFOSEL register are 0000b. Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY flag is 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBA changes the PID[1:0] bits to 00b (NAK response).

BUFNMB[7:0] Bits (Buffer Number)

Specify the first block number of the FIFO buffer to be allocated to the selected pipe. Blocks of the FIFO buffer that the USBA allocates to the selected pipe are shown below.
Block number: BUFNMB[7:0] to block number: BUFNMB[7:0] + (PIPEBUF.BUFSIZE[4:0] + 1) × (DBLB + 1) -1

Set the BUFNMB[7:0] bits to a value that observes the conditions below, and has a allocated block number between 04h and 87h (8.5 Kbytes of the on-chip memory).

- 00h is for default control pipe only.
- 04h is for pipe 6 only, but is available for other pipes when pipe 6 is not used. When pipe 6 is selected, writing a value to these bits is disabled. The USBA automatically allocates 04h to the BUFNMB[7:0] bits for pipe 6.
- 05h is for pipe 7 only, but is available for other pipes when pipe 7 is not used. When pipe 7 is selected, writing a value to these bits is disabled. The USBA automatically allocates 05h to the BUFNMB[7:0] bits for pipe 7.
- 06h is for pipe 8 only, but is available for other pipes when pipe 8 is not used. When pipe 8 is selected, writing a value to these bits is disabled. The USBA automatically allocates 06h to the BUFNMB[7:0] bits for pipe 8.
- 07h is for pipe 9 only, but is available for other pipes when pipe 9 is not used. When pipe 9 is selected, writing a value to these bits is disabled. The USBA automatically allocates 07h to the BUFNMB[7:0] bits for pipe 9.

BUFSIZE[4:0] Bits (Buffer Size)

Specify the FIFO buffer size (number of blocks) to be allocated to the selected pipe. One block is 64 bytes.
When the DBLB bit is set to 1, the USBA allocates twice the FIFO buffer size specified by these bits for the selected pipe. Setting the DBLB bit to 1 is effective only when the selected pipe is pipe 1 to 5. The FIFO buffer size that the USBA allocates to the selected pipe is shown below.

(BUFSIZE[4:0] + 1) × 64 × (DBLB + 1) [bytes]

Set a value within the following range for the BUFSIZE[4:0] bits.

(1) When the selected pipe is pipe 1 to 5, a value of 00h to 1Fh can be set (up to 2 Kbytes).
(2) When the selected pipe is pipe 6 to 9, only a value of 00h (64 bytes) can be set.

39.2.34 Pipe Maximum Packet Size Register (PIPEMAXP)

Address: 000D 046Ch

Bit	Symbol	Bit Name	Description	R/W
b10 to b0	MXPS[10:0]	Maximum Packet Size *1, *2	• For pipes 1 and 2 1 byte (001h) to 1,023 bytes (3FFh) • For pipes 3 to 5 8 bytes (008h), 16 bytes (010h), 32 bytes (020h), 64 bytes (040h) • For pipes 6 to 9 1 byte (001h) to 64 bytes (040h)	R/W
b11	—	Reserved	This bit is read as 0. Set it to 0 when writing.	R/W
b15 to b12	DEVSEL[3:0]	Device Select *3	b15 b12 0 0 0 0: Address 0000b 0 0 0 1: Address 0001b 0 0 1 0: Address 0010b 0 0 1 1: Address 0011b 0 1 0 0: Address 0100b 0 1 1 0: Address 0110b 0 1 1 1: Address 0111b 1 0 0 0: Address 1000b 1 0 0 1: Address 1001b 1 0 1 0: Address 1010b 1011 to 1111: Reserved	R/W

Note 1. The initial value of the MXPS[10:0] bits is 000h when the PIPESEL.PIPESEL[3:0] bits are 0000b, or 040h when the PIPESEL[3:0] bits are 0001b to 1001b.

Note 2. Only change the settings of the MXPS[10:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK response) and the CURPIPE[3:0] bits in the D0FIFOSEL/D1FIFOSEL register are 0000b. Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming the value of the PIPEnCTR.PBUSY flag is 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBA changes the PID[1:0] bits to 00b (NAK response).

Note 3. Only change the settings of the DEVSEL[3:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK response) and the CURPIPE[3:0] bits in the D0FIFOSEL/D1FIFOSEL register are 0000b. Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY flag is 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBA changes the PID[1:0] bits to 00b (NAK response).

MXPS[10:0] Bits (Maximum Packet Size)

These bits set the maximum data payload (maximum packet size) for the selected pipe. Set the value of the MXPS[10:0] bits compliant with USB 2.0 for each transfer type. While the MXPS[10:0] bits are 000h, do not write data to the FIFO buffer or set the PIPEnCTR.PID[1:0] bits to 01b (BUF response).

DEVSEL[3:0] Bits (Device Select)

In host controller operation, set the address of the communication target USB device for these bits. Set these bits after the settings for the DEVADDm register corresponding to the set value of the DEVSEL[3:0] bits has been made (m = 0 to A). When setting the DEVSEL[3:0] bits to 0010b, for example, set the DEVADD2 register. In function controller operation, set the DEVSEL[3:0] bits to 0000b.

39.2.35 Pipe Cycle Control Register (PIPEPERI)

Address: 000D 046Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[2:0]		
Value after reset:	x	x	x	0	x	x	x	x	x	x	x	x	x	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	IITV[2:0]	Interval Error Detection Interval *1	These bits specify the interval error detection interval for the selected pipe as 2^n of the frame timing.	R/W
b11 to b3	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b12	IFIS	Isochronous IN Buffer Flush	0: The buffer is not flushed. 1: The buffer is flushed.	R/W
b15 to b13	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. Only change the settings of the IITV[2:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK response). Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY flag is 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBA changes the PID[1:0] bits to 00b (NAK response).

IITV[2:0] Bits (Interval Error Detection Interval)

To modify these bits to another value after these bits have been set and USB communication has been performed, set the PIPEnCTR.PID[1:0] bits to 00b (NAK response) and then set the PIPEnCTR.ACLRM bit to 1 to initialize the interval timer. The IITV[2:0] bits are not provided for pipes 3 to 5. Write 000b to bit positions of the IITV[2:0] bits corresponding to pipes 3 to 5.

The IITV[2:0] bit function differs in host controller operation and function control operation. Refer to section 39.3.11 and section 39.3.12 for details.

IFIS Bit (Isochronous IN Buffer Flush)

This bit specifies whether to flush the buffer when the pipe specified by the PIPESEL.PIPESEL[3:0] bits is used for isochronous IN transfer.

In function controller operation, this bit provides a buffer flush function for the USBA to automatically clear the FIFO buffer if the USBA does not receive IN token from the USB host in frames at each interval set by the IITV[2:0] bits for the selected pipe of isochronous transfer type in the IN transfer direction.

When double buffer is specified (PIPECFG.DBLB bit is 1), the USBA only clears the data in the plane used earlier.

The USBA clears the FIFO buffer on receiving the SOF packet immediately after the frame in which the USBA has expected to receive IN token. Even if the SOF packet is corrupted, the FIFO buffer is cleared by the recovery function at the timing when the SOF packet should be received.

In host controller operation, set the IITV[2:0] bits to 000b.

Set the IITV[2:0] bits to 000b when the selected pipe is not for isochronous transfer.

39.2.36 Pipe n Control Register (PIPEnCTR) (n = 1 to 9)

Addresses: PIPE1CTR 000D 0470h, PIPE2CTR 000D 0472h, PIPE3CTR 000D 0474h, PIPE4CTR 000D 0476h, PIPE5CTR 000D 0478h, PIPE6CTR 000D 047Ah, PIPE7CTR 000D 047Ch, PIPE8CTR 000D 047Eh, PIPE9CTR 000D 0480h

	b15	b14	b13	b12	b11	b10	b9	b8
	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
Value after reset:	0	0	0	0	x	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	x	x	x	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on buffer state) 1 0: STALL response 1 1: STALL response	R/W
b4 to b2	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b5	PBUSY	Pipe Busy Flag	0: Pipe n is not used for the USB bus. 1: Pipe n is in use for the USB bus.	R
b6	SQMON	Sequence Toggle Bit Monitor Flag	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set *1	This bit is set to 1 when the expected value of the sequence toggle bit for the next transaction of pipe n is set for DATA1. Writing 0 has no effect. This bit is read as 0.	R/W
b8	SQCLR	Sequence Toggle Bit Clear *1	This bit is set to 1 when the expected value of the sequence toggle bit for the next transaction of pipe n is returned to DATA0. Writing 0 has no effect. This bit is read as 0.	R/W
b9	ACLRM	Auto Buffer Clear Mode *2	0: Disabled 1: Enabled (all buffers are initialized)	R/W
b10	ATREPM	Auto Response Mode *1, *3	0: Auto response mode is disabled. 1: Auto response mode is enabled	R/W
b13 to b11	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b14	INBUFM	Transmit Buffer Monitor Flag *3	0: No transmittable data is present in the FIFO buffer. 1: Transmittable data is present in the FIFO buffer.	R
b15	BSTS	Buffer Status Flag	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only change the settings of the ATREPM bit and write 1 to the SQCLR or SQSET bit while the PIPEnCTR.PID[1:0] bits are 00b (NAK response). Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY flag is 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBA changes the PIPEnCTR.PID[1:0] bits to 00b (NAK response).

Note 2. Only change the settings of the ACLRM bit while the PIPEnCTR.PID[1:0] bits are 00b (NAK response) and the CURPIPE[3:0] bits in the D0FIFOSEL/D1FIFOSEL register are 0000b. Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY flag is 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBA changes the PID[1:0] bits to 00b (NAK response).

Note 3. The ATREPM bit and the INBUFM flag in registers PIPE6CTR to PIPE9CTR are reserved. This bit is undefined when read. Set it to 0 when writing.

PID[1:0] Bits (Response PID)

These bits specify the response type for the next transaction of pipe n.
The initial setting of the PIPEnCTR.PID[1:0] bits is 00b (NAK response). Modify these bits to 01b (BUF response) to use pipe n for USB transfer. The basic operation (with no error in communication packets) of the USBA according to the setting of the PIPEnCTR.PID[1:0] bits is shown in Table 39.8 and Table 39.9.
After modifying the setting of the PIPEnCTR.PID[1:0] bits from 01b (BUF response) to 00b (NAK response) during communication using pipe n, check that the PBUSY flag is 0 to confirm that the USB transfer using pipe n has actually entered the NAK state.

The USBA modifies the PIPEnCTR.PID[1:0] bits in the following cases.

- If pipe n is in the receiving direction, and the PIPECFG.SHTNAK bit for the selected pipe is set to 1, when the USBA sets the PIPEnCTR.PID[1:0] bits to 00b (NAK response) on recognizing the completion of transfer.
- The USBA sets the PIPEnCTR.PID[1:0] bits to 11b (STALL response) on receiving a data packet with a payload exceeding the maximum packet size of pipe n.
- The USBA sets the PIPEnCTR.PID[1:0] bits to 00b (NAK response) on detecting a USB bus reset in function controller operation.
- The USBA sets the PIPEnCTR.PID[1:0] bits to 00b (NAK response) on detecting a reception error, such as a CRC error, three consecutive times in host controller operation.
- The USBA sets the PIPEnCTR.PID[1:0] bits to 11b (STALL response) on receiving the STALL handshake in host controller operation.

The PIPEnCTR.PID[1:0] bits should be set according to the procedure below.

- To make a transition from 00b (NAK response) to 10b (STALL response), write 10b.
- To make a transition from 01b (BUF response) to 11b (STALL response), write 11b.
- To make a transition from 11b (STALL response) to 00b (NAK response), write 10b and then 00b.
- To make a transition from 11b (STALL response) to 01b (BUF response), modify the bits to 00b (NAK response) and then to 01b (BUF response).

Table 39.8 USBA Operation According to the PIPEnCTR.PID[1:0] Bit Setting (In Host Controller Operation)

PID[1:0] Bit Value	Transfer Type (TYPE[1:0] Bit Value)	Transfer Direction (DIR Bit Value)	USBA Operation
00b (NAK response)	Does not depend on the setting.	Does not depend on the setting.	Does not issue tokens.
01b (BUF response)	Bulk or Interrupt	Does not depend on the setting.	Issues tokens when the DVSTCTR0.UACT bit is 1 and the FIFO buffer corresponding to the relevant pipe is ready for transmission and reception. Does not issue tokens when the DVSTCTR0.UACT bit is 0 or the FIFO buffer corresponding to the relevant pipe is not ready for transmission or reception.
	Isochronous	Does not depend on the setting.	Issues tokens when the DVSTCTR0.UACT bit is 1, regardless of the state of the FIFO buffer corresponding to pipe n. Does not issue tokens when UACT is 0.
10b or 11b (STALL response)	Does not depend on the setting.	Does not depend on the setting.	Does not issue tokens.

Table 39.9 USBA Operation According to the PIPEnCTR.PID[1:0] Bit Setting (In Function Controller Operation)

PID[1:0] Bit Value	Transfer Type (TYPE[1:0] Bit Value)	Transfer Direction (DIR Bit Value)	USBA Operation
00b (NAK response)	Bulk or Interrupt	Does not depend on the setting.	Returns a NAK response to the token from the USB host.
	Isochronous	Receiving direction (DIR bit is 0)	Returns no response to the token from the USB host.
		Transmitting direction (DIR bit is 1)	Transmits a zero-length packet in response to the token from the USB host.
01b (BUF response)	Bulk	Receiving direction (DIR bit is 0)	Receives data and returns an ACK response in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception. Otherwise, returns a NAK response.
	Interrupt	Receiving direction (DIR bit is 0)	Receives data and returns an ACK response in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception. Otherwise, returns a NAK response.
	Bulk or Interrupt	Transmitting direction (DIR bit is 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Otherwise, returns a NAK response.
	Isochronous	Receiving direction (DIR bit is 0)	Receives data in response to the OUT token from the USB host if the corresponding FIFO buffer is ready for reception. Otherwise, discards data.
		Transmitting direction (DIR bit is 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Otherwise, transmits a zero-length packet.
10b or 11b (STALL response)	Bulk or Interrupt	Does not depend on the setting.	Returns a STALL response to the token from the USB host.
	Isochronous	Does not depend on the setting.	Returns no response to the token from the USB host.

PBUSY Flag (Pipe Busy Flag)

This flag indicates whether pipe n is being used for the current transaction. The USBA modifies the PBUSY flag from 0 to 1 at the beginning of a USB transaction for pipe n, and modifies the PBUSY flag from 1 to 0 upon successful completion of one transaction. After setting the PIPEnCTR.PID[1:0] bits to 00b (NAK response), reading the PBUSY flag allows checking whether modification of the pipe setting is possible. Refer to section 39.3.7.1 for details.

SQMON Flag (Sequence Toggle Bit Monitor Flag)

This flag indicates the value of the sequence toggle bit for the next transaction of pipe n. When pipe n is not the isochronous transfer type, the USBA toggles the SQMON flag upon successful completion of the transaction. However, the USBA does not toggle the SQMON flag when a data PID inconsistency occurs during transfer in the receiving direction.

SQSET Bit (Sequence Toggle Bit Set)

This bit is set to 1 when the value of the sequence toggle bit for the next transaction of pipe n is set for DATA1. When the SQSET bit is set to 1, the USBA sets the expected value of the sequence toggle bit of pipe n for DATA1.

SQCLR Bit (Sequence Toggle Bit Clear)

This bit is set to 1 when the value of the sequence toggle bit for the next transaction of pipe n is returned to DATA0. When the SQCLR bit is set to 1, the USBA sets the expected value of the sequence toggle bit of pipe n for DATA0.

ACLRM Bit (Auto Buffer Clear Mode)

This bit enables or disables auto buffer clear mode for pipe n. To completely clear the information in the FIFO buffer allocated to pipe n, write 1 and then 0 to the ACLRM bit continuously. Also, allow 100 ns between modifying the ACLRM bit from 1 to 0 as sequence execution time for the internal hardware. The following lists the information cleared by the USBA by writing 1 and 0 continuously to the ACLRM bit.

(a) All data in the FIFO buffer allocated to pipe n (both planes in double buffer mode).

(b) The interval counter value when the transfer type for pipe n is isochronous transfer or interrupt transfer.

Perform this processing when the following conditions are satisfied:

(a) When all data in the FIFO buffer allocated to pipe n needs to be cleared.

(b) When the interval counter value needs to be reset.

(c) When modifying the PIPECFG.DIR, DBLB, or BFRE bit.

(d) When the transaction count function is forcibly ended.

ATREPM Bit (Auto Response Mode)

This bit enables or disables the auto response of pipe n.

When the transfer type of pipe n is set to bulk IN function controller operation, this bit can be set to 1.

When this bit is set to 1, the USBA responds as follows to the token from the USB host.

(1) When pipe n is bulk IN transfer type (PIPECFG.TYPE[1:0] bits are 01b and the PIPECFG.DIR bit is 1)
When the ATREPM bit is 1 and the PIPEnCTR.PID[1:0] bits are 01b (BUF response), the USBA transmits a zero-length packet in response to IN token. The USBA updates (toggles) the sequence toggle bit (data PID) each time the USBA receives ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received). In this case, the USBA generates no BRDY interrupt or BEMP interrupt.

(2) When pipe n is bulk OUT transfer type (PIPECFG.TYPE[1:0] bits are 01b and the PIPECFG.DIR bit is 0)
When the ATREPM bit is 1 and the PIPEnCTR.PID[1:0] bits are 01b (BUF response), the USBA returns a NAK response to OUT token and generates an NRDY interrupt. When performing USB communication with this bit set to 1, set this bit while the FIFO buffer is empty. Do not write data to the FIFO buffer during USB communication with this bit set to 1. Be sure to set this bit to 0 when pipe n is used for isochronous transfer. In host controller operation, be sure to set this bit to 0.

INBUFM Flag (Transmit Buffer Monitor Flag)

This flag indicates the FIFO buffer status for pipe n in the transmitting direction. With pipe n set in the transmitting direction (PIPECFG.DIR bit is 1), the USBA sets the INBUFM flag to 1 when the CPU or DMAC/DTC has completed writing data to at least one FIFO buffer plane. The USBA sets the INBUFM flag to 0 when the USBA has completely transmitted the data from the FIFO buffer plane to which all the data has been written. In double buffer mode (PIPECFG.DBLB bit is 1), the USBA sets the INBUFM flag to 0 when the USBA has completely transmitted data from the two FIFO buffer planes before the CPU or DMAC/DTC completes writing data to one FIFO buffer plane. The INBUFM flag indicates the same value as the BSTS flag when pipe n is in the receiving direction (PIPECFG.DIR bit is 0).

BSTS Flag (Buffer Status Flag)

This flag indicates the FIFO buffer status for pipe n. The meaning of the BSTS flag varies as shown in Table 39.10 depending on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DCLRM bit in the D0FIFOSEL/ D1FIFOSEL register.

Table 39.10 Operation of the BSTS Flag

DIR Bit Value	BFRE Bit Value	DCLRM Bit Value	Meaning of the BSTS Flag
0	0	0	Becomes 1 when receive data can be read from the FIFO buffer, and becomes 0 upon completion of data read.
		1	Do not set this combination.
	1	0	Becomes 1 when receive data can be read from the FIFO buffer, and becomes 0 when the BCLR bit is set in a FIFO port control register to 1 after the data read has been completed.
		1	Becomes 1 when receive data can be read from the FIFO buffer, and becomes 0 upon completion of data read.
1	0	0	Becomes 1 when transmit data can be written to the FIFO buffer, and becomes 0 upon completion of data write.
		1	Do not set this combination.
	1	0	Do not set this combination.
		1	Do not set this combination.

39.2.37 Pipe n Transaction Counter Enable Register (PIPEnTRE) (n = 1 to 5)

Addresses: PIPE1TRE 000D 0490h, PIPE2TRE 000D 0494h, PIPE3TRE 000D 0498h, PIPE4TRE 000D 049Ch, PIPE5TRE 000D 04A0h

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	TRENB	TRCLR
Value after reset:	x	x	x	x	x	x	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b8	TRCLR	Transaction Counter Clear	This bit is set to 1 when the transaction counter is reset. Writing 0 has no effect. This bit is read as 0.	R/W
b9	TRENB	Transaction Counter Enable	0: The transaction count function is disabled. 1: The transaction count function is enabled.	R/W
b15 to b10	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

Note: Only change the settings of the PIPEnTRE register while the PIPEnCTR.PID[1:0] bits are 00b (NAK response). Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY flag is 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBA changes the PID[1:0] bits to 00b (NAK response).

TRCLR Bit (Transaction Counter Clear)

The USBA clears the current count value of the transaction counter corresponding to pipe n and then sets the TRCLR bit to 0.

TRENB Bit (Transaction Counter Enable)

This bit enables or disables the transaction counter.

When the total number of packets is set in the PIPEnTRN.TRNCNT[15:0] bits and then the TRENB bit is set to 1 for the pipe in the receiving direction, the USBA performs the following control upon completion of receiving packets as many as the value of the TRNCNT[15:0] bits.

- In continuous transfer mode (CNTMD bit is 1), the USBA toggles the TRENB bit to the CPU side even if the FIFO buffer is not full at the end of reception.
- When the PIPECFG.SHTNAK bit is 1, the USBA modifies the PIPEnCTR.PID[1:0] bits of the corresponding pipe to 00b (NAK response) upon completion of receiving packets as many as the value of the TRNCNT[15:0] bits.
- When the PIPECFG.BFRE bit is 1, the USBA asserts the BRDY interrupt upon completion of reading the final data after receiving packets as many as the value of the TRNCNT[15:0] bits.

For pipes in the transmitting direction, set the TRENB bit to 0.

When the transaction count function is not used, set the TRENB bit to 0.

When the transaction count function is used, set the TRNCNT[15:0] bits and then set the TRENB bit to 1. Furthermore, set the TRENB bit to 1 before receiving the first packet to be counted by the transaction counter.

39.2.38 Pipe n Transaction Counter Register (PIPEnTRN) (n = 1 to 5)

Addresses: PIPE1TRN 000D 0492h, PIPE2TRN 000D 0496h, PIPE3TRN 000D 049Ah, PIPE4TRN 000D 049Eh, PIPE5TRN 000D 04A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TRNCNT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TRNCNT[15:0]	Transaction Counter *1	• When writing to the register These bits specify the number of total packets (number of transactions) to be received by the relevant pipe. • When reading the register When the PIPEnTRE.TRENB bit is 0, these bits indicate the specified number of transactions. When the PIPEnTRE.TRENB bit is 1, these bits indicate the number of transaction in the current count (current count value).	R/W

Note 1. Only change the settings of the TRNCNT[15:0] bits while the PIPEnTRE.TRENB bit is 0 and the PIPEnCTR.PID[1:0] bits are 00b (NAK response). Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY flag is 0. However, the PIPEnCTR.PBUSY flag does not need to be checked by software if the USBA changes the PID[1:0] bits to 00b (NAK response).

The value set to the PIPEnTRN register is retained even if a USB bus reset occurs.

TRNCNT[15:0] Bits (Transaction Counter)

The USBA increments the value of the TRNCNT[15:0] bits (current counter value) when all of the following conditions for reception state are satisfied.

- The PIPEnTRE.TRENB bit is 1.
- When receiving a packet, the value of the TRNCNT[15:0] bits is different from the current counter value + 1.
- The payload of the received packet matches the value set to the PIPEMAXP.MXPS[10:0] bits.

The USBA sets the value of the TRNCNT[15:0] bits (current counter value) to 0 when any of the following conditions is satisfied.

(1) All of the following conditions are satisfied.
- The PIPEnTRE.TRENB bit is 1.
- When receiving a packet, the value of the TRNCNT[15:0] bits is the same as the current counted value + 1.
- The payload of the received packet matches the value set to the PIPEMAXP.MXPS[10:0] bits.

(2) Both of the following conditions are satisfied.
- The PIPEnTRE.TRENB bit is 1.
- The USBA has received a short packet.

(3) The following condition is satisfied.
- The PIPEnTRE.TRCLR bit is set to 1.

For pipes in the transmitting direction, set the TRNCNT[15:0] bits to 0. When the transaction count function is not used, set the TRNCNT[15:0] bits to 0. The number of transactions can be set to the TRNCNT[15:0] bits only when the PIPEnTRE.TRENB bit is 0. If the specified number of transactions is modified, set the PIPEnTRE.TRCLR bit to 1 (the current counter is cleared) before setting the PIPEnTRE.TRENB bit to 1 (count starts).

39.2.39 Device Address m Configuration Register (DEVADDm) (m = 0 to A)

Addresses: DEVADD0 000D 04D0h, DEVADD1 000D 04D2h, DEVADD2 000D 04D4h, DEVADD3 000D 04D6h, DEVADD4 000D 04D8h, DEVADD5 000D 04DAh, DEVADD6 000D 04DCh, DEVADD7 000D 04DEh, DEVADD8 000D 04E0h, DEVADD9 000D 04E2h, DEVADDA 000D 04E4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	USBSPD[1:0]		—	—	—	—	—	—
Value after reset:	x	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b7, b6	USBSPD[1:0]	Transfer Speed of Communication Target Device	b7 b6 0 0: DEVADDm register is not used. 0 1: Low-speed 1 0: Full-speed 1 1: Reserved	R/W
b15 to b8	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

The DEVADDm register specifies the communication speed of the communication target peripheral device for pipes 0 to 9. In host controller operation, set each bit in the DEVADDm register before communications start to pipes.
Modify each bit in the DEVADDm register while there is no valid pipe specified by such a bit. The valid pipe is defined as the pipe satisfying both the conditions listed below.

- The DEVSEL[3:0] bits specify the DEVADDm register.
- The PIPEnCTR.PID[1:0] bits of the selected pipe are 01b (BUF response), or the default control pipe is the selected pipe and the DCPCTR.SUREQ bit is set to 1.

In function controller operation, set each bit in this register to 0.

USBSPD[1:0] Bits (Transfer Speed of Communication Target Device)

In host controller operation, the USBA generates a packet according to the value of these bits.

39.2.40 Low Power Control Register (LPCTRL)

Address: 000D 0500h

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	0

	b7	b6	b5	b4	b3	b2	b1	b0
	HWUPM	—	—	—	—	—	—	—
Value after reset:	0	x	x	x	x	x	x	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b7	HWUPM	Return Mode Setting	0: Hardware return is not made while the CPU clock is stopped. 1: Hardware return is made while the CPU clock is stopped.	R/W
b8	—	Reserved	This bit is read as 0. Set it to 0 when writing.	R/W
b15 to b9	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

HWUPM Bit (Return Mode Setting)

This bit specifies whether to enable the hardware processing for return from a low power state of the LPM L1 state even while the CPU clock is stopped. In function controller operation, the processing for return from low power mode upon detecting resume signal is enabled even while the CPU clock is stopped. The HWUPM bit specifies whether to detect resume signal while the CPU clock is stopped. The PL1CTRL1.L1EXTMD bit controls whether to make hardware return. To make hardware return from a low power state of the LPM L1 state while the CPU clock is stopped, set the HWUPM bit and the PL1CTRL1.L1EXTMD bit to 1.

39.2.41 Low Power Status Register (LPSTS)

Address: 000D 0502h

	b15	b14	b13	b12	b11	b10	b9	b8
	—	SUSPENDM	—	—	—	—	—	—
Value after reset:	x	0	x	0	x	x	x	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	0	x	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b2	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b3	—	Reserved	This bit is read as 0. Set it to 0 when writing.	R/W
b7 to b4	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b8	—	Reserved	This bit is read as 0. Set it to 0 when writing.	R/W
b11 to b9	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b12	—	Reserved	This bit is read as 0. Set it to 0 when writing.	R/W
b13	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b14	SUSPENDM	SuspendM Control	0: Suspend mode 1: Normal mode	R/W
b15	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W

SUSPENDM Bit (SuspendM Control)

This bit controls the SuspendM signal output to the PHY. The initial value is 0 and the PHY is in suspend mode. To supply the PHY clock to operate the USB2.0 host/function controller, set this bit to 1. Refer to section 39.3.3 for details on setting clocks.

When the SUSPENDM bit is 0, the registers in the USBA are not writable, but are readable. However, registers listed in Table 39.11 below are write accessible even when the SUSPENDM bit is 0.

Table 39.11 Registers That Can be Written by Software when the SUSPENDM Bit is 0

Address	Register/Bit
000D 0400h	SYSCFG register
000D 0402h	BUSWAIT register
000D 0432h	INTENB1.PDDETINTE bit
000D 043Eh	PHYSET register
000D 0500h	LPCTRL register
000D 0502h	LPSTS register
000D 0540h	BCCTRL register

However, the value written to the SYSCFG register while the PHY clock is stopped is updated after the PHY clock is oscillated. The PHY clock is oscillated in the following cases.

(1) The SUSPENDM bit is set to 1 and then the PLLSTA.PLLLOCK flag is set to 1 after the predetermined time has passed.

(2) 48 and 60 MHz are supplied to the USBA from the main clock with the SUSPENDM bit set to 0 (Refer to section 39.2.16 for details on classic-only mode.)

While the PL1CTRL1.L1EXTMD bit is 0, the SUSPENDM bit must be controlled by software. When the PL1CTRL1.L1EXTMD bit is 1, when transitioning to the L1 or L2 state (suspended state), the SUSPENDM bit must be set to 0 by software, but when exiting the L1 or L2 state, the USBA sets the SUSPENDM bit to 1.

39.2.42 Battery Charging Control Register (BCCTRL)

Address: 000D 0540h

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	PDDETSTS	CHGDETSTS
Value after reset:	x	x	x	x	x	x	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	VDMSRCE	IDPSINKE	VDPSRCE	IDMSINKE	IDPSRCE
Value after reset:	x	x	x	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IDPSRCE	D+ Line IDPSRC Output Control	0: Stopped 1: 10 µA output	R/W
b1	IDMSINKE	D– Line 0.6 V Input Detection (Comparator and Sink) Control	0: Detection off 1: Detection on (comparator and sink current on)	R/W
b2	VDPSRCE	D+ Line VDPSRC (0.6 V) Output Control	0: Stopped 1: 0.6 V output	R/W
b3	IDPSINKE	D+ Line 0.6 V Input Detection (Comparator and Sink) Control	0: Detection off 1: Detection on (comparator and sink current on)	R/W
b4	VDMSRCE	D– Line VDMSRC (0.6 V) Output Control	0: Stopped 1: 0.6 V output	R/W
b7 to b5	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b8	CHGDETSTS	D– Line 0.6 V Input Detection Status Flag *1	0: Not detected 1: Detected	R
b9	PDDETSTS	D+ Line 0.6 V Input Detection Status Flag *2	0: Not detected 1: Detected	R
b15 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. Valid when the IDMSINKE bit is 1.
Note 2. Valid when the IDPSINKE bit is 1.

IDPSRCE Bit (D+ Line IDPSRC Output Control)

In function controller operation, set this bit to 1 to perform Data Contact Detect. In the Battery Charging Specification Revision 1.2, there are two methods to handle Data Contact Detect; the method realized by software wait and the method to contact the data line by hardware. The IDPSRCE bit adopts the latter method. When the IDPSRCE bit is set to 1, the USBA enables the IDP_SRC circuit and, at the same time, controls D– line pull-down.

IDMSINKE Bit (D– Line 0.6 V Input Detection (Comparator and Sink) Control)

In function controller operation, the following can be detected when the IDMSINKE bit is set to 1:

- During Primary Detection, VDMSRC (0.6 V) is input to the USBA_DM pin from the host
- During Primary Detection, VDPSRC (0.6 V) output from the USBA to the USBA_DP pin is input to the USBA_DM pin via the USB host

VDPSRCE Bit (D+ Line VDPSRC (0.6 V) Output Control)

In function controller operation, if Primary Detection is executed, this bit controls VDPSRC (0.6 V) output from the USBA_DP pin.

IDPSINKE Bit (D+ Line 0.6 V Input Detection (Comparator and Sink) Control)

When the IDPSINKE bit is set to 1, the following states can be detected:

- In host controller operation
  During Primary Detection, VDPSRC (0.6 V) is input to the USBA_DP pin from a peripheral device.
- In function controller operation
  During Secondary Detection, VDPSRC (0.6 V) output from the USBA to the USBA_DM pin is input to the USBA_DP pin via the host.

VDMSRCE Bit (D– Line VDMSRC (0.6 V) Output Control)

- In host controller operation
  During Primary Detection, this bit controls VDMSRC (0.6 V) output from the USBA_DM pin.
- In function controller operation
  During Secondary Detection, this bit controls VDMSRC (0.6 V) output from the USBA_DM pin.

CHGDETSTS Flag (D– Line 0.6 V Input Detection Status Flag)

In function controller operation, this flag is enabled when the IDPSINKE bit is 1.
The CHGDETSTS flag becomes 1 under the following conditions:

- During Primary Detection, VDMSRC (0.6 V) is input to the USBA_DM pin from the USB host.
- During Primary Detection, the VDPSRC (0.6 V) output from the USBA to the USBA_DP pin is input to the USBA_DM pin via the USB host.

PDDETSTS Flag (D+ Line 0.6 V Input Detection Status Flag)

The PDDETSTS flag is enabled when the IDPSINKE bit is 1.
The PDDETSTS flag becomes 1 under the following conditions:

- In host controller operation
  During Primary Detection, VDPSRC (0.6 V) is input to the USBA_DP pin from a peripheral device.
- In function controller operation
  During Secondary Detection, the VDMSRC (0.6 V) output from the USBA to the USBA_DM pin is input to the USBA_DP pin via the host.

39.2.43 Function L1 Control Register 1 (PL1CTRL1)

Address: 000D 0544h

	b15	b14	b13	b12	b11	b10	b9	b8
	—	L1EXTMD	—	—	HIRDTHR[3:0]			
Value after reset:	x	0	x	x	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	DVSQ[3]	DVSQ[2:0]			L1NEGOMD	L1RESPMD[1:0]		L1RESPEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	L1RESPEN	L1 Response Enable	0: LPM is not supported. 1: LPM is supported.	R/W
b2, b1	L1RESPMD[1:0]	L1 Response Mode	b2 b1 0 0: NYET response 0 1: ACK response 1 0: STALL response 1 1: Comply with the L1NEGOMD bit	R/W
b3	L1NEGOMD	L1 Response Negotiation Control *1	0: When the received HIRD field is larger than the HIRDTHR[3:0] bit value, an ACK response is returned. When the received HIRD field is smaller than or equal to the value set to the HIRDTHR[3:0] bits, an NYET response is returned. 1: When the received HIRD field is smaller than the HIRDTHR[3:0] bit value, an ACK response is returned. When the received HIRD field is greater than or equal to the value set to the HIRDTHR[3:0] bits, an NYET response is returned.	R/W
b6 to b4	DVSQ[2:0]	Extension Device State Flag	These flags indicate the same value with the INTSTS0.DVSQ[2:0] flags.	R
b7	DVSQ[3]		This flag indicates the L1 state together with the INTSTS0.DVSQ[2:0] flags. 0000: Powered state 0001: Default state 0010: Address state 0011: Configured state 0100: Suspended state 0101: Suspended state 0110: Suspended state 0111: Suspended state 1000 to 1011: L1 state	R
b11 to b8	HIRDTHR[3:0]	L1 Response Negotiation Threshold Value *1	HIRD threshold value used when the L1RESPMD[1:0] bits are 11b. The format is the same as the HIRD field in the HL1CTRL2 register.	R/W
b13, b12	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b14	L1EXTMD	PHY Control Mode at L1 Return	0: The LPSTS.SUSPENDM bit is not set by hardware when exiting the L1 state. 1: The LPSTS.SUSPENDM bit is set by hardware when exiting the L1 state.	R/W
b15	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W

Note 1. The L1NEGOMD bit is only valid when the L1RESPMD[1:0] bits are 11b.

L1RESPEN Bit (L1 Response Enable)

When the L1RESPEN bit is 0, even if the LPM token is received, the USBA returns no response in regards to the LPM token. When the L1RESPEN bit is 1, if the LPM token is received, the USBA returns a response in accordance with the L1RESPMD[1:0] bit setting.

L1RESPMD[1:0] Bits (L1 Response Mode)

When the L1RESPEN bit is set to 1, the USBA returns a response to the LPM token in accordance with the L1RESPMD[1:0] bit setting.

L1NEGOMD Bit (L1 Response Negotiation Control)

The L1NEGOMD bit sets the response conditions for the HIRD[3:0] bit value. The L1NEGOMD bit is valid when the L1RESPMD[1:0] bits are 11b.

DVSQ[2:0] Flags (DVSQ Extension Flag)

These flags mirror the INTSTS0.DVSQ[2:0] flags.

DVSQ[3] Flag (DVSQ Extension Flag)

This flag is the fourth bit of the device state flags (DVSQ[2:0] flags).

HIRDTHR[3:0] Bits (L1 Response Negotiation Threshold Value)

The HIRDTHR[3:0] bits specify the HIRD threshold value used by the HIRD[3:0] bit value in the negotiation. The HIRDTHR[3:0] bits are valid when the L1RESPMD[1:0] bits are 11b.
The format of the set value is the same as the HL1CTRL2.HIRD[3:0] bits.

L1EXTMD Bit (PHY Control Mode at L1 Return)

In the L1 state, with the LPSTS.SUSPENDM bit set to 0 to stop the PHY clock, this bit specifies the LPSTS.SUSPENDM bit control method when receiving the resume signal.
When supporting the L1 state, set the L1EXTMD bit to 1 during the initialization process.
When entering the L1 state, the LPSTS.SUSPENDM bit must be set to 0 by software regardless of the L1EXTMD bit value.
When the L1EXTMD bit is set to 1, the USBA also sets the LPSTS.SUSPENDM bit to 1 when exiting the L2 state (suspended state).

39.2.44 Function L1 Control Register 2 (PL1CTRL2)

Address: 000D 0546h

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	RWEMON	HIRDMON[3:0]			
Value after reset:	x	x	x	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b11 to b8	HIRDMON[3:0]	HIRD Value Monitor Flag	These bits reflect the HIRD field value of the recently received LPM token.	R/W
b12	RWEMON	bRemoteWake Value Monitor Flag	This bit reflects the bRemoteWake bit value of the recently received LPM token.	R/W
b15 to b13	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

HIRDMON[3:0] Flags (HIRD Value Monitor Flag)

These flags monitor the HIRD field value of the received LPM token. These bits reflect the HIRD field value of the recently received LPM token.

RWEMON Flag (bRemoteWake Value Monitor Flag)

This flag monitors the bRemoteWake field value of the received LPM token. This flag reflects the bRemoteWake field value of the recently received LPM token.

39.2.45 Host L1 Control Register 1 (HL1CTRL1)

Address: 000D 0548h

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	L1STATUS[1:0]		L1REQ
Value after reset:	x	x	x	x	x	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	L1REQ	L1 Transition Request	Set this bit to 1 when making a request that a peripheral device transition to the L1 state. This bit is automatically set to 0 when the LPM transaction is completed.	R/W
b2, b1	L1STATUS[1:0]	L1 Request Completion Status	b2 b1 0 0: ACK response received 0 1: NYET response received 1 0: STALL response received 1 1: Transaction error occur ed	R/W
b15 to b3	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W

L1REQ Bit (L1 Transition Request)

Set this bit to 1 to transition a peripheral device to the L1 state. When the L1REQ bit is set to 1, the USBA starts the LPM transaction, and when the transaction is complete, the L1REQ bit is set to 0.

L1STATUS[1:0] Bits (L1 Request Completion Status)

These bits reflect the result of the LPM transaction executed by the L1REQ bit.

39.2.46 Host L1 Control Register 2 (HL1CTRL2)

Address: 000D 054Ah

	b15	b14	b13	b12	b11	b10	b9	b8
	BESL	—	—	L1RWE	HIRD[3:0]			
Value after reset:	0	x	x	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	L1ADDR[3:0]			
Value after reset:	x	x	x	x	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	L1ADDR[3:0]	LPM Token Device Address	These bits specify the value to be set in the ADDR field of the LPM token.	R/W
b7 to b4	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b11 to b8	HIRD[3:0]	LPM Token HIRD	These bits specify the value to be set in the BESL field of the LPM token.	R/W
b12	L1RWE	LPM Token L1 Remote Wakeup Enable	These bits specify the value to be set in the bRemoteWake field of the LPM token.	R/W
b14, b13	—	Reserved	These bits are undefined when read. Set them to 0 when writing.	R/W
b15	BESL	BESL & Alternate HIRD	This bit selects the resume signal output period when exiting the L1 state.	R/W

L1ADDR[3:0] Bits (LPM Token Device Address)

When the HL1CTRL1.L1REQ bit is set to 1, these bits specify the value to be set in the ADDR field of the LPM token transmitted by the USBA.

HIRD[3:0] Bits (LPM Token HIRD)

When the HL1CTRL1.L1REQ bit is set to 1, these bits specify the value to be set in the BESL field of the LPM token transmitted by the USBA. Table 39.12 indicates the Relationship Between the Values Set to the HIRD[3:0] and BESL Bits With the HIRD and BESL Values.

Table 39.12 Relationship Between the Values Set to the HIRD[3:0] and BESL Bits With the HIRD and BESL Values

Value Set to the HIRD[3:0] Bits	When the BESL Bit is 0	When the BESL Bit is 1	
	HIRD value	HIRD value	BESL value
0000b	50 µs (do not set this value)	75 µs	125 µs
0001b	125 µs	100 µs	150 µs
0010b	200 µs	150 µs	200 µs
0011b	275 µs	250 µs	300 µs
0100b	350 µs	350 µs	400 µs
0101b	425 µs	450 µs	500 µs
0110b	500 µs	950 µs	1000 µs
0111b	575 µs	1,950 µs	2,000 µs
1000b	650 µs	2,950 µs	3,000 µs
1001b	725 µs	3,950 µs	4,000 µs
1010b	800 µs	4,950 µs	5,000 µs
1011b	875 µs	5,950 µs	6,000 µs
1100b	950 µs	6,950 µs	7,000 µs
1101b	1,025 µs (do not set this value)	7,950 µs	8,000 µs
1110b	1,100 µs (do not set this value)	8,950 µs	9,000 µs
1111b	1,175 µs (do not set this value)	9,950 µs	10,000 µs

Note: In host controller operation, the HIRD[3:0] bit setting value is used for the resume signal output period when exiting the L1 state.

L1RWE Bit (LPM Token L1 Remote Wakeup Enable)

When the HL1CTRL1.L1REQ bit is set to 1, this bit specifies the value to be set in the bRemoteWake field of the LPM token transmitted by the USBA. Depending on the L1RWE bit setting, the USBA does not control detection of the remote wakeup signal in the L1 state. The remote wakeup signal is controlled by the DVSTCTR0.RWUPE bit as is the case when the USBA is in the suspended state.

BESL Bit (BESL & Alternate HIRD)

In host controller operation, this bit selects the resume signal output period when exiting the L1 state. Refer to the description of the HIRD[3:0] bits for details.

39.2.47 Deep Standby USB Transceiver Control/Pin Monitor Register (DPUSR0R)

Address: 000D 0560h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	DVBSTSHM	—	DOVCBHM	DOVCAHM	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b19 to b0	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b20	DOVCAHM	OVRCURA Input Flag	This flag indicates the level of the input signal to the USBA_OVRCURA pin.	R
b21	DOVCBHM	OVRCURB Input Flag	This flag indicates the level of the input signal to the USBA_OVRCURB pin.	R
b22	—	Reserved	This bit is read as 0. Set it to 0 when writing.	R/W
b23	DVBSTSHM	VBUS Input Flag	This flag indicates the level of the input signal to the USBA_VBUS pin.	R
b31 to b24	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

39.2.48 Deep Standby USB Suspend/Resume Interrupt Register (DPUSR1R)

Address: 000D 0564h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	DVBSTSH	—	DOVCBH	DOVCAH	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	DVBSTSHE	—	DOVCBHE	DOVCAHE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b4	DOVCAHE	OVRCURA Interrupt Enable/Clear	0: Exit from deep software standby mode disabled, and the DOVCAH flag is cleared 1: Exit from deep software standby mode enabled	R/W
b5	DOVCBHE	OVRCURB Interrupt Enable/Clear	0: Exit from deep software standby mode disabled, and the DOVCBH flag is cleared 1: Exit from deep software standby mode enabled	R/W
b6	—	Reserved	This bit is read as 0. Set it to 0 when writing.	R/W
b7	DVBSTSHE	VBUS Interrupt Enable/Clear	0: Exit from deep software standby mode disabled, and the DVBSTSH flag is cleared 1: Exit from deep software standby mode enabled	R/W
b19 to b8	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b20	DOVCAH	OVRCURA Interrupt Source Return Status Flag *1	0: USBA is in deep software standby mode 1: USBA exited deep software standby mode	R
b21	DOVCBH	OVRCURB Interrupt Source Return Status Flag *2	0: USBA is in deep software standby mode 1: USBA exited deep software standby mode	R
b22	—	Reserved	This bit is read as 0. Set it to 0 when writing.	R/W
b23	DVBSTSH	VBUS Interrupt Source Return Status Flag *3	0: USBA is in deep software standby mode 1: USBA exited deep software standby mode	R
b31 to b24	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. This bit becomes 0 when the DOVCAHE bit is set to 0.
Note 2. This bit becomes 0 when the DOVCBHE bit is set to 0.
Note 3. This bit becomes 0 when the DVBSTSHE bit is set to 0.

39.3 Operation

39.3.1 System Control

This section describes register settings necessary to initialize the USBA and registers necessary for power consumption control.

39.3.1.1 Setting Registers Associated with the USBA

After supplying the PHY clock, set the SYSCFG.USBE bit to 1 to enable USBA operation.
Refer to section 39.3.3 for details on supplying the PHY clock.

39.3.1.2 Selecting the Controller Function

The USBA can be used in host controller operation or function controller operation. Set the SYSCFG.DCFM bit to select a controller operation. However, modify the DCFM bit in the initial settings immediately after a reset, or when pulling up the D+ line is disabled, and pulling down the D+ and D– lines is disabled (both SYSCFG.DPRPU bit and SYSCFG.DRPD bit are 0).

39.3.2 Controlling the USB Data Bus Resistors

The USBA incorporates pull-up and pull-down resistors for the D+ and D– lines. Set the SYSCFG.DPRPU bit and SYSCFG.DRPD bit to select pull up or pull down.
In function controller operation, confirm there is a connection to the USB host, then set the SYSCFG.DPRPU bit to 1 to pull up the D+ line.
If the SYSCFG.DPRPU bit is set to 0 while communicating with a personal computer, the pull-up resistor for the USB data line is disabled, and the USB host can be notified that the device has been disconnected.
In host controller operation, set the SYSCFG.DRPD bit to 1 to pull down the D+ and D– lines.
Table 39.13 lists the settings for the USB data bus resistors. Set the DRPD and DPRPU bits to select the USB data bus resistors.

Table 39.13 Controlling the USB Data Bus Resistors (Excluding OTG Operation)

Settings		USB Data Bus Resistor Control		
DRPD bit	DPRPU bit	D– line	D+ line	Remarks
0	0	Open	Open	When the resistor is not used
0	1	Open	Pull-up	When operating as the function controller
1	0	Pull-down	Pull-down	When operating as the host controller
1	1	—	—	Do not set this value except for OTG operation

39.3.3 Supplying Clocks

The two input clocks required for the USBA are shown in Table 39.14.

Table 39.14 Input Clocks

Input Clock Name	Description
Peripheral module clock (PCLKA)	PCLKA is used to access USBA registers from the CPU, DMAC, and DTC. The clock frequency has no restrictions on the USBA side.
PHY clock	Generating the PHY clock: External input or internal supply • External input A 20 MHz or 24 MHz clock is supplied to the EXTAL pin from outside the MCU, and the PHY clock is generated by the PLL in the UTMI-PHY. Specifications for the external clock must be strictly followed, especially the jitter characteristics of ±50 ppm. • Internal supply The PHY clock is supplied when 48 and 60 MHz are supplied to the UTMI-PHY module and the PHYSET.HSEB bit is set to 1 (classic-only mode is enabled).

Figure 39.2 shows the PHY Clock Settings in Classic-Only Mode.

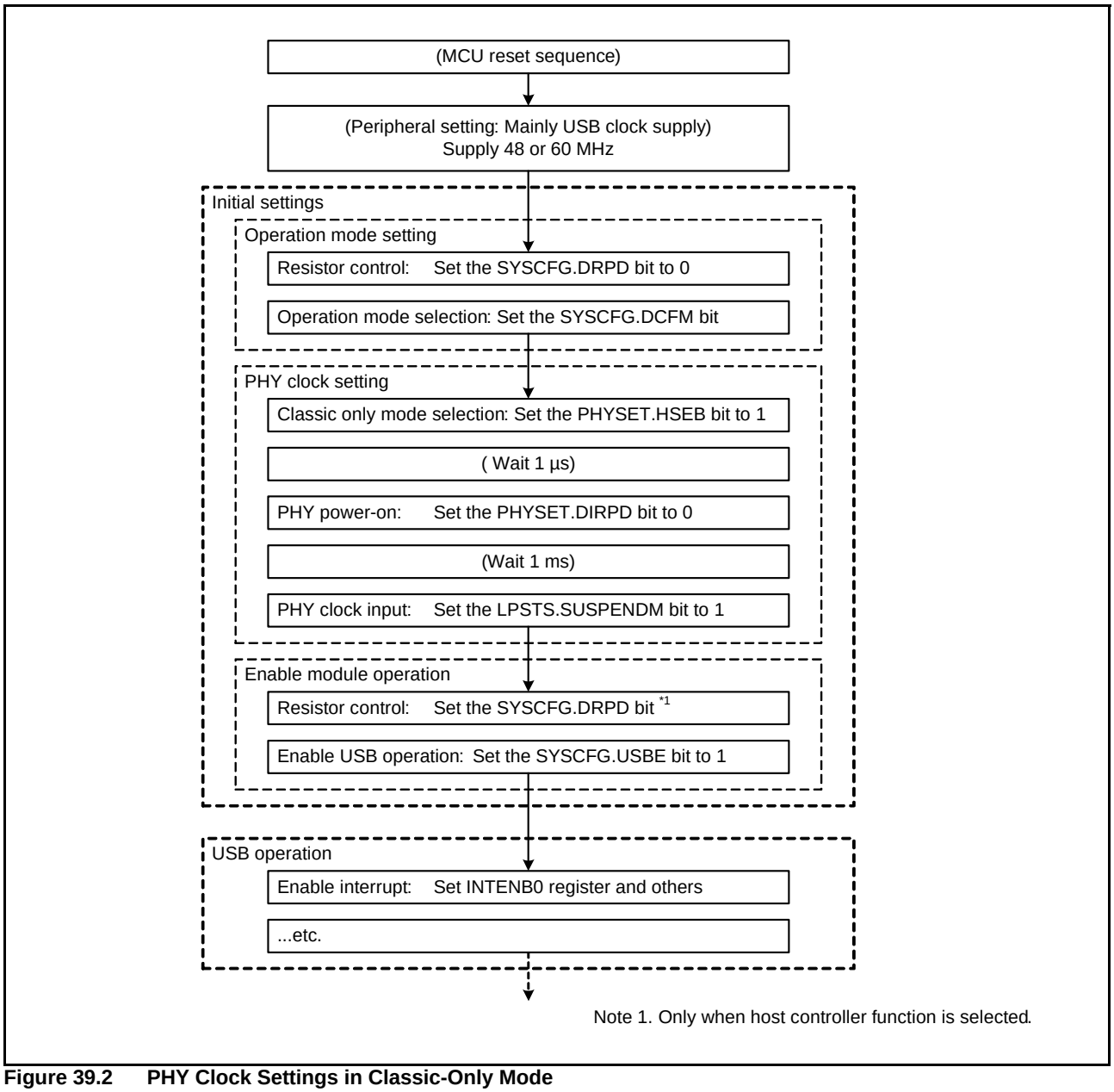


Figure 39.2 PHY Clock Settings in Classic-Only Mode

Figure 39.3 shows the PHY Clock Settings When Not Using Classic-Only Mode.

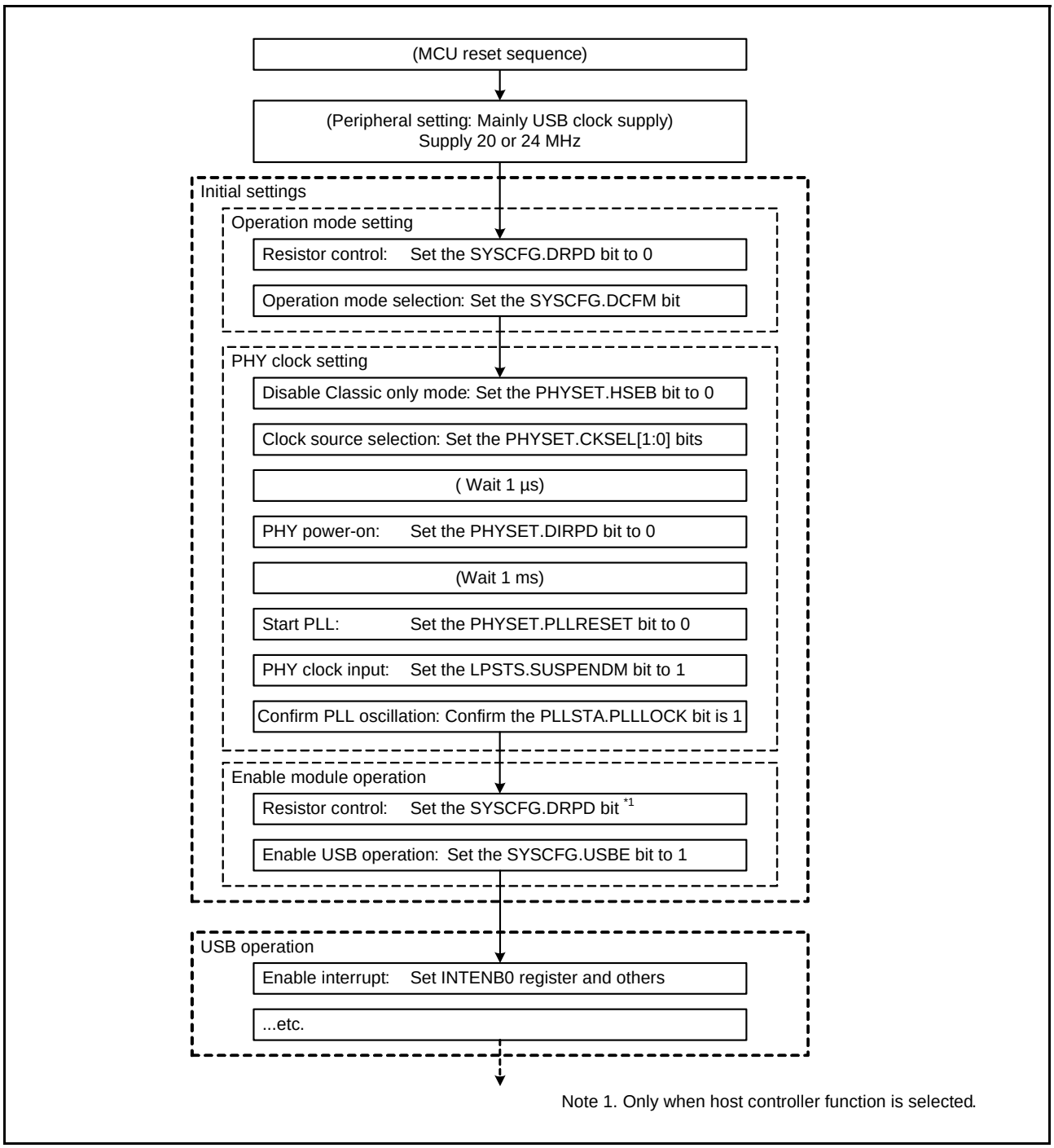


Figure 39.3 PHY Clock Settings When Not Using Classic-Only Mode

39.3.4 Notes on Stopping Clock

The CPU clock and PHY clock can be stopped when a device is disconnected or the USBA is in the suspend state. However, in function controller operation, if either of these clocks are stopped while the USBA is in the suspended state, they must be resupplied when exiting the suspended state. The PHY clock must be resupplied within 9.5 ms of generating the resume interrupt.

39.3.5 Interrupts

Table 39.15 lists the interrupt functions of the USBA. When a condition to generate an interrupt is satisfied, and the interrupt output is enabled using the corresponding interrupt enable register, the USBA issues a USBAR interrupt request to the interrupt controller (ICU).

Table 39.15 List of Interrupt Functions (1/2)

Flag That Becomes 1	Interrupt	Interrupt Source	Function Generated	Status Flag
VBINT	VBUS interrupt	• VBUS input pin state change was detected (from low to high and from high to low).	Host *1, function	INTSTS0. VBSTS
RESM	Resume interrupt	• USB bus state change was detected in the suspended state. (J state to K state or J state to SE0)	Function	—
SOFR	Frame number refresh interrupt	In host controller operation • An SOF packet with a different frame number was transmitted In function controller operation • An SOF packet with a different frame number was received. • Failed to receive an SOF packet due to packet corruption.	Host, function	—
DVST	Device state transition interrupt	• When device state transition was detected USB bus reset detection Suspension detection SetAddress() request received SetConfiguration request received	Function	INTSTS0. DVSQ[2:0]
CTRT	Control transfer stage transition interrupt	• When control transfer stage transition was detected Completed setup stage Transition to control write transfer status stage Transition to control read transfer status stage Completed control transfer A control transfer sequence error occurred	Function	INTSTS0. CTSQ[2:0]
BEMP	Buffer empty interrupt (BEMP interrupt)	• The buffer was empty after all FIFO buffer data was transmitted. • A packet exceeding the maximum packet size was received.	Host, function	BEMPSTS. PIPEBEMP[9:0]
NRDY	Buffer not ready interrupt (NRDY interrupt)	In host controller operation • The STALL response from the function was received in response to the token issued. • The response from the function in response to the issued token was not received successfully (no response three times in succession or packet receive error three times in succession). • An overrun or underrun error occurred during isochronous transfer. In function controller operation • When NAK has been returned for an IN or OUT token while the PID[1:0] bits are set to 01b (BUF response) • A CRC error or bit stuffing error occurred during data reception in isochronous transfer. • An interval error occurred during data reception in isochronous transfer.	Host, function	NRDYSTS. PIPENRDY[9:0]
BRDY	Buffer ready interrupt (BRDY interrupt)	• The buffer is ready (readable or writable state).	Host, function	BRDYSTS. PIPEBRDY[9:0]
OVRCR	Overcurrent change interrupt (OVRCR interrupt)	• USBA_OVRCURA pin or USBA_OVRCURB pin state change was detected (both low to high and high to low).	Host, function	SYSSTS0. OVCMON[1:0]
BCHG	Bus change interrupt (BCHG interrupt)	• USB bus state change was detected.	Host, function	SYSSTS0. LNST[1:0]
DTCH	Device disconnection detect interrupt (DTCH interrupt)	• Disconnection of peripheral device was detected.	Host	SYSSTS0. RHST[2:0]
ATTCH	Device connection detect interrupt (ATTCH interrupt)	• The J state (continuous USB bus state for 2.5 µs) or K state (continuous USB bus state for 2.5 µs) was detected. This interrupt can be used for detecting connection of peripheral device.	Host	—
EOFERR	EOF error detect interrupt (EOFERR interrupt)	• EOF error of peripheral device was detected.	Host	—

Table 39.15 List of Interrupt Functions (2/2)

Flag That Becomes 1	Interrupt	Interrupt Source	Function Generated	Status Flag
SACK	SETUP transaction normal response interrupt (SACK interrupt)	• The SETUP transaction normal response (ACK) was received.	Host	—
SIGN	SETUP transaction error interrupt (SIGN interrupt)	• A SETUP transaction error (no response or corrupted ACK packet) was detected three times in succession.	Host	—
PDDETINT	Portable Device detection interrupt	• Portable Device connection was detected.	Host	BCCTRL. PDDETSTS
LPMEND	LPM transaction end interrupt (LPMEND interrupt)	• LPM transaction has been completed.	Host	INTSTS0. DVSQ[2:0]
L1RSMEND	L1 resume end interrupt (L1RSMEND interrupt)	• Resume processing from L1 state has been completed.	Host	—

Note 1. Though this interrupt can be generated in host controller operation, it is not usually used in host controller operation.

39.3.5.1 Setting the USBAR Interrupt Signal Output Method

The USBAR interrupt signal output method can be selected by setting the SOFCFG.INTL bit. However, since the detection method for detecting USBAR interrupts in this MCU is level sense, set the INTL bit to 1 (level output).

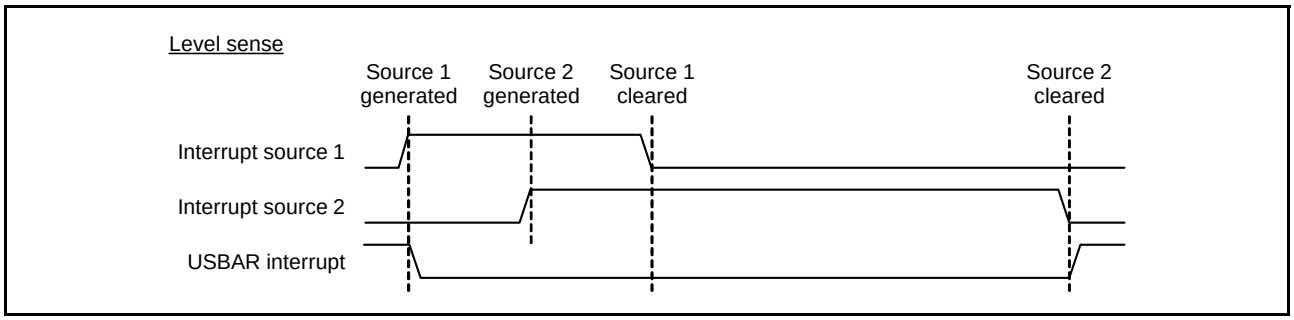


Figure 39.4 USBAR Interrupt Operation

Figure 39.5 shows the Interrupt Association Diagram.

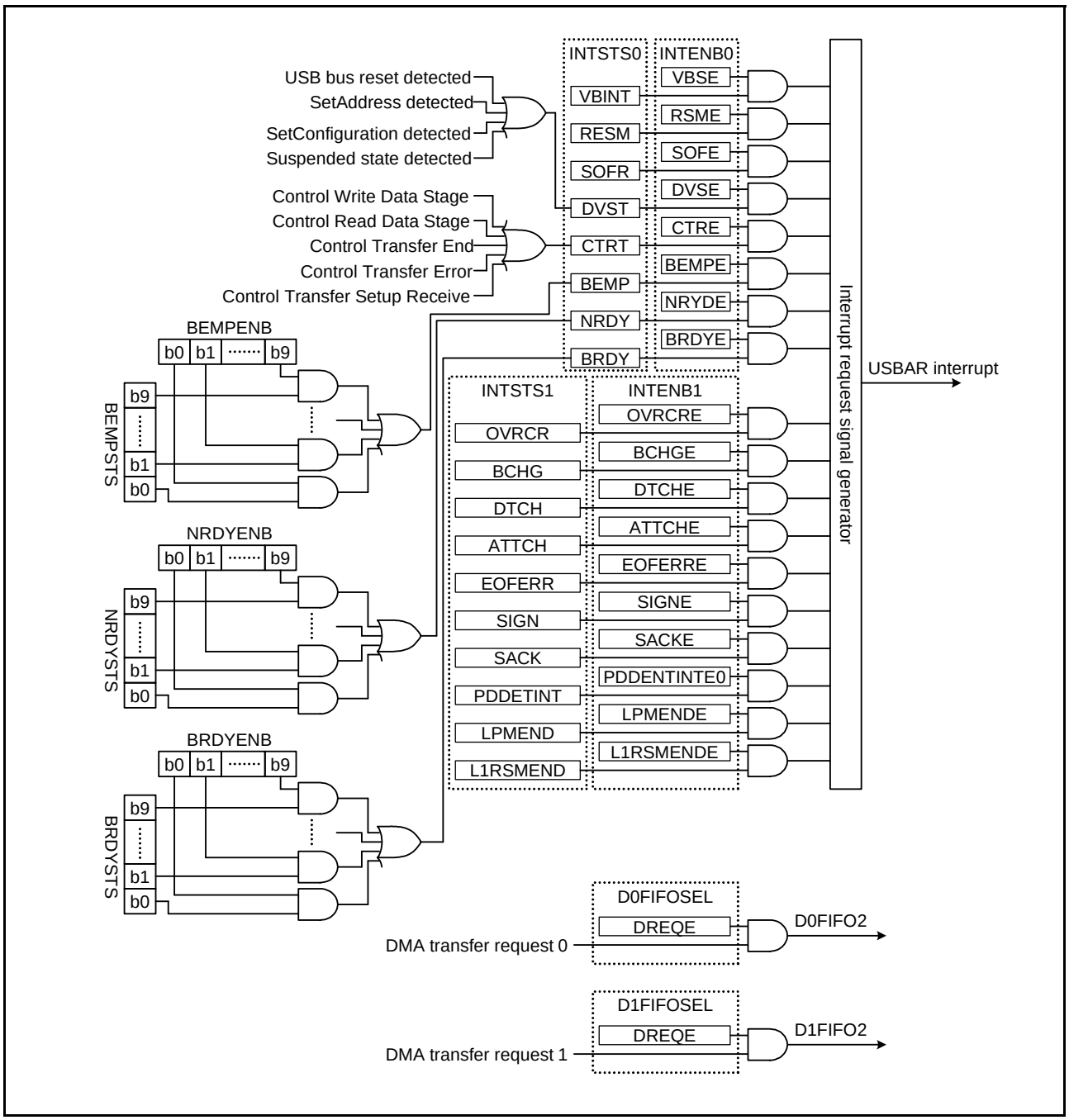


Figure 39.5 Interrupt Association Diagram

Table 39.16 lists the USBA Interrupt Sources.

Table 39.16 USBA Interrupt Sources

Interrupt	Interrupt Status Flag	DTC/DMAC Triggerable	Priority
D0FIFO2	DMA/DTC transfer request 0	Yes	High
D1FIFO2	DMA/DTC transfer request 1	Yes	↑
USBAR	VBUS interrupt, resume interrupt, frame number refresh interrupt, device state transition interrupt, control transfer stage transition interrupt, buffer empty interrupt, buffer not ready interrupt, buffer ready interrupt, OVRCR interrupt, bus change interrupt, device disconnection detection, device connection detection, EOF error detection, normal SETUP operation, SETUP error, PDDETSTS change detection, LPM transaction end interrupt, and L1 resume end interrupt	No	Low

39.3.6 Interrupt Descriptions

39.3.6.1 BRDY Interrupt

The BRDY interrupt is generated in both host controller operation and function controller operation. When each pipe satisfies the conditions listed below, the USBA sets the corresponding bit in the BRDYSTS register to 1. Under these conditions, if a bit in the BRDYENB register that corresponds to a given pipe and the INTENB0.BRDYE bit are 1, the USBA generates the BRDY interrupt request.
The conditions for generating and method for clearing the BRDY interrupt depend on the settings of the SOFCFG.BRDYM bit, and the PIPECFG.BFRE bit for each pipe as described below.

(1) When the SOFCFG.BRDYM Bit is 0 and the PIPECFG.BFRE Bit is 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible. On any of the following conditions, the USBA generates an internal BRDY interrupt request trigger and sets the BRDYSTS.PIPEBRDY[9:0] flag corresponding to the pertinent pipe to 1.

(a) For pipes set to transmit:

- When the DIR bit is changed from 0 to 1.
- When writing by the CPU to the FIFO buffer for a given pipe is disabled (when the BSTS flag is read as 0) and the USBA module has completed packet transmission.
  In continuous transfer mode, the BRDY interrupt is generated when data from one FIFO buffer is done being transmitted.
- If a FIFO buffer is specified as a double buffer, when data is done being written to one plane of the FIFO buffer, the other plane becomes empty.
- While data is being written to one plane of a FIFO buffer, even if the other plane is done transmitting, a request trigger is not generated until the data being written to the current plane is complete.
- When the hardware flushes the buffer of the pipe for isochronous transfers.
- Setting the PIPEnCTR.ACLRM bit to 1 changes the state of the FIFO buffer from write disabled to write enabled.

A request trigger is not generated for the default control pipe (during data transmission for control transfers).

(b) For pipes set to receive:

- When packet reception is completed successfully thus enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS flag is 0).
  However, no request trigger is generated for the transaction in which data PID mismatch has occurred.
  In continuous transfer mode, when the maximum packet size of data is received and there is still space in the FIFO buffer, the request trigger is not generated. When a short packet is received, the request trigger is generated even if the FIFO buffer has available space. When the transaction counter is used, the request trigger is generated on receiving the specified number of packets. In this case, the request trigger is generated even if the FIFO buffer has available space.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode. No request trigger is generated until completion of reading data from the currently-read FIFO buffer even if reception by the other FIFO buffer is completed.

In function controller operation, the BRDY interrupt is not generated in the status stage of control transfers. The PIPEBRDY interrupt status of the pertinent pipe can be set to 0 by writing 0 to the corresponding PIPEBRDY flag. In this case, 1's should be written to the corresponding bits for the other pipes. Clear the BRDY status before accessing the FIFO buffer.

(2) When the SOFCFG.BRDYM Bit is 0 and the PIPECFG.BFRE Bit is 1

With these settings, the USBA determines that a BRDY interrupt is generated on completion of reading all data for a single transfer using the pipe in the receiving direction, and sets 1 to the flag in the BRDYSTS register corresponding to the pertinent pipe.
On any of the following conditions, the USBA determines that the last data for a single transfer has been received.

- When a short packet including a zero-length packet is received.
- When the PIPEnTRN register is used and the number of packets specified by the PIPEnTRN.TRNCNT[15:0] bits are completely received.

When the pertinent data is completely read after any of the above conditions has been satisfied, the USBA determines that all data for a single transfer has been completely read. When a zero-length packet is received while the FIFO buffer is empty, the USBA determines that all data for a single transfer has been completely read when the FRDY flag in a FIFO port control register is 1 and the DTLN[11:0] flags are 000h. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding FIFO port control register. With these settings, the USBA does not detect a BRDY interrupt for the pipe in the transmitting direction. The PIPEBRDY interrupt status of the pertinent pipe can be set to 0 by writing 0 to the corresponding BRDYSTS.PIPEBRDY[9:0] flag. In this case, 1's should be written to the corresponding bits for the other pipes. In this mode, the PIPECFG.BFRE bit setting should not be modified until all data for a single transfer has been processed. When it is necessary to modify the PIPECFG.BFRE bit before completion of processing, all FIFO buffers for the pertinent pipe should be cleared using the PIPEnCTR.ACLRM bit.

(3) When the SOFCFG.BRDYM Bit is 1 and the PIPECFG.BFRE Bit is 0

With these settings, the BRDYSTS.PIPEBRDY[9:0] flag values are linked to the BSTS flag setting for each pipe. The BRDY interrupt status bits are set to 1 or 0 by the USBA depending on the FIFO buffer status.

(a) For the pipe in the transmitting direction:

The BRDY interrupt status bits are set to 1 when the FIFO port is ready for write access, and are set to 0 when it is not ready. However, the BRDY interrupt is not generated even if the default control pipe in the transmitting direction is ready for write access.

(b) For the pipe in the receiving direction:

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for read access, and are set to 0 when all data have been read (not ready for read access). When a zero-length packet is received while the FIFO buffer is empty, the pertinent bit is set to 1 and the BRDY interrupt is continuously generated until 1 is written to BCLR bit.

With this setting, the PIPEBRDY[9:0] flags cannot be set to 0 by software. When the SOFCFG.BRDYM bit is set to 1, the PIPECFG.BFRE bit (for all pipes) should be set to 0, and the SOFCFG.INTL bit should be set to 1 (level control).

Figure 39.6 shows the BRDY Interrupt Generation Timing.

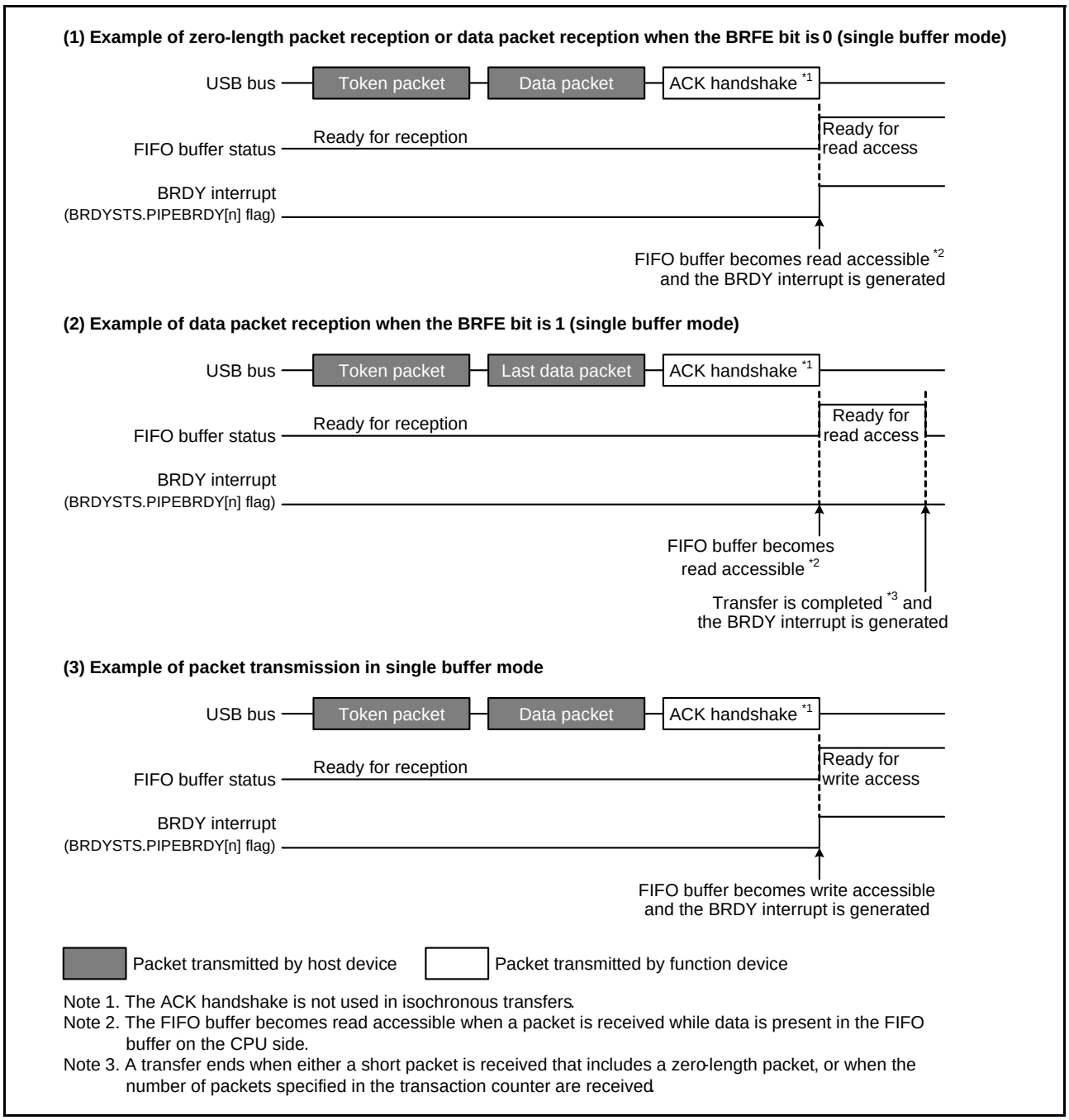


Figure 39.6 BRDY Interrupt Generation Timing

The condition that the USBA clears the INTSTS0.BRDY flag depends on the SOFCFG.BRDYM bit setting value. Table 39.17 shows the condition for clearing the BRDY flag.

Table 39.17 Condition for Clearing BRDY Flag

BRDYM Bit	Condition for Clearing the BRDY Flag
0	The USBA sets the BRDY flag to 0 when all flags in the BRDYSTS register have been set to 0 by software.
1	The USBA sets the BRDY flag to 0 when the BSTS flags for all pipes have become 0.

39.3.6.2 NRDY Interrupt

On generating an internal NRDY interrupt request for the pipe whose PID[1:0] bits are set to 01b (BUF response), the USBA sets the bit corresponding to the NRDYSTS.PIPENRDY[9:0] flag to 1. If the corresponding bit in NRDYENB has been set to 1, the USBA sets the INTSTS0.NRDY flag to 1 and generates a USBAR interrupt request.
The following describes the conditions in which the USBA generates the internal NRDY interrupt request for a given pipe.
Note that the internal NRDY interrupt request is not generated during setup transaction execution in host controller operation. During setup transactions in host controller operation, the SACK or SIGN interrupt is detected.
The internal NRDY interrupt request is not generated during status stage execution of the control transfer in function controller operation.

(1) In Host Controller Operation

(a) For the pipe in the transmitting direction:

On any of the following conditions, the USBA detects an NRDY interrupt.

- For the pipe for isochronous transfers, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer.
  In this case, the USBA transmits a zero-length packet following the OUT token and sets the bit corresponding to the NRDYSTS.PIPENRDY[9:0] flag and the FRMNUM.OVRN flag to 1.
- During communications other than setup transactions using the pipe for transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USBA sets the bit corresponding to the PIPENRDY[9:0] flag to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 00b (NAK response).
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device.
  In this case, the USBA sets the bit corresponding to the PIPENRDY[9:0] flag to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL response).

(b) For the pipe in the receiving direction:

- For the pipe for isochronous transfers, when the time to issue an IN token comes while there is no space available in the FIFO buffer.
  In this case, the USBA discards the received data for the IN token and sets the PIPENRDY[9:0] flag corresponding to the pipe and the OVRN flag to 1. When a packet error is detected in the received data for the IN token, the USBA also sets the FRMNUM.CRCE flag to 1.
- For the pipe for transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device for the IN token issued by the USBA (when timeout is detected before detection of the data packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USBA sets the PIPENRDY[9:0] flag corresponding to each pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 00b (NAK response).
- For the pipe for isochronous transfers, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the data packet from the peripheral device) or an error is detected in the packet from the peripheral device.
  In this case, the USBA sets the NRDYSTS.PIPENRDY[9:0] flag corresponding to each pipe to 1. (The setting of the PID[1:0] bits of the pipe is not modified.)
- For the pipe for isochronous transfers, when a CRC error or a bit stuffing error is detected in the received data packet.
  In this case, the USBA sets the NRDYSTS.PIPENRDY[9:0] flag corresponding to each pipe and the CRCE flag to 1.
- When the STALL handshake is received.
  In this case, the USBA sets the NRDYSTS.PIPENRDY[9:0] flag corresponding to each pipe to 1 and modifies the

setting of the PID[1:0] bits of the corresponding pipe to STALL.

(2) In Function Controller Operation

(a) For the pipe in the transmitting direction:

- When an IN token is received while there is no data to be transmitted in the FIFO buffer.
  In this case, the USBA generates an NRDY interrupt request at the reception of the IN token and sets the NRDYSTS.PIPENRDY[9:0] flag to 1. For the pipe for isochronous transfers in which an interrupt is generated, the USBA transmits a zero-length packet and sets the FRMNUM.OVRN flag to 1.

(b) For the pipe in the receiving direction:

- When an OUT token is received while there is no space available in the FIFO buffer.
  For the pipe for isochronous transfers in which an interrupt is generated, the USBA generates an NRDY interrupt request at the reception of the OUT token and sets the NRDYSTS.PIPENRDY[9:0] flag to 1 and FRMNUM.OVRN flag to 1. For the pipe for transfers other than isochronous transfers in which an interrupt is generated, the USBA generates an NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the NRDYSTS.PIPENRDY[9:0] flag to 1. However, during re-transmission (due to data PID mismatch), the NRDY interrupt request is not generated. In addition, if an error occurs in the data packet, the NRDY interrupt request is not generated.
- For the pipe for isochronous transfers, when a token is not received successfully within an interval frame.
  In this case, the USBA generates an NRDY interrupt request when SOF is received, and sets the NRDYSTS.PIPENRDY[9:0] flags to 1.

Figure 39.7 shows the NRDY Interrupt Generation Timing (In Function Controller Operation).

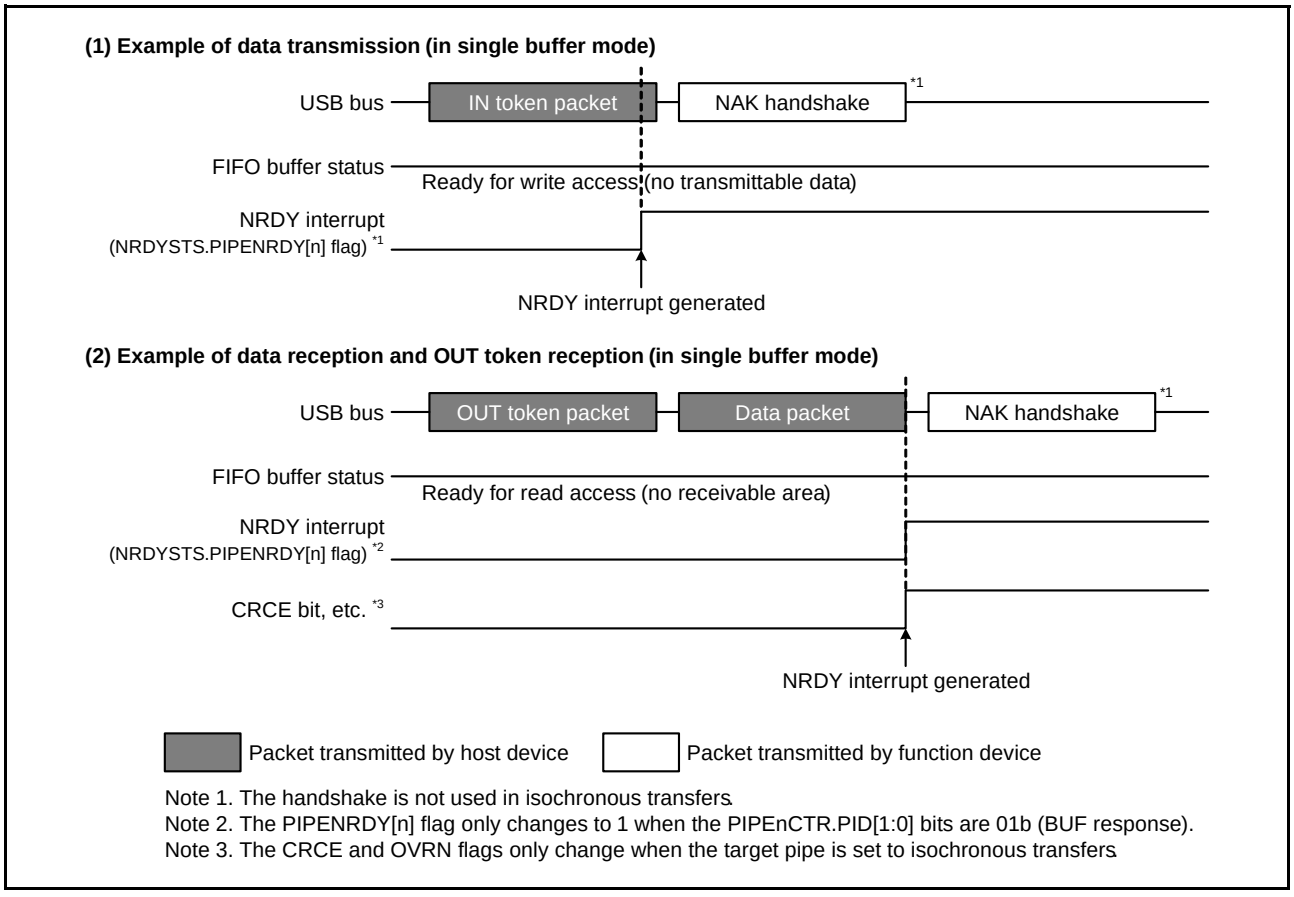


Figure 39.7 NRDY Interrupt Generation Timing (In Function Controller Operation)

39.3.6.3 BEMP Interrupt

On detecting a BEMP interrupt for the pipe whose PID[1:0] bits in the pipe control register are set to 01b (BUF response), the USBA sets the bit corresponding to the BEMPSTS.PIPEBEMP[9:0] flag to 1. If the corresponding bit in the BEMPENB register is set to 1, the USBA sets the INTSTS0.BEMP flag to 1 and generates a USBAR interrupt request.
The following describes the conditions in which the USBA generates an internal BEMP interrupt request.

(1) For the pipe in the transmitting direction:

When the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission), and during single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for the pipe other than default control pipe. However, the internal BEMP interrupt request is not generated on any of the following conditions.

- In double buffer mode, data starts to be written to the FIFO buffer on the CPU side when data transfer from one plane is complete.
- When the buffer is cleared by setting the PIPEnCTR.ACLRM or the BCLR bit in a FIFO port control register to 1.
- When IN transfer (zero-length packet transmission) is performed during the control transfer status stage during function controller operation.

(2) For the pipe in the receiving direction:

When the successfully-received data packet size exceeds the specified maximum packet size. In this case, the USBA generates a BEMP interrupt request, sets the corresponding BEMPSTS.PIPEBEMP[9:0] flag to 1, discards the received data, and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL response). Here, the USBA returns no response in host controller operation, and returns a STALL response in function controller operation. However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When a CRC error or a bit stuffing error is detected in the received data.
- When a setup transaction is being performed

Figure 39.8 shows the BEMP Interrupt Generation Timing (In Function Controller Operation).

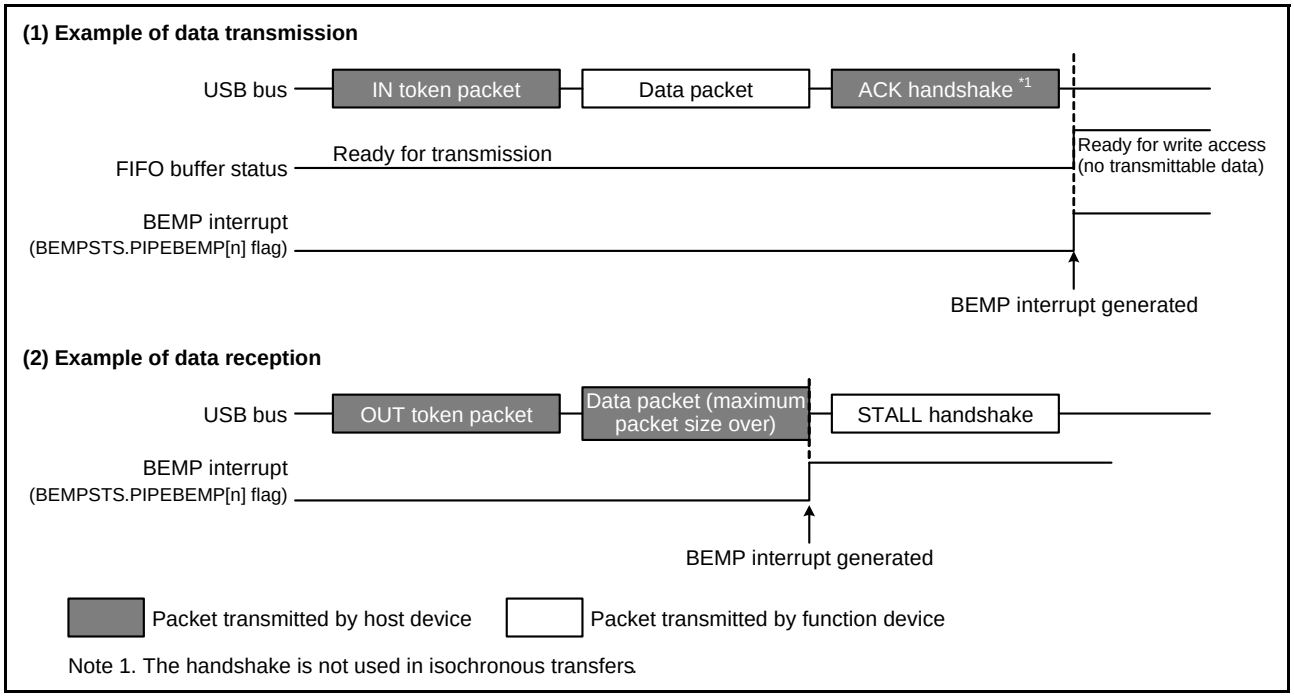
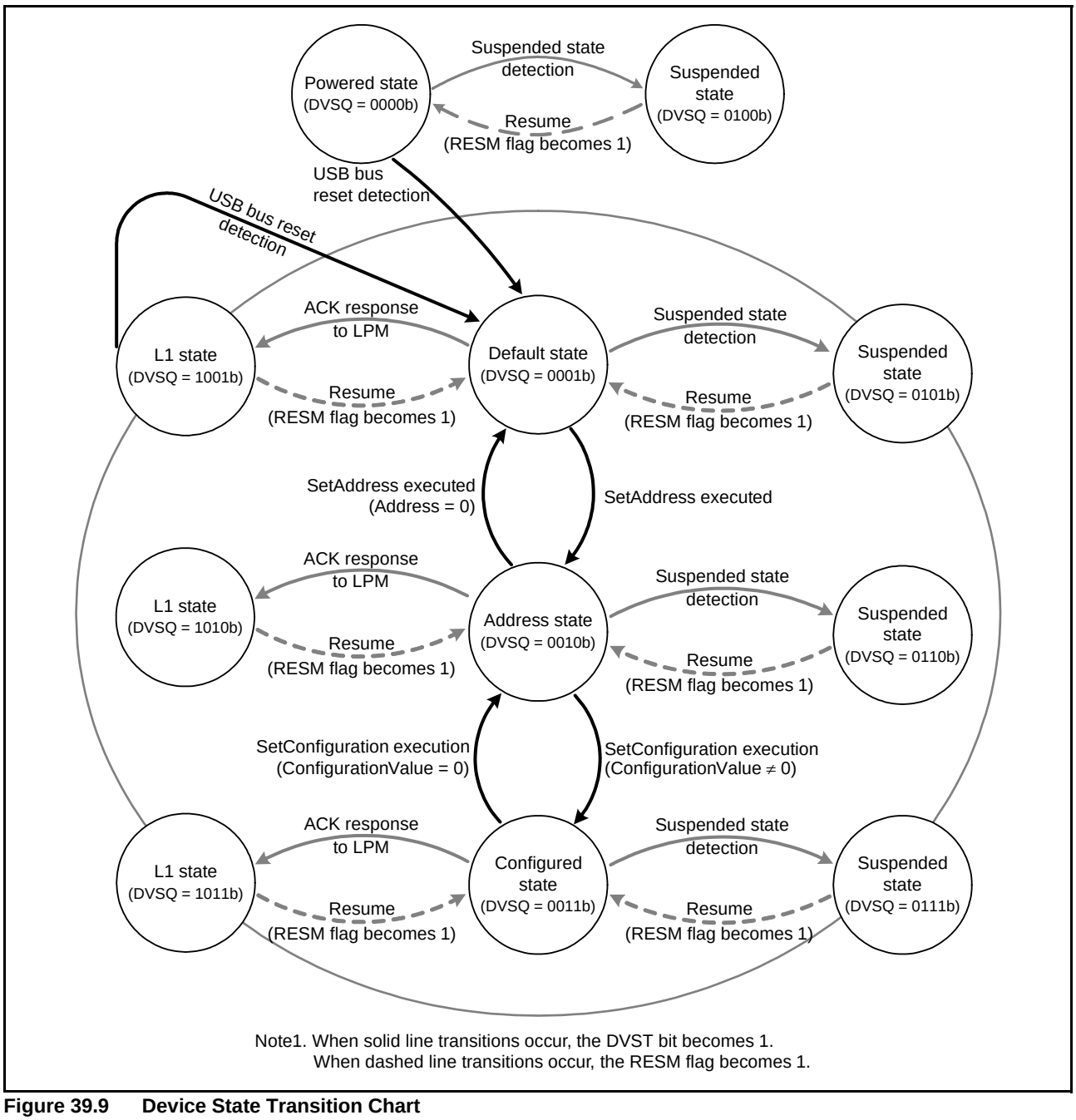


Figure 39.8 BEMP Interrupt Generation Timing (In Function Controller Operation)

39.3.6.4 Device State Transition Interrupt in Function Controller Operation

Figure 39.9 shows the Device State Transition Chart for the USBA. The USBA controls the device states and generates device state transition interrupts. However, the USBA detects a return from the suspended state (by detecting the resume signal) by the resume interrupt. Device state transition interrupts can be enabled or disabled by the INTENB0 register. The transitioned device state can be checked by the INTSTS0.DVSQ[2:0] flags. To make a transition to the default state, a device state transition interrupt is generated after a USB bus reset is detected. The USBA controls device states only in function controller operation. Device state transition interrupts are also generated only in function controller operation.



Note1. When solid line transitions occur, the DVST bit becomes 1.
When dashed line transitions occur, the RESM flag becomes 1.

Figure 39.9 Device State Transition Chart

39.3.6.5 Control Transfer Stage Transition Interrupt in Function Controller Operation

Control transfer stage transitions of the USBA are illustrated in Figure 39.10. The USBA controls the control transfer sequence and generates control transfer stage transition interrupts. Control transfer stage transition interrupts can be enabled or disabled by the INTENB0 register. The transitioned transfer stage can be checked by the INTSTS0.CTSQ[2:0] flags. Control transfer stage transition interrupts are generated only in function controller operation. Control transfer sequence errors are shown below. When an error occurs, the DCPCTR.PID[1:0] bits are set to 10b or 11b (STALL response).

(1) Control read transfer
  (a) The OUT token is received with no data transferred in response to the IN token in the data stage.
  (b) The IN token is received in the status stage.
  (c) A data packet of data PID = DATA0 is received in the status stage.
(2) Control write transfer
  (a) The IN token is received with no ACK response returned in response to the OUT token in the data stage.
  (b) The first data packet of data PID = DATA0 is received in the data stage.
  (c) The OUT token is received in the status stage.
(3) Control write no data transfer
  (a) The OUT token is received in the status stage.

If the number of pieces of receive data has exceeded the *wLength* value in the USB request in the control write transfer data stage, it is not recognized as a control transfer sequence error. When the USBA receives a packet other than the zero-length packet in the control read transfer status stage, the USBA returns an ACK response and completes the processing successfully.
When a control transfer stage transition interrupt is generated (INTSTS0.CTRT flag is 1) due to a sequence error, the value of CTSQ[2:0] flags are 110b is retained until the CTRT flag is set to 0 (interrupt status clear). Therefore, while CTSQ[2:0] bits are 110b is retained, the control transfer stage transition interrupt of setup stage completion is not generated even if a new USB request is received. (Completion of setup stage is retained by the USBA, and a control transfer stage transition interrupt is generated after the interrupt status is cleared by software.)

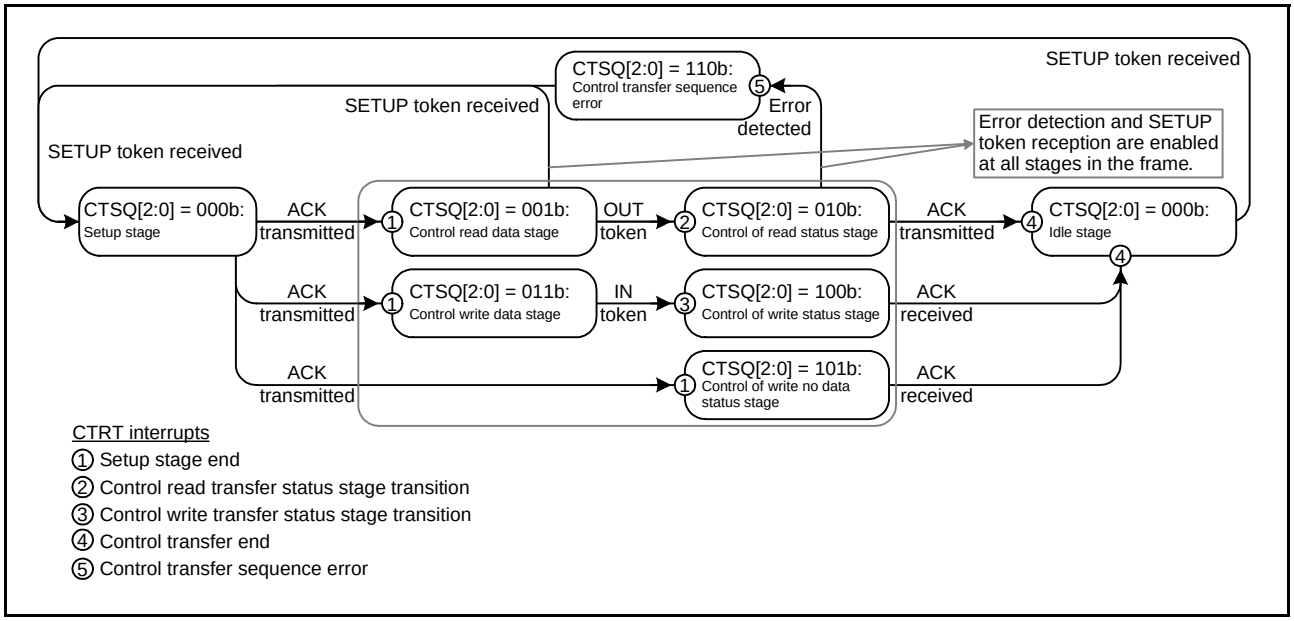


Figure 39.10 Control Transfer Stage Transition Chart

39.3.6.6 Frame Number Refresh Interrupt

In host controller operation, an interrupt is generated at the timing when the frame number is updated.
In function controller operation, a frame number refresh interrupt is generated when the frame number is updated. If the SOF cannot be received due to something like packet corruption, the frame number refresh interrupt is generated by the SOF recovery function. However, in this case the FRMNUM.FRNM[10:0] flags will not be updated.

39.3.6.7 VBUS Interrupt

When the USBA_VBUS pin level changes, a VBUS interrupt is generated. The level of the USBA_VBUS pin can be checked with the INTSTS0.VBSTS flag. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. However, if the system is activated with the host controller connected, the first VBUS interrupt is not generated because there is no change in the USBA_VBUS pin level.

39.3.6.8 Resume Interrupt

In function controller operation, a resume interrupt is generated when the device state is the suspended state and the USB bus state has changed (from J state to K state, or from J state to SE0). Recovery from the suspended state is detected by means of the resume interrupt. When the host controller is selected, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

39.3.6.9 OVRCR Interrupt

An OVRCR interrupt is generated when the USBA_OVRCURA or USBA_OVRCURB pin level has changed. The levels of the USBA_OVRCURA and USBA_OVRCURB pins can be checked with the SYSSTS0.OVCMON[1:0] flags. The external power supply IC can check whether overcurrent has been detected using the OVRCR interrupt.
For OTG connection, whether a change has been detected in the external power supply IC VBUS comparator can be checked using the OVRCR interrupt.

39.3.6.10 BCHG Interrupt

A BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether the peripheral device is connected and can also be used to detect a remote wakeup in host controller operation. The BCHG interrupt is generated regardless during host controller operation or function controller operation.

39.3.6.11 DTCH Interrupt

A DTCH interrupt is generated when disconnection of the USB bus is detected in host controller operation. The USBA detects bus disconnection based on USB 2.0.
After detecting a DTCH interrupt, the USBA controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and make a transition to the wait state for bus connection to the pertinent port (wait state for ATTCH interrupt generation).

- Modifies and indicates the DVSTCTR0.UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the Idle state.

39.3.6.12 SACK Interrupt

A SACK interrupt is generated when an ACK response for the transmitted setup packet has been received from the peripheral device in host controller operation. The SACK interrupt can be used to confirm that the setup transaction has been completed successfully.

39.3.6.13 SIGN Interrupt

A SIGN interrupt is generated when an ACK response for the transmitted setup packet has not been correctly received from the peripheral device three consecutive times in host controller operation. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

39.3.6.14 ATTCH Interrupt

An ATTCH interrupt is generated when J state or K state of the full-speed signal level is detected on the USB port for 2.5 µs in host controller operation. To be more specific, an ATTCH interrupt is detected on any of the following conditions.

- When K state, SE0, or SE1 changes to J state, and J state continues 2.5 µs.
- When J state, SE0, or SE1 changes to K state, and K state continues 2.5 µs.

39.3.6.15 EOFERR Interrupt

An EOFERR interrupt is generated when it is detected that communication is not completed at the EOF2 timing defined in USB 2.0.
After detecting an EOFERR interrupt, the USBA controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and perform re-enumeration of the pertinent port.

- Modifies and indicates the DVSTCTR0.UACT bit for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the Idle state.

39.3.6.16 Portable Device Detection Interrupt

The USBA sets the INTSTS1.PDDETINT flag to 1 on detecting a portable device and generates the portable device detection interrupt. When the portable device detection interrupt is generated, use software to repeat reading the BCCTRL.PDDETSTS flag until the same value is read three or more times, and perform debounce processing.

39.3.6.17 LPMEND Interrupt

When the LPM transaction is ended because a response from the peripheral device or a timeout is detected, the INTSTS1.LPMEND flag is set to 1 and the LPMEND interrupt is generated.

39.3.6.18 L1RSMEND Interrupt

When performing processing to exit the L1 state, when the exit processing is complete, the INTSTS1.L1RSMEND flag becomes 1, and the L1RSMEND interrupt is generated.

39.3.7 Pipe Control

Pipe setting items of the USBA are listed in Table 39.18. USB data transfer is performed through logical pipes called endpoints. The USBA is provided with 10 pipes for data transfer. Set each pipe according to the user system specifications.

Table 39.18 Pipe Setting Items

Register Symbol	Bit Name	Setting	Note
DCPCFG PIPECFG	TYPE[1:0]	Transfer type	Pipes 1 to 9: Settable
	BFRE	BRDY interrupt mode	Pipes 1 to 5: Settable
	DBLB	Selection of double buffer	Pipes 1 to 5: Settable
	CNTMD	Select continuous transfer mode or discontinuous transfer mode	Pipes 1, 2: Settable only when bulk transfer is selected Pipes 3 to 5: Settable
	DIR	Selection of transfer direction	IN or OUT settable
	EPNUM[3:0]	Endpoint number	Pipes 1 to 9: Settable Set this number to a value other than 0000b when one or more pipes are used.
	SHTNAK	Selection of pipe disable at the end of transfer	Pipes 1, 2: Settable only when bulk transfer is selected Pipes 3 to 5: Settable
PIPEBUF	BUFSIZE[4:0]	FIFO buffer size	Default control pipe: Setting disabled (fixed to 64 bytes) Pipes 1 to 5: Settable (up to 2 Kbytes specifiable) Pipes 6 to 9: Setting disabled (fixed to 64 bytes)
	BUFNMB[7:0]	FIFO buffer number	default control pipe: Setting disabled (fixed to area 00h) Pipes 1 to 5: Setting disabled (area 08h to 87h specifiable) Pipes 6 to 9: Setting disabled (fixed to area 04h to 07h respectively)
DCPMAXP PIPEMAXP	DEVSEL[3:0]	Selection of device	Viewable only in host controller operation
	MXPS	Maximum packet size	Setting compliant with USB 2.0
PIPEPERI	IFIS	Buffer flush	Pipes 1, 2: Settable only when isochronous transfer is selected Pipes 3 to 5: Setting disabled Pipes 6 to 9: Setting disabled
	IITV[2:0]	Interval counter	Pipes 1, 2: Settable only when isochronous transfer is selected Pipes 3 to 5: Setting disabled Pipes 6 to 9: Settable only in host controller operation
DCPCTR, PIPEnCTR	BSTS	Buffer status	States of receive/transmit buffers of default control pipe are switched by the ISEL bit
	INBUFM	IN buffer monitor	Provided only for pipes 1 to 5
	SUREQ	SETUP request	Settable only for the default control pipe Controllable only in host controller operation
	SUREQCLR	SUREQ bit clear	Settable only for the default control pipe Controllable only in host controller operation
	ATREPM	Auto response mode	Pipes 1 to 5: Settable only in function controller operation
	ACLRM	Auto buffer clear	Pipes 1 to 9: Settable
	SQCLR	Sequence clear	Clearing the data toggle bit
	SQSET	Sequence set	Setting the data toggle bit
	SQMON	Sequence check	Checking the data toggle bit
	PBUSY	PIPE busy check	
	PID[1:0]	Response PID	
PIPEnTRE	TRENB	Transaction count enable	Pipes 1 to 5: Settable
	TRCLR	Current transaction counter clear	Pipes 1 to 5: Settable
PIPEnTRN	TRNCNT[15:0]	Transaction counter	Pipes 1 to 5: Settable

39.3.7.1 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be modified only when USB communication is prohibited (PID[1:0] bits are 00b (NAK response)). Figure 39.11 shows pipe control register switching procedures when USB communication is enabled (PID[1:0] bits are 01b (BUF response)).

The following shows the registers and bits that should not be modified when USB communication is enabled (PID[1:0] bits are 01b (BUF response)).

- Bits in registers DCPCFG and DCPMAXP
- DCPCTR.SQCLR and SQSET bits
- Bits in the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers
- PIPEnCTR.ATREPM, ACLRM, SQCLR, and SQSET bits
- Bits in registers PIPEnTRE and PIPEnTRN
- Bits in the DEVADDm register (m = 0 to A)

For details on setting the bits in the DEVADDm register, follow the procedures of each bit described in section 39.2, Register Descriptions.

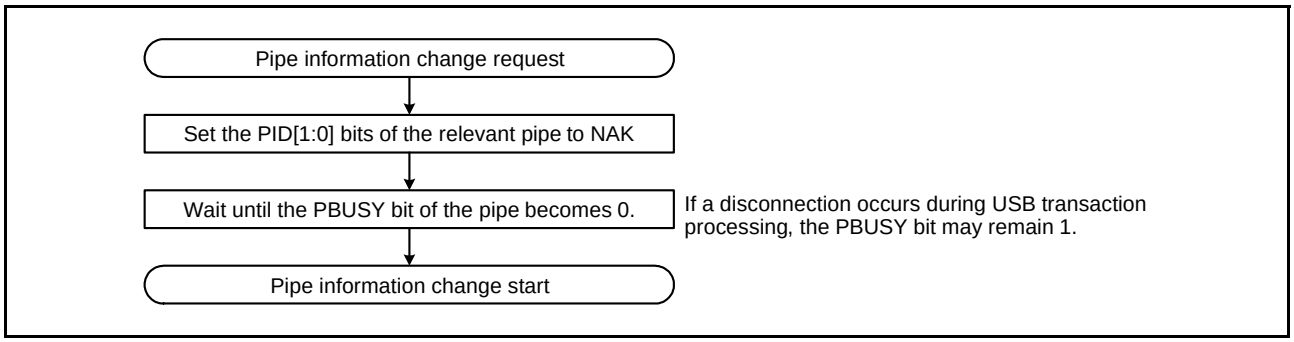


Figure 39.11 Procedure for Changing Pipe Information from the USB Communication Enabled State (PID[1:0] Bits are 01b (BUF Response))

The following bits in the pipe control registers can be modified only when the pertinent pipe information has not been set by the CURPIPE[3:0] bits in registers CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Registers that should not be set while the CURPIPE[3:0] bits are set:

- Bits in registers DCPCFG and DCPMAXP
- Bits in registers PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI
- PIPEnCTR.ACLRM bit

In order to modify pipe information, CURPIPE[3:0] bits should be set to a pipe other than the pipe to be modified. For the default control pipe, the buffer should be cleared using the BCLR bit after the pipe information is modified.

39.3.7.2 Transfer Types

The PIPECFG.TYPE[1:0] bits are used to specify the transfer type for each pipe. The transfer types that can be set for the pipes are as follows.

- Pipe 0 (default control pipe): No setting is necessary (fixed at control transfer).
- Pipes 1 and 2: These should be set to bulk transfer or isochronous transfer.
- Pipes 3 to 5: These should be set to bulk transfer.
- Pipes 6 to 9: These should be set to interrupt transfer.

39.3.7.3 Endpoint Number

The PIPECFG.EPNUM[3:0] bits are used to set the endpoint number for each pipe. The default control pipe is fixed at endpoint 0. The other pipes can be set from endpoint 1 to endpoint 15.

- Pipe 0 (default control pipe): No setting is necessary (fixed at endpoint 0).
- Pipes 1 to 9: The endpoint numbers from 1 to 15 should be selected and set.
  These should be set so that the combination of the PIPECFG.DIR bit and EPNUM[3:0] bits do not overlap.

39.3.7.4 Setting the Maximum Packet Size

Set the maximum packet size of each pipe with the DCPMAXP.MXPS[6:0] and PIPEMAXP.MXPS[10:0] bits. For the default control pipe and pipes 1 to 5, all maximum packet sizes defined in USB 2.0 can be set. For pipes 6 to 9, the maximum packet size is 64 bytes. Set the maximum packet size before starting a transfer (PID[1:0] bits are set to 01b (BUF response)).

- Default control pipe: Select 8, 16, 32, or 64.
- Pipes 1 to 5: Select 8, 16, 32, or 64 for bulk transfer.
- Pipes 1, 2: Set a value from 1 to 1023 for isochronous transfer.
- Pipes 6 to 9: Set a value of 1 to 64.

39.3.7.5 Transaction Counter (For Pipes 1 to 5 in the Reading Direction)

When the specified number of transactions has been completed in the data packet receiving direction, the USBA recognizes that the transfer has ended. Two transaction counters are provided: one is the PIPEnTRN register that specifies the number of transactions to be executed and the other is the current counter that internally counts the number of executed transactions. With the PIPECFG.SHTNAK bit set to 1, when the current counter value matches the specified number of transactions, the corresponding PIPEnCTR.PID[1:0] bits are set to 00b (NAK response) and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the PIPEnTRE.TRCLR bit. The information read from the PIPEnTRN register differs depending on the setting of the PIPEnTRE.TRENB bit.

- TRENB bit is 0: The specified transaction counter value can be read.
- TRENB bit is 1: The current counter value indicating the internally counted number of executed transactions can be read.

When operating the TRCLR bit, the following should be noted.

- If the transactions are being counted and the PIPEnCTR.PID[1:0] bits are set to 01b (BUF response), the current counter cannot be cleared.
- If there is any data left in the buffer, the current counter cannot be cleared.

39.3.7.6 Response PID

Set the response PID of each pipe by the PIPEnCTR.PID[1:0] bits DCPCTR.PID[1:0]. The following shows the USB operation with various response PID settings:

(1) Setting the response PID in host controller operation

Specify the execution of transaction for the response PID.

(a) NAK setting: Pipe prohibited state where no transaction is made.

(b) BUF setting: Transaction is made according to the FIFO buffer state.
OUT direction: The OUT token is issued when the FIFO buffer contains transmit data.
IN direction: The IN token is issued when the FIFO buffer is not full and can receive data.

(c) STALL setting: Pipe prohibited state where no transaction is made

Note: Make the default control pipe's setup transaction using the DCPCTR.SUREQ bit.

(2) Setting the response PID in function controller operation

For the response PID, specify the response to transactions from the host.

(a) NAK setting: A NAK response is always returned to generated transactions.

(b) BUF setting: A response is returned to transactions according to the FIFO buffer.

(c) STALL setting: A STALL response is always returned to generated transactions.

Note: An ACK response is always returned to the setup transaction regardless of the value of the PID[1:0] bits and the USB request is stored in the register.

The USBA may write to the PID[1:0] bits, depending on the results of the transaction as described below.

(1) USBA sets the response PID in host controller operation

(a) NAK setting: The PID[1:0] bits are set to 00b (NAK response) in the following cases, stopping issuance of tokens automatically.

- In the response to the transmitted token during a transfer other than the isochronous transfer, an NRDY interrupt occurs (refer to section 39.3.6.2 for details).
- A short packet was received during bulk transfer with the PIPECFG.SHTNAK bit set to 1.
- The transaction counter ended during bulk transfer with the PIPECFG.SHTNAK bit set to 1.

(b) BUF setting: BUF is not written by the USBA.

(c) STALL setting: The PID[1:0] bits are set to STALL in the following cases, stopping issuance of tokens automatically.

- A STALL response was received in response to the transmitted token.
- A data packet exceeding the maximum packet size was received.

(2) USBA sets the response PID in function controller operation

(a) NAK setting:

- The SETUP token was successfully received (only default control pipe).
- The PIPECFG.SHTNAK bit was set to 1 during the bulk transfer and the transaction counter ended, or a short packet was received.

(b) BUF setting: BUF is not written by the USBA.

(c) STALL setting:

- A maximum packet size over error was detected in the receive data packet.
- A control transfer sequence error was detected.

39.3.7.7 Data PID Sequence Bit

When data is successfully transferred in the data stage, bulk transfer, or interrupt transfer of the control transfer, the sequence bit of the data PID is automatically toggled by the USBA. The sequence bit of the data PID transmitted next can be checked by the SQMON flag in DCPCTR and PIPEnCTR. The sequence bit changes upon receiving the ACK handshake during data transmission, or upon transmitting the ACK handshake during data reception. Furthermore, the data PID sequence bit can be modified by the SQCLR and SQSET bits in DCPCTR and PIPEnCTR.
In the control transfer during function controller operation, the USBA automatically sets the sequence bit at the stage transition. The data PID becomes DATA1 at the end of the setup stage. The sequence bit is not accessed in the status stage, and a response is returned with PID = DATA1. Therefore, sequence bit setting by software is not necessary. In the control transfer during host controller operation, the sequence bit must be set by software at the stage transition.
Note that the data PID sequence bit must be set by software when transmitting or receiving the ClearFeature() request regardless during the host or function controller operation.

39.3.7.8 Response PID = NAK Function

The USB has a function that disables pipe operation (PID[1:0] bits are set to 00b (NAK response)) at the timing at which the final data packet of a transaction is received (the USB automatically distinguishes this based on reception of a short packet or the transaction counter) by setting the PIPECFG.SHTNAK bit to 1. When the double buffer mode is being used for the FIFO buffer, using this function enables reception of data packets in transfer units. If pipe operation has been disabled, the pipe should be set to the enabled state again (PID[1:0] bits are set to 01b (BUF response)) by software. The response PID = NAK function can be used only when bulk transfers are used.

39.3.7.9 Auto Response Mode

With the pipes for bulk transfer (pipe 1 to pipe 5), when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode. During an OUT transfer (PIPECFG.DIR bit is 0), OUT-NAK mode is entered, and during an IN transfer (DIR bit is 1), null auto response mode is entered.

39.3.7.10 OUT-NAK Mode

With the pipes for bulk OUT transfer, NAK is returned in response to an OUT token and an NRDY interrupt is output when the PIPEnCTR.ATREPM bit is set to 1. To make a transition from normal mode to OUT-NAK mode, OUT-NAK mode should be specified in the pipe operation disabled state (PID[1:0] bits are set to 00b (NAK response)) before enabling pipe operation (PID[1:0] bits are set 01b (BUF response)). After pipe operation has been enabled, OUT-NAK mode becomes valid. However, if an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is retuned to the host. To transition from OUT-NAK mode to normal mode, OUT-NAK mode should be canceled in the pipe operation disabled state (PID[1:0] bits are set to 00b (NAK response)) before enabling pipe operation (PID[1:0] bits are set to 01b (BUF response)). In normal mode, reception of OUT data is enabled.

39.3.7.11 Null Auto Response Mode

With the pipes for bulk IN transfer, zero-length packets are continuously transmitted when the PIPEnCTR.ATREPM bit is set to 1. To make a transition from normal mode to null auto response mode, null auto response mode should be set in the pipe operation disabled state (PID[1:0] bits are set to 00b (NAK response)) before enabling pipe operation (PID[1:0] bits are set to 01b (BUF response)). After pipe operation has been enabled, null auto response mode becomes valid. Before setting null auto response mode, confirm that the PIPEnCTR.INBUFM flag is 0 because the mode can be set only when the buffer is empty. If the INBUFM flag is 1, the buffer should be emptied with the PIPEnCTR.ACLRM bit. While a transition to null auto response mode is being made, data should not be written from the FIFO port. To make a transition from null auto response mode to normal mode, pipe operation disabled state (PID[1:0] bits are set to 00b (NAK response)) should be retained for the period of zero-length packet transmission (about 10 µs) before canceling null auto response mode. In normal mode, data can be written from the FIFO port; therefore, packet transmission to the host is enabled by enabling pipe operation (PID[1:0] bits are set to 01b (BUF response)).

39.3.8 FIFO Buffers

39.3.8.1 FIFO Buffers

The USBA incorporates FIFO buffers for data transfer. The area used for each pipe is managed by the USBA. Access rights to FIFO buffers can be assigned to the system (CPU side) or the USBA (SIE side).

(1) Status of the FIFO Buffers

Table 39.19 and Table 39.20 list the states of the FIFO buffers in the USBA. The state of the FIFO buffers can be confirmed by reading the DCPCTR.BSTS flag and PIPEnCTR.INBUFM flag. The transfer direction for the FIFO buffers can be specified using either the PIPECFG.DIR bit or the CFIFOSEL.ISEL bit (when default control pipe is selected). The INBUFM flag is valid for pipe 1 to pipe 5 when transmitting. When a transfer pipe on the transmitting side is set to double buffer mode, the BSTS flag is used to monitor the buffer status on the CPU side, and the INBUFM flag is used to monitor the buffer status on the SIE side. When the CPU or DMAC/DTC write access to the FIFO port is slow, and the buffer empty status cannot be determined by the BEMP interrupt, the end of transmission can be confirmed by the INBUFM flag.

Table 39.19 FIFO Buffer Status Indicated by the BSTS Flag

ISEL or DIR Bit	BSTS Flag	FIFO Buffer Status
0 (data reception)	0	There is no received data, or data is being received. Data cannot be read from the FIFO port.
0 (data reception)	1	There is received data, or a zero-length packet has been received. Data can be reading from the FIFO port. Note that when a zero-length packet is received, data cannot be read and the FIFO buffer must be cleared.
1 (data transmission)	0	Data transmission is not complete. Data cannot be written to the FIFO port.
1 (data transmission)	1	Data transmission has been completed. Data can be written to the FIFO buffer.

Table 39.20 Buffer Status Indicated by the INBUFM Flag

DIR Bit	INBUFM Flag	FIFO Buffer Status
0 (data reception)	Invalid	Invalid
1 (data transmission)	0	Data transmission has been completed. There is no data to be transmitted, or transmit data is being written to the FIFO port.
1 (data transmission)	1	Transmittable data is present.

39.3.8.2 Clearing the FIFO Buffers

Table 39.21 lists information pertaining to clearing FIFO buffers by the USBA. FIFO buffers can be cleared using the BCLR bit in a FIFO port control register, DCLRM bit in the D0FIFOSEL/D1FIFOSEL register, and PIPEnCTR.ACLRM bit. Pipes 1 to 5 can be set to single buffer mode or double buffer mode using the PIPECFG.DBLB bit.

Table 39.21 Methods of Clearing the FIFO Buffers

Cleared	FIFO Buffers on the CPU Side	FIFO Buffer That Has Had All Transmit Data Read	All Data in the FIFO Buffer Allocated to Pipe n
Bit used	CFIFOCTR.BCLR bit D0FIFOCTR.BCLR bit D1FIFOCTR.BCLR bit	D0FIFOSEL.DCLRM bit D1FIFOSEL.DCLRM bit	PIPEnCTR.ACLRM bit
Operation	When the BCLR bit is set to 1, the FIFO buffers on the CPU side of the pipes specified by the CFIFOSEL.CURPIPE[3:0] bits and CURPIPE[3:0] bits in the D0FIFOSEL/D1FIFOSEL register are cleared.	If the DCLRM bit is set to 1, after reading the received data for the pipe specified by the CURPIPE[3:0] bits in the D0FIFOSEL/D1FIFOSEL register, the FIFO buffer is automatically cleared.	If the ACLRM bit is set to 1 and then set to 0, all data in the FIFO buffer (both planes in double buffer mode) allocated to pipe n is cleared. Also, If the PID[1:0] bits are set to 01b (BUF response) while the ACLRM bit is still set to 1, all packets received are discarded.

(1) Received Packet Read-and-Discard Function in Auto Buffer Clear Mode

If the PIPEnCTR.ACLRM bit is set to 1 and the PID[1:0] bits are set to 01b (BUF response), the USBA discards all received data packets. However, if a normal data packet is received, the ACK response is returned to the host controller. The received packet read-and-discard function can only be set to FIFO buffers that are reading data. In addition, do not use pipes that are specified as isochronous transfer or interrupt transfer.

39.3.8.3 FIFO Port Functions

Table 39.22 shows the settings for the functions of the FIFO ports in the USBA. When writing data to a FIFO buffer, data is written until the maximum packet size is reached, and the FIFO buffer automatically becomes enabled to transmit data. To enable transmission before the maximum packet size is reached, set the BVAL bit in a FIFO port control register to end the write access. To transmit a zero-length packet, after using the BCLR bit to clear the FIFO buffer, the BVAL bit must be set to end the write access.
When reading data from a FIFO buffer, after all data has been read from the FIFO buffer, it automatically becomes ready to receive new packets. However, when a zero-length packet is received (DTLN[11:0] flags are 000h), data cannot be read from the FIFO buffer, and the BCLR bit must be used to clear the buffer. The length of the receive data can be confirmed using the DTLN[11:0] flags in a FIFO port control register.

Table 39.22 FIFO Port Function Settings

Register Name	Bit Name	Description
CFIFOSEL, D0FIFOSEL, D1FIFOSEL	RCNT	Selects DTLN[11:0] read mode.
	REW	FIFO buffer rewind (reread, rewrite).
	DCLRM	Automatically clears receive data for a current pipe after the data has been read (only for D0FIFO and D1FIFO).
	DREQE	Enables DMA/DTC transfers (only for D0FIFO and D1FIFO).
	MBW[1:0]	FIFO port access bit width.
	BIGEND	Selects FIFO port endian.
	ISEL	FIFO port access direction (only for the default control pipe).
	CURPIPE[3:0]	Selects the current pipe.
CFIFOCTR, D0FIFOCTR, D1FIFOCTR	BVAL	Ends writing to the FIFO buffer.
	BCLR	Clears FIFO buffers on the CPU side.
	DTLN[11:0]	Checks the length of the receive data.

(1) FIFO Port Selection

Table 39.23 shows the pipes that can be selected with the various FIFO ports. Use the CURPIPE[3:0] bits in a FIFO port select register to select the pipe to be accessed. After the pipe is selected, check to see that the value written to the CURPIPE[3:0] bits can be read correctly, (if the previous pipe number is read, it indicates the USBA is modifying the pipe). Then, check to see if the FRDY flag in a FIFO port control register is 1, and access the FIFO buffers. In addition, use the MBW[1:0] bits in a FIFO port select register to select the bus width to be accessed. The PIPECFG.DIR bit is used to select the FIFO buffer access direction, and the ISEL bit in a FIFO port select register determines the direction of the default control pipe.

Table 39.23 FIFO Port Access by Pipe

Pipe	Access Method	Port that can be Used
default control pipe	CPU access	CFIFO port register
Pipe 1 to pipe 9	CPU access	CFIFO port register D0FIFO/D1FIFO port register
	DMAC/DTC access	D0FIFO/D1FIFO port register

(2) REW Bit

The pipe currently being accessed can be halted, another pipe can be accessed, and then it is possible to continue processing on the current pipe. Set the REW bit in a FIFO port select register to perform this. If a pipe is selected by setting the CURPIPE[3:0] bits in a FIFO port select register and the REW bit to 1 simultaneously, the pointer used to read from and write to the FIFO buffer is reset, and the read or write access can be carried out from the first byte. If a pipe is selected by setting the REW bit to 0, the pointer used to read from and write to the FIFO buffer is not reset, and the data access previously selected can be continued. To access a FIFO port, after selecting a pipe, check to see if an FRDY flag in a FIFO port control register is 1.

39.3.8.4 DMA/DTC Transfers (D0FIFO and D1FIFO Ports)

(1) Overview of DMA/DTC Transfers

The FIFO port can be accessed using the DMAC/DTC for pipes 1 to 9. When access to the buffer for the pipe targeted for DMA/DTC transfer is enabled, a DMA/DTC transfer request is issued.
Set the MBW[1:0] bits in the D0FIFOSEL/D1FIFOSEL register to select the transfer width to the FIFO port, set the CURPIPE[3:0] bits to select the pipe targeted for DMA/DTC transfer. Do not change the selected pipe during DMA/DTC transfer.

(2) D0FIFO/D1FIFO Auto Clear Mode (Reading Ports D0FIFO and D1FIFO)

When the DCLRM bit in the D0FIFOSEL/D1FIFOSEL register is set to 1, if the FIFO buffer read access is completed, the USBA automatically clears the FIFO buffer of the selected pipe.
In Table 39.24, the relationship between packet reception and FIFO buffer clearing processing by software is shown for various bit settings. As shown in Table 39.24, the conditions for clearing a FIFO buffer depends on the PIPECFG.BFRE bit setting. Using the DCLRM bit eliminates the need for a FIFO buffer to be cleared by software in any situation that requires FIFO buffer clearing. This enables DMA/DTC transfers without involving software.
D0FIFO/D1FIFO auto clear mode can be used when reading a FIFO buffer.

Table 39.24 Relationship Between Packet Reception and FIFO Buffer Clearing Processing by Software

Buffer Status When a Packet is Received	Bit Setting			
	DCLRM bit is 0		DCLRM bit is 1	
	BFRE bit is 0	BFRE bit is 1	BFRE bit is 0	BFRE bit is 1
Buffer full	Clearing not necessary	Clearing not necessary	Clearing not necessary	Clearing not necessary
Zero-length packet reception	Clearing necessary	Clearing necessary	Clearing not necessary	Clearing not necessary
Normal short packet reception	Clearing not necessary	Clearing necessary	Clearing not necessary	Clearing not necessary
Transaction count end	Clearing not necessary	Clearing necessary	Clearing not necessary	Clearing not necessary

39.3.8.5 Allocating the FIFO Buffers

Figure 39.12 is an example of the memory map for a USBA FIFO buffer. The FIFO buffer is an area shared by the CPU and the USBA. Access rights to the FIFO buffers can be given to the CPU side or USBA side (SIE side).

An independent area is set for the FIFO buffer of each pipe. A memory area is set by the first block number and the number of blocks (specified by the PIPEBUF.BUFSIZE[4:0] and BUFNMB[7:0] bits), regarding 64 bytes as one block. When the PIPECFG.CNTMD bit is 1 (continuous transfer mode), set the BUFSIZE[4:0] bits to an integral multiple of the maximum packet size. In addition, when the PIPECFG.DBLB bit is set to 1 (double buffer mode), twice the memory area specified by the PIPEBUF.BUFSIZE[4:0] bits is allocated to the same pipe.

Three FIFO ports are used to access (read data from and write data to) a FIFO buffer. Set the CFIFOSEL/D0FIFOSEL/D1FIFOSEL.CURPIPE[3:0] bits to specify the pipe the FIFO port is allocated to.

The BSTS flags in registers DCPCTR and PIPEnCTR, and the PIPEnCTR.INBUFM flag can be used to check the status of a FIFO buffer for each pipe. Use the CFIFOCTR/D0FIFOCTR/D1FIFOCTR.FRDY flag to check access rights for a FIFO port.

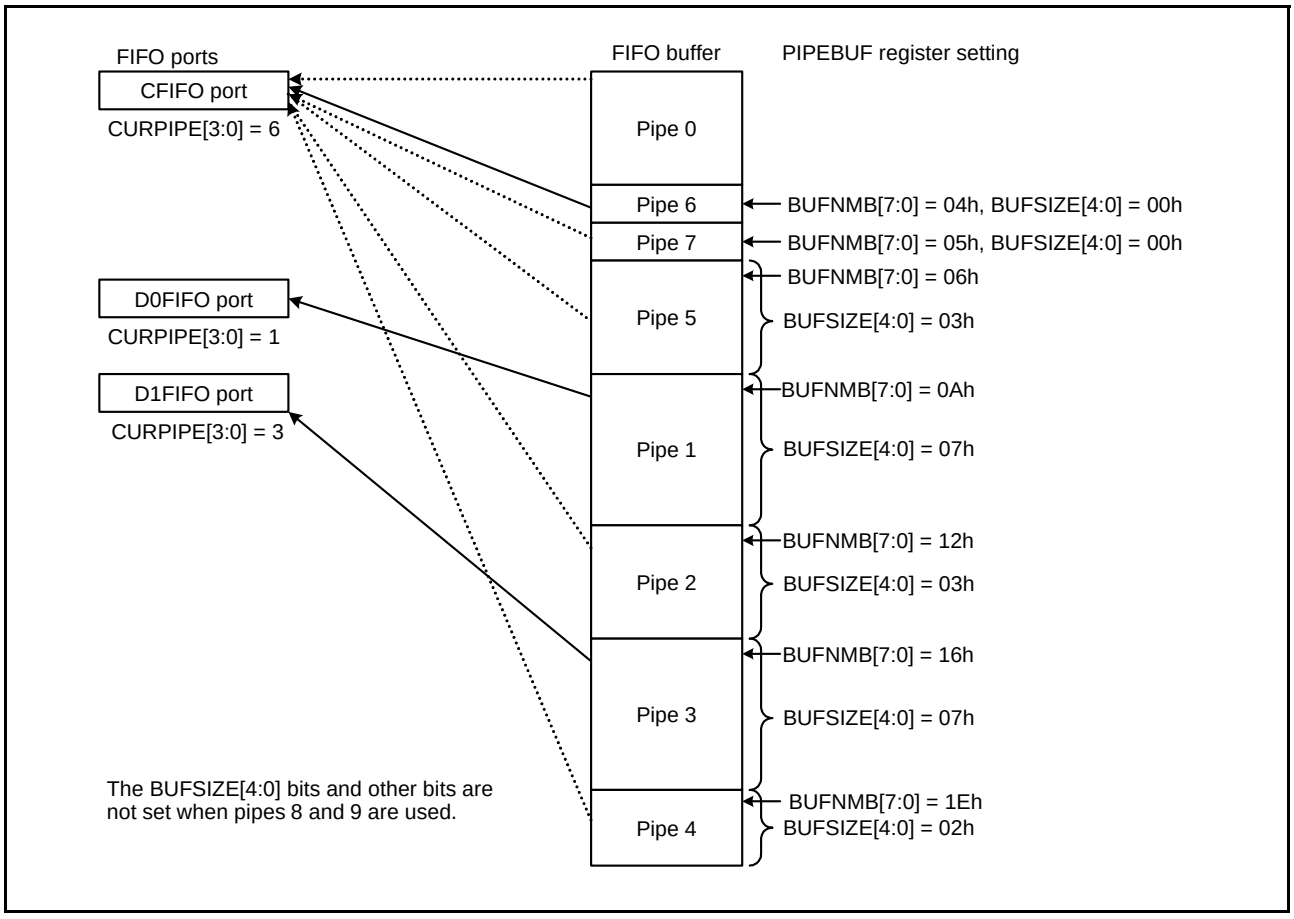


Figure 39.12 Example of a FIFO Buffer Memory Map

39.3.9 Control Transfer Using the Default Control Pipe

The default control pipe is used for data transfer in the control transfer data stage. The default control pipe's FIFO buffer is a 64-byte single buffer of the fixed area for both control read and control write. This FIFO buffer can only be accessed through the CFIFO port.

39.3.9.1 Control Transfer in Host Controller Operation

(1) Setup Stage

Registers USBREQ, USBVAL, USBINDX, and USBLENG are used to transmit the USB request of the SETUP transaction. Writing the setup packet data to the register and then setting the DCPCTR.SUREQ bit to 1 transmits the set data as the SETUP transaction. The SUREQ bit becomes 0 when the transaction is completed. Do not modify the above USB request registers while the SUREQ bit is 1. After the USBA detects that the peripheral device is connected, issue the first SETUP transaction for the device using the sequence described above by setting the DCPMAXP.DEVSEL[3:0] bits to 0 and setting the DEVADD0.USBSPD[1:0] bits. After the connected peripheral device enters the Address state, issue SETUP transactions using the sequence described above with the assigned USBAddress set in the DEVSEL[3:0] bits and the bits in the DEVADDm register corresponding to the specified USBAddress set appropriately (m = 0 to A). For example, set the DEVADD2 register when the PIPEMAXP.DEVSEL[3:0] bits are 0010b, and set the DEVADD5 register when the PIPEMAXP.DEVSEL[3:0] bits are 0101b. When a transaction is transmitted, an interrupt request is generated by the response from the peripheral device (INTSTS1.SIGN and SACK flags). This interrupt request allows the USBA to check the SETUP transaction result. The DATA0 data packet (USB request) of the SETUP transaction is transmitted regardless of the DCPCTR.SQMON flag setting.

(2) Data Stage

The data stage is used to transfer data using the default control pipe's FIFO buffer.
Before accessing the default control pipe's FIFO buffer, set the CFIFOSEL.ISEL bit to specify the access direction. In addition, set the DCPCFG.DIR bit to specify the transfer direction. The first data packet in the data stage must be transferred with the data PID set to DATA1. Transactions can be executed by setting the data PID to DATA1 using the DCPCTR.SQSET bit and setting the PID[1:0] bits to 01b (BUF response). Completion of data transfer is detected by the BRDY interrupt or the BEMP interrupt. Data transfer of multiple packets is enabled in continuos transfer mode. If the transmit data size is an integral multiple of the maximum packet size, control the control write transfer by software to transmit a zero-length packet last.

(3) Status Stage

The status stage is used for zero-length packet data transfer in the reverse direction of the data stage. As in the data stage, data transfer using the default control pipe's FIFO buffer is performed in the status stage. Transactions are executed using the same procedure as the data stage. Data packets in the status stage must be transmitted and received with the data PID set to DATA1. Set the data PID to DATA1 by the DCPCTR.SQSET bit. When a zero-length packet is received, check the receive-data length by the CFIFOCTR.DTLN[11:0] flags after the BRDY interrupt is generated, and then clear the FIFO buffer by using the BCLR bit.

39.3.9.2 Control Transfer in function controller operation

(1) Setup Stage

The USBA returns an ACK response in response to a normal setup packet for the USBA. The following describes the USBA operation in the setup stage.
The USBA sets the following bits upon receiving a new setup packet.

- Sets the INTSTS0.VALID flag to 1.
- Sets the DCPCTR.PID[1:0] bits to 00b (NAK response).
- Sets the DCPCTR.CCPL bit to 0.

When the USBA receives a data packet following a setup packet, the USBA stores the USB request parameters in registers USBREQ, USBVAL, USBINDX, and USBLENG.

Before performing the response processing for the control transfer, be sure to set VALID flag to 0. While VALID flag is 1, the PID[1:0] bits cannot be set to 01b (BUF response) and the data stage cannot be finished. With the VALID flag function, the USBA can suspend the ongoing request processing upon receiving a new USB request during the control transfer and return a response to the latest request.

Furthermore, the USBA automatically checks the direction bit (bit 8 of *bmRequestType*), and the request data length (*wLength*) in the received USB request, and recognizes the control read transfer, control write transfer, and control write no data transfer to control stage transitions. In the case of an incorrect sequence, a sequence error of the control transfer stage transition interrupt occurs, and interrupt is reported to software. For details on stage control by the USBA, see Figure 39.10, Control Transfer Stage Transition Chart.

(2) Data Stage

Perform data transfer by the default control pipe in response to the received USB request. Before accessing the default control pipe's FIFO buffer, set the CFIFOSEL.ISEL bit to specify the access direction. A transaction is executed when the DCPCTR.PID[1:0] bits are set to 10b (BUF response). Completion of data transfer is detected by the BRDY interrupt or BEMP interrupt. Use the BRDY interrupt for control write transfers, and use the BEMP interrupt for control read transfers.

(3) Status Stage

While the DCPCTR.PID[1:0] bits are set to 01b (BUF response), control transfer can be finished by setting the DCPCTR.CCPL bit to 1.
After the DCPCTR.CCPL bit is set to 1, the USBA automatically executes the status stage according to the data transfer direction determined in the setup stage. The status stage is executed specifically as follows.

- For control read transfer:

The USBA receives a zero-length packet from the USB host and transmits an ACK response.

- For control write transfer and no data control transfer:

The USBA transmits a zero-length packet and receives an ACK response from the USB host.

(4) Control Transfer Auto Response Function

The USBA automatically responds to a normal SetAddress() request. If the SetAddress() request contains any of the following errors, a response must be returned by software.

- When *bmRequestType* is not 00h: Except control write transfer
- When the *wIndex* value is not 00h: Request error
- When the *wLength* value is not 00h: Except no data control transfer
- When the *wValue* value is larger than 7Fh: Request error
- When INTSTS0.DVSQ[2:0] flags are 011b (configured state): control transfer of device state error

To all requests other than SetAddress(), a response by the corresponding software is required.

39.3.10 Bulk Transfer for Pipes 1 to 5

The FIFO buffer usage (setting of single buffer/double buffer or continuous transfer mode/discontinuous transfer mode) is selectable for the bulk transfer. Up to 2 Kbytes of the FIFO buffer size can be set.

39.3.11 Interrupt Transfer for Pipes 6 to 9

In function controller operation, the USBA performs the interrupt transfer according to the cycle managed by the USB host.
In host controller operation, the token issuing timing can be set by the interval counter.

39.3.11.1 Interval Counter for Interrupt Transfer in Host Controller Operation

To perform the interrupt transfer, specify the transaction interval for the PIPEPERI.IITV[2:0] bits. The USBA issues interrupt transfer tokens according to the set interval.

(1) Initializing the Counter

When the MCU is reset, or when the PIPEnCTR.ACLRM bit is set to 1, the interval counter is initialized. Note that the PIPEPERI.IITV[2:0] bits are not initialized when the ACLRM bit is initialized.

Note that the interval counter is not initialized in the following case.

- USB bus reset, USB suspend
  The IITV[2:0] bits are not initialized. When the DVSTCTR0.UACT bit is set to 1, the count continues from the value before entering the USB bus reset state or the USB suspend state.

(2) Operation when Tokens Cannot Be Transmitted or Received at the Token Generation Timing

No token is generated in the following cases even at the token generation timing. In this case, try to execute a transaction in the next interval.

- The PID[1:0] bits are set to NAK response or STALL response.
- No available area is left in the FIFO buffer at the token transmit timing during transfer in the IN (receiving) direction.
- The FIFO buffer contains no transmit data at the token transmit timing during transfer in the OUT (transmitting) direction.

39.3.12 Isochronous Transfer for Pipes 1 and 2

The USBA has the following functions for isochronous transfers.

- Notification of isochronous transfer error
- Interval counter (specified by the PIPEPERI.IITV[2:0] bits)
- Isochronous IN transfer data setup control (IDLE function)
- Isochronous IN transfer buffer flush function (specified by the PIPEPERI.IFIS bit)

39.3.12.1 Error Detection in Isochronous Transfers

The USBA has a function for detecting the following error information for software to manage isochronous transfer errors. Table 39.25 and Table 39.26 show the priority in which errors are detected by the USBA and the interrupts generated corresponding to errors.

(1) PID errors
- The PID value of the received packet is invalid.

(2) CRC errors and bit stuffing errors
- A CRC error was found in the received packet or the bit stuffing is incorrect.

(3) Maximum packet size over
- The data size of the received packet exceeded the maximum packet size.

(4) Overrun and underrun errors
- In host controller operation:
  No available area is left in the FIFO buffer at the token transmit timing during transfer in the IN (receiving) direction.
  The FIFO buffer contains no transmit data at the token transmit timing during transfer in the OUT (transmitting) direction.
- In function controller operation:
  The FIFO buffer contains no data when an IN token is received during transfer in the IN (transmitting) direction.
  No available area is left in the FIFO buffer when an OUT token is received during transfer in the OUT (receiving) direction.

(5) Interval error
In function controller operation, the following cases are treated as an interval error.
- Failed to receive an IN token in the interval frame during the isochronous IN transfer.
- Failed to receive an OUT token in the interval frame during the isochronous OUT transfer.

Table 39.25 Detecting an Error When a Token is Issued or Received

Detection Priority	Error Type	Interrupt Generated at Error Detection and Status
1	PID error	No interrupt is generated regardless of the selected controller function. (Ignored as a corrupted packet.)
2	CRC error, bit stuffing error	No interrupt is generated regardless of the selected controller function. (Ignored as a corrupted packet.)
3	Overrun error, underrun error	An NRDY interrupt is generated and the OVRN flag is set to 1 regardless of the selected controller function. In function controller operation, a zero-length packet is transmitted in response to the IN token. No data packet is received in response to the OUT token.
4	Interval error	An NRDY interrupt is generated in function controller operation, but no interrupt is generated in host controller operation.

Table 39.26 Detecting an Error When Receiving a Data Packet

Detection Priority	Error Type	Interrupt Generated at Error Detection and Status
1	PID error	No interrupt is generated (ignored as a corrupted packet.)
2	CRC error, bit stuffing error	An NRDY interrupt is generated and the CRCE flag is set to 1 regardless of the selected controller function.
3	Maximum packet size over error	A BEMP interrupt is generated and the PID[1:0] bits are set to STALL response regardless of the selected controller function.

39.3.12.2 Data PID

The following describes actions to be taken for the received token in function controller operation.

(1) IN token:
- DATA0: Transmitted as PID of the data packet.
- DATA1: Not transmitted.

(2) OUT token:
- DATA0: Successfully received as PID of the data packet.
- DATA1: Successfully received as PID of the data packet.

39.3.12.3 Interval Counter

The isochronous transfer interval can be set by the PIPEPERI.IITV[2:0] bits. In function controller operation, the interval counter enables functions shown in Table 39.27 below. In host controller operation, the token issuing timing is generated. In host controller operation, the interval counter operates in the same manner as in the interrupt transfer.

Table 39.27 Interval Counter Functions Provided in Function Controller Operation

Transfer Direction	Function	Detecting Condition
IN	Transmit buffer flush function	Failed to receive an IN token successfully in the interval frame during the isochronous IN transfer.
OUT	Notification of reception of no token	Failed to receive an OUT token successfully in the interval frame during the isochronous OUT transfer.

Since the interval counting is performed upon reception of SOF or by the recovered SOF, the isochronism can be maintained even if the SOF packet is corrupted. The settable frame interval is $2^{IITV[2:0]}$ frame.

(1) Initializing the Interval Counter in Function Controller Operation

When the MCU is reset, a USB bus reset is detected, or when the PIPEnCTR.ACLRM bit is set to 1, the interval counter is initialized. Note that the PIPEPERI.IITV[2:0] bits are initialized when the MCU is reset.

After the interval counter is initialized, the interval count starts under the following conditions after a packet is successfully transferred.
1. SOF is received after data is transmitted in response to the IN token with PID[1:0] bits set to 01b (BUF response).
2. SOF is received after data is received in response to the OUT token with PID[1:0] bits set to 01b (BUF response).

The interval counter is not initialized under the following conditions.
- When PID[1:0] bits are set to NAK response or STALL response
  The interval timer does not stop. Try to execute a transaction in the next interval.
- USB suspend
  The IITV[2:0] bits are not initialized. When SOF is received, the interval counter starts counting from the value before entering the suspended state.

(2) Interval Counting and Transfer Control in Host Controller Operation

The USBA controls the interval between token issuance operations based on the PIPEPERI.IITV[2:0] bit settings. Specifically, the USBA issues a token for a selected pipe once every $2^{IITV[2:0]}$ frames.
The USBA starts counting the token issuance interval at the frame following the frame in which the PID[1:0] bits have been set to 01b (BUF response).

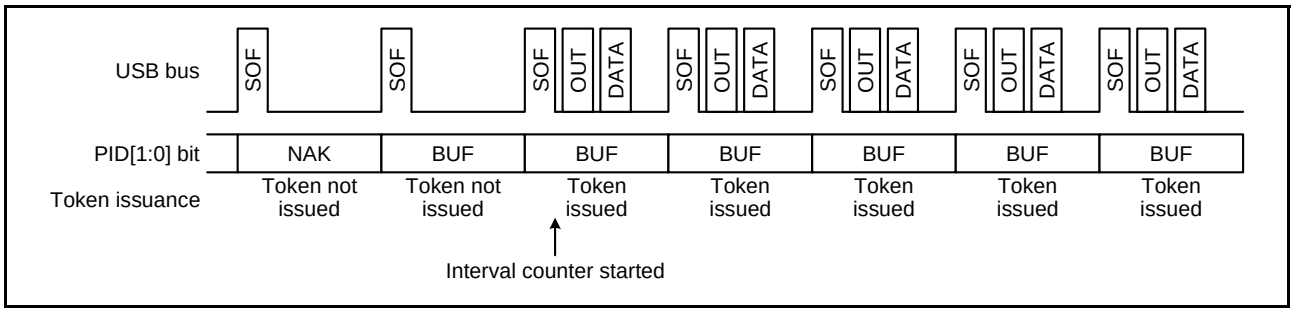


Figure 39.13 Token Issuance When IITV[2:0] Bits are 000b

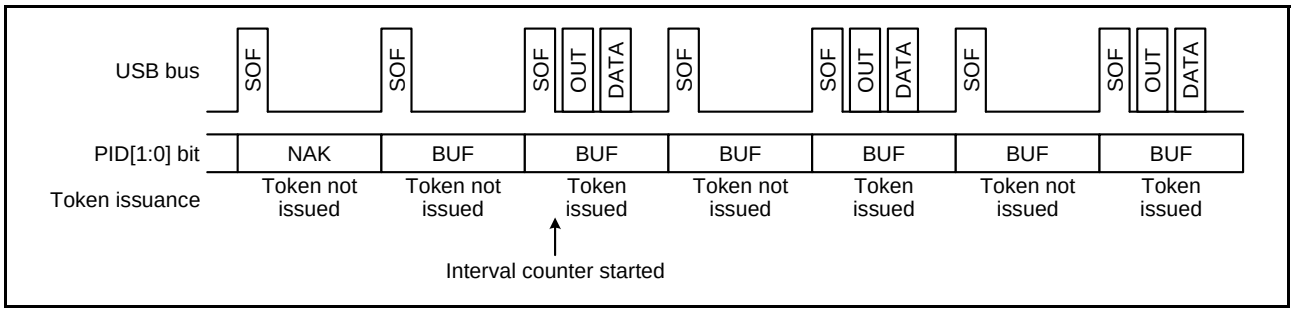


Figure 39.14 Token Issuance When IITV[2:0] Bits are 001b

When the selected pipe is set for isochronous transfers, the USBA carries out the following operation in addition to controlling the token issuance interval. The USBA issues a token even when the NRDY interrupt generation condition is satisfied.

(a) When the selected pipe is for isochronous IN transfers

The USBA generates an NRDY interrupt when the USBA issues an IN token but does not receive a packet successfully from a peripheral device (no response or packet error).
The USBA sets the FRMNUM.OVRN flag to 1 generating an NRDY interrupt when the time to issue an IN token comes while the USBA cannot receive data because the FIFO buffer is full (due to the fact that the CPU or DMAC/DTC is too slow to read data from the FIFO buffer).

(b) When the selected pipe is for isochronous OUT transfers

The USBA sets the OVRN flag to 1 generating an NRDY interrupt and transmitting a zero-length packet when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer (because the CPU or DMAC/DTC is too slow to write data to the FIFO buffer).
The token issuance interval is reset on any of the following conditions.

- When the MCU is reset
  (The IITV[2:0] bits are also set to 000b).
- When the PIPEnCTR.ACLRM bit has been set to 1.

(3) Interval Counting and Transfer Control in Function Controller Operation

(a) When the selected pipe is for isochronous OUT transfers

The USBA generates an NRDY interrupt when the USBA fails to receive a data packet during the frame over the interval set by the PIPEPERI.IITV[2:0] bits.
The USBA also generates an NRDY interrupt when the USBA fails to receive data because of a CRC error or other errors contained in the data packet or because of the FIFO buffer being full.
The NRDY interrupt is generated at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal recovery function allows the interrupt to be generated at the timing to receive the SOF packet.
However, when the IITV[2:0] bits are set to a value other than 000b, the USBA generates an NRDY interrupt on receiving an SOF packet for every interval after starting interval counting operation.
When the PID[1:0] bits are set to 00b (NAK response) after starting the interval timer, the USBA does not generate an NRDY interrupt on receiving an SOF packet.
The condition to start interval counting depends on the setting of IITV[2:0] bits as shown below.

- When the IITV[2:0] bits are 000b:
  The interval counting starts when the PID[1:0] bits of the selected pipe are modified to BUF.

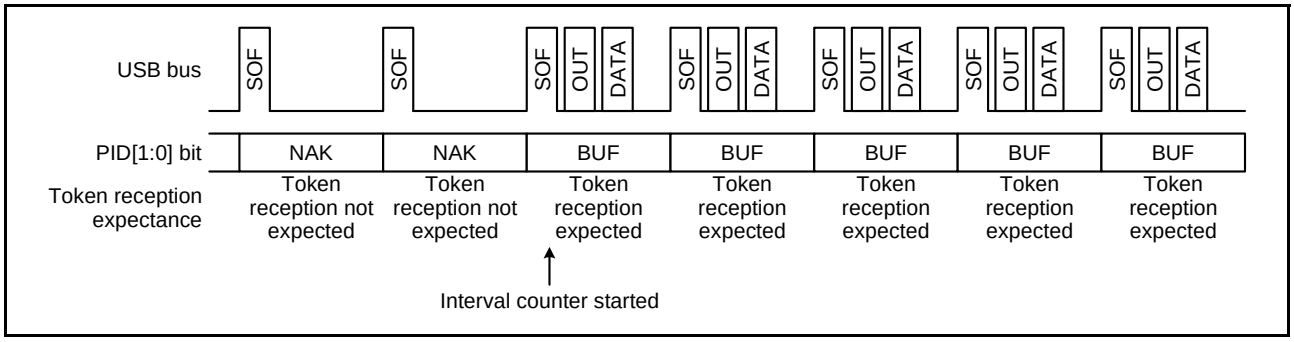


Figure 39.15 Relationship between Frames and Token Reception Expectance When the IITV[2:0] Bits are 000b

- When the IITV[2:0] bits are a value other than 000b:
  The interval counter starts on completion of successful reception of the first data packet after the PID[1:0] bits for the selected pipe have been modified to 01b (BUF response).

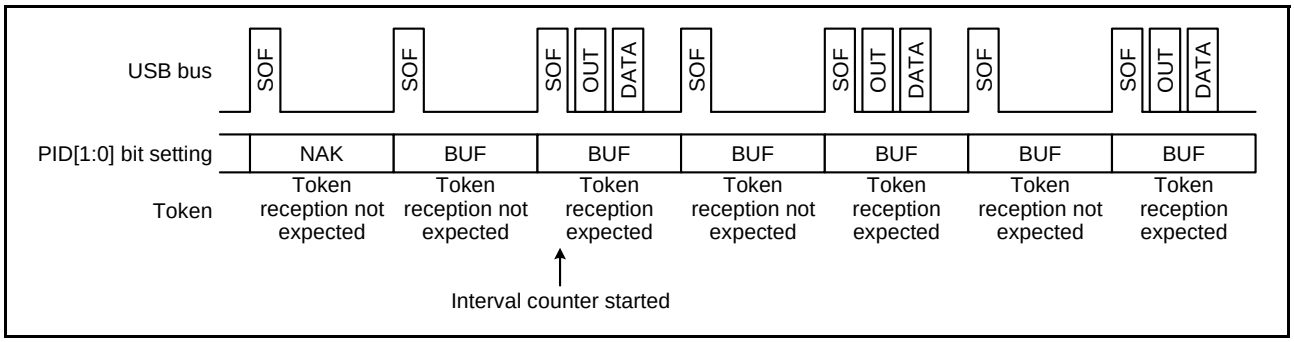


Figure 39.16 Relationship between Frames and Token Reception Expectance When the IITV[2:0] Bits are 001b

(b) When the selected pipe is for isochronous IN transfers

When the PIPEPERI.IFIS bit is cleared to 0, the USBA transmits a data packet in response to a received IN token irrespective of the setting of the PIPEPERI.IITV[2:0] bits.
When the IFIS bit is set to 1, since the FIFO buffer contains data that can be transmitted, the USBA clears the FIFO buffer if it fails to receive an IN token during the frame over the interval set by the IITV[2:0] bits.
The USBA also clears the FIFO buffer when the USBA fails to receive an IN token successfully because of a bus error such as a CRC error contained in the IN token.
The FIFO buffer is cleared at the timing of SOF packet reception. Even if the SOF packet is corrupted, the recovery function allows the FIFO buffer to be cleared at the timing to receive the SOF packet.

The timing to start interval counting depends on the setting of the IITV[2:0] bits (similar to the timing during OUT transfers).
The interval counter is cleared when any of the following conditions is fulfilled in function controller operation.

- When the MCU is reset (here, the IITV[2:0] bits are also set to 000b).
- When the PIPEnCTR.ACLRM bit is set to 1.
- When a USB bus reset is detected.

(4) Isochronous Transfer Transmit Data Setup in Function Controller Operation

In function controller operation, in the isochronous data transmission of the USBA, after data is written to the FIFO buffer, a data packet can be transmitted in the first frame after the SOF packet is detected. This function is called the isochronous transfer transmit data setup function. When double FIFO mode is used, only the one buffer that finished writing earlier can enter the transmittable state even when writing to both buffers has been completed. For this reason, even if two or more IN tokens are received in the same frame, only one packet of FIFO buffer data is transmitted.

When the FIFO buffer is ready to transmit data when an IN token is received, the data is transmitted and a normal response is returned. However, if the FIFO buffer cannot transmit data, a zero-length packet is transmitted and an underrun error occurs.

Figure 39.17 shows an example of transmission using the isochronous transfer transmit data setup function when the IITV[2:0] bits are set to 000b (each frame) by the USBA.

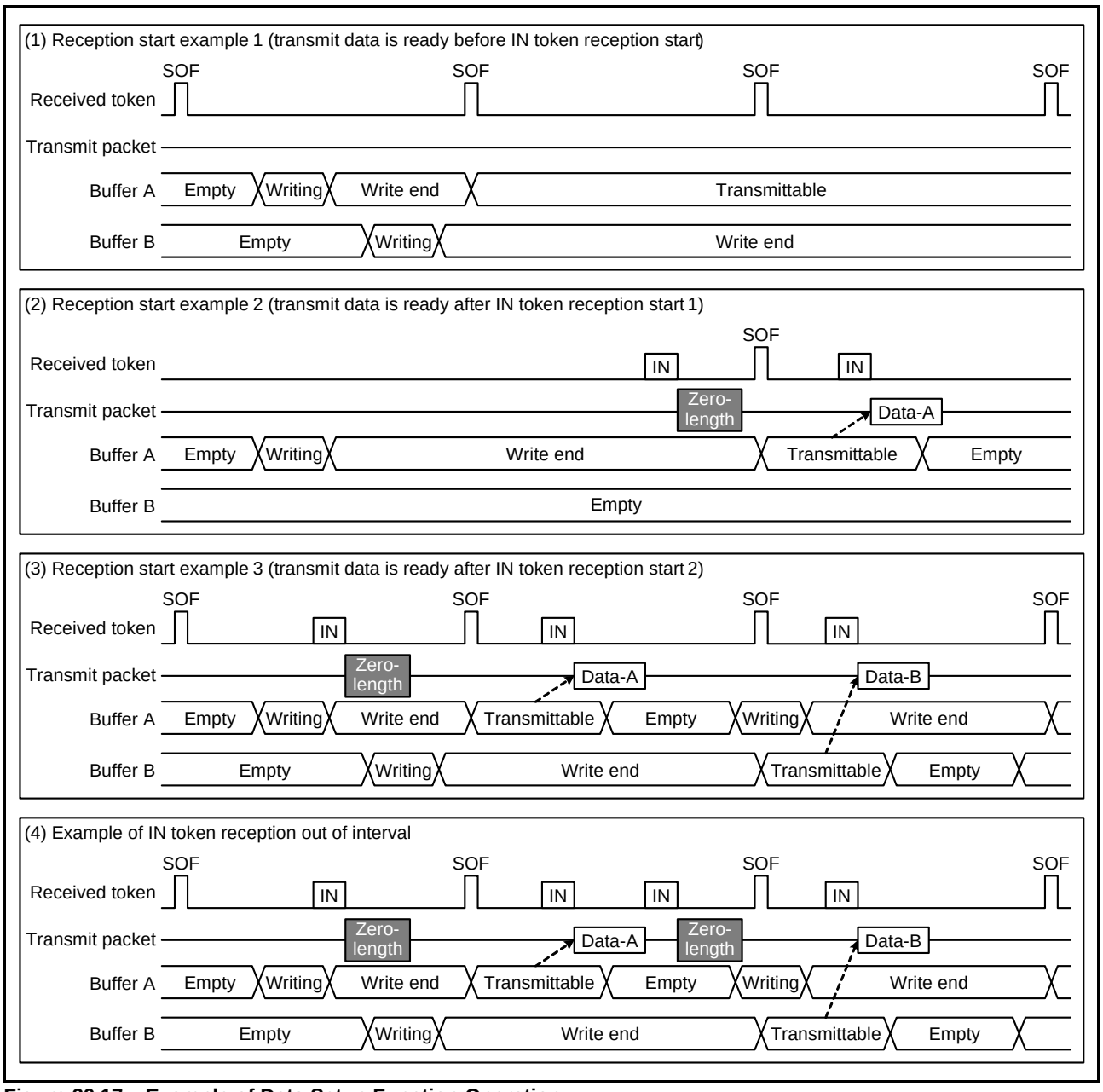


Figure 39.17 Example of Data Setup Function Operation

(5) Isochronous Transfer Transmit Buffer Flush in Function Controller Operation

In function controller operation, the USBA does not receive an IN token in the interval frame during the isochronous data transmission. When the USBA receives the SOF packet of the next frame, the USBA treats it as a corrupted IN token and clears the buffer that is ready to transmit data to make it writable.
When the double-buffer is used and data has already been written to both of the buffers, the USBA regards that the discarded FIFO buffer was transmitted in the same interval frame and makes the FIFO buffer (that is not discarded after the next SOF packet was received) transmittable.
The operation start timing of the buffer flush function varies depending on the value of the PIPEPERI.IITV[2:0] bits.

(1) When the PIPEPERI.IITV[2:0] bits are 000b, the buffer flush operation starts from the first frame after the pipe is enabled.
(2) When the PIPEPERI.IITV[2:0] bits are not 000b, the buffer flush operation starts after the first normal transaction.

Figure 39.18 shows an example of operation of the buffer flush function of the USBA. However, for tokens prior to the interval frame (out of interval), the written data or a zero-length packet as an underrun error is transmitted according to the data setup status.

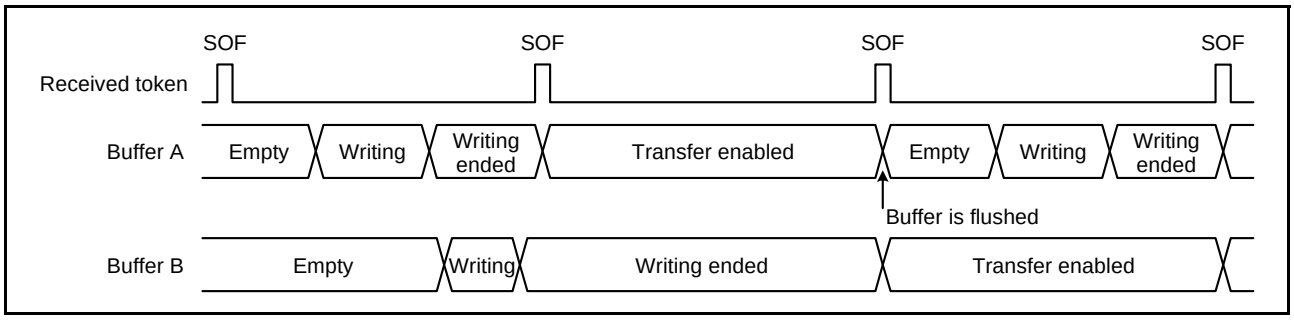


Figure 39.18 Example of Buffer Flush Operation

Figure 39.19 shows an example of occurrence of interval errors of the USBA. There are five types of interval errors ((2) to (6)) shown in the figure below. An interval error occurs at timing ① and the buffer flush function works.
Occurrence of an interval error during the IN transfer activates the buffer flush function, and occurrence of an interval error during the OUT transfer generates an NRDY interrupt.
Distinguish NRDY interrupt (such as receive packet error) from overrun error by the FRMNUM.OVRN flag.
A response according to the FIFO buffer status is returned to shaded tokens.

(1) IN token:

- When the FIFO buffer is ready to transmit data, the data is transmitted and a normal response is returned.
- When the FIFO buffer is not ready to transmit data, a zero-length packet is transmitted and an underrun error occurs.

(2) OUT token:

- When the FIFO buffer is ready to receive data, the data is received and a normal response is returned.
- When the FIFO buffer is not ready to receive data, the received data is discarded and an overrun error occurs.

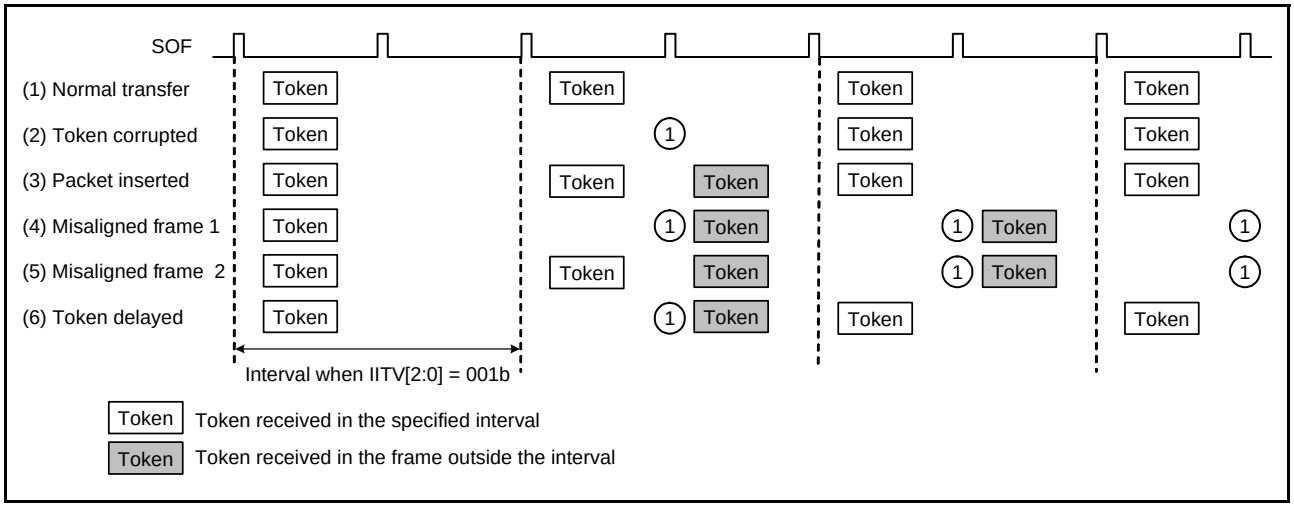


Figure 39.19 Example of Interval Error Occurrence When the PIPEPERI.IITV[2:0] Bits are 001b

39.3.13 SOF Recovery Function

In function controller operation, if an SOF packet is not received at intervals of 1 ms (full-speed operation) due to a corrupted or lost SOF packet, SOF recovery is made in the controller. The SOF recovery starts working when an SOF packet is received while the SYSCFG.USBE bit is set to 1 and the LPSTS.SUSPENDM bit is set to 1. The recovery function is initialized under the following conditions.

- MCU reset
- USB bus reset
- Detection of suspension

The SOF recovery function works with the following specifications.

(1) The frame interval (1 ms) follows the reset handshake result.
(2) The recovery function does not work until an SOF packet is received.
(3) After the first SOF packet is received, 1 ms is counted by the internal 60-MHz clock to recover SOF packets.
(4) After the second (or subsequent) SOF packet is received, SOF packets are recovered using the previous reception interval.
(5) Recovery is not made during suspension or reception of a USB bus reset.

The USBA carries out the following operations when receiving the SOF packet reception. These operations also operate normally with SOF recovery, if the SOF packet was missing.

(1) Updating frame number
(2) Frame number refresh interrupt
(3) Isochronous transfer interval count

If an SOF packet is lost during full-speed operation, the FRMNUM.FRNM[10:0] flags are not updated.

39.3.14 Pipe Schedule

39.3.14.1 Conditions for Generating a Transaction

In host controller operation, after the DVSTCTR0.UACT bit has been set to 1, the USBA generates a transaction under the conditions shown in Table 39.28.

Table 39.28 Conditions for Generating a Transaction

Transaction	Conditions for Generation				
	DIR	PID[1:0]	IITV[2:0]	Buffer State	SUREQ
Setup	— *1	— *1	— *1	— *1	Set 1
Control transfer data stage, status stage, bulk transfer	IN	BUF	— *1	Receive area exists	— *1
	OUT	BUF	— *1	Transmit data exists	— *1
Interrupt transfer	IN	BUF	Valid *2	Receive area exists	— *1
	OUT	BUF	Valid *2	Transmit data exists	— *1
Isochronous transfer	IN	BUF	Valid *2	See note 3	— *1
	OUT	BUF	Valid *2	See note 4	— *1

Note 1. The hyphen (—) in the table indicates that the condition is unrelated to the generating of tokens.
Note 2. "Valid" indicates that, for interrupt transfers and isochronous transfers, a transaction is generated only in transfer frames that are based on the interval counter.
Note 3. A transaction is generated regardless of whether there is a receive area. If there is no receive area, however, the received data is discarded.
Note 4. A transaction is generated regardless of whether there is any data to be transmitted. If there is no data to be transmitted, however, a zero-length packet is transmitted.

39.3.14.2 Transfer Schedule

This section describes the transfer scheduling within a frame of the USBA. After the USBA transmits an SOF packet, the transfer is carried out in the sequence described below.

1. Execution of interrupt transfers and isochronous transfers
   A pipe is searched in the order of Pipe 1 → Pipe 2 → Pipe 6 → Pipe 7 → Pipe 8 → Pipe 9, and then, if there is a pipe for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.
2. Setup transactions for control transfers
   If a setup transaction is possible, it is transmitted.
3. Execution of bulk transfers, control transfer data stages, and control transfer status stages
   A pipe is searched in the order of default control pipe → Pipe 1 → Pipe 2 → Pipe 3 → Pipe 4 → Pipe 5, and then, if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.
   When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

39.3.14.3 Enabling USB Communication

Setting the DVSTCTR0.UACT bit to 1 initiates SOF packet transmission and transaction generation is enabled. Setting the UACT bit to 0 stops SOF packet transmission and the USBA enters the suspend state. If the setting of the UACT bit is changed from 1 to 0, processing stops after the next SOF packet is transmitted.

39.3.15 Battery Charging Detection Processing

Data Contact Detect processing (checking D+ line contact), Primary Detection processing (charger detection processing), and Secondary Detection processing (charger determination processing) defined by Battery Charging Specification Revision 1.2 can be controlled.
Sections 39.3.15.1 and 39.3.15.2 describe operations in host controller operation and function controller operation.

39.3.15.1 Processing in Function Controller Operation

To operate a USBA as a battery charging Portable Device, the following processing is required. Figure 39.20 shows the processing flowchart.

(1) VBUS supply is detected by the VBUS interrupt and VBSTS flag.

(2) Start Primary Detection processing after detecting contact with the D+ and D– lines. Battery Charging Specification Revision 1.2 describes two processing methods for Data Contact Detect. The USBA supports both methods in the ways described below.

- Software wait
  Perform a 300 to 900 ms software wait
- Detecting contact by hardware
  Apply 7 to 13 µA of current to the D+ line to hold the D+ line at the logical high level. This is done to detect the D+ and D– lines going to the logical low level due to pull-down resistors in the host when the D+ and D– lines come in contact with those of the host. The USBA sets the BCCTRL.IDPSRCE bit to 1 and sets the D+ line high. When the D+ line is detected as becoming low by the SYSSTS0.LNST[1:0] bits, the BCCTRL.IDPSRCE bit is set to 0.

(3) Set the BCCTRL.VDPSRCE bit and BCCTRL.IDPSRCE bit to 1 simultaneously to start the Primary Detection processing. Then, perform a 40 ms software wait, and check the BCCTRL.CHGDETSTS flag. If the CHGDETSTS flag is 1, it is determined that a charger is detected, and the BCCTRL.VDPSRCE and BCCTRL.IDPSRCE bits are set to 0.

(4) Set the BCCTRL.VDPSRCE bit and BCCTRL.IDPSRCE bit to 1 simultaneously to start the Secondary Detection processing. Then, perform a 40 ms software wait, check the BCCTRL.PDDETSTS flag, and determine the result of the Secondary Detection processing.

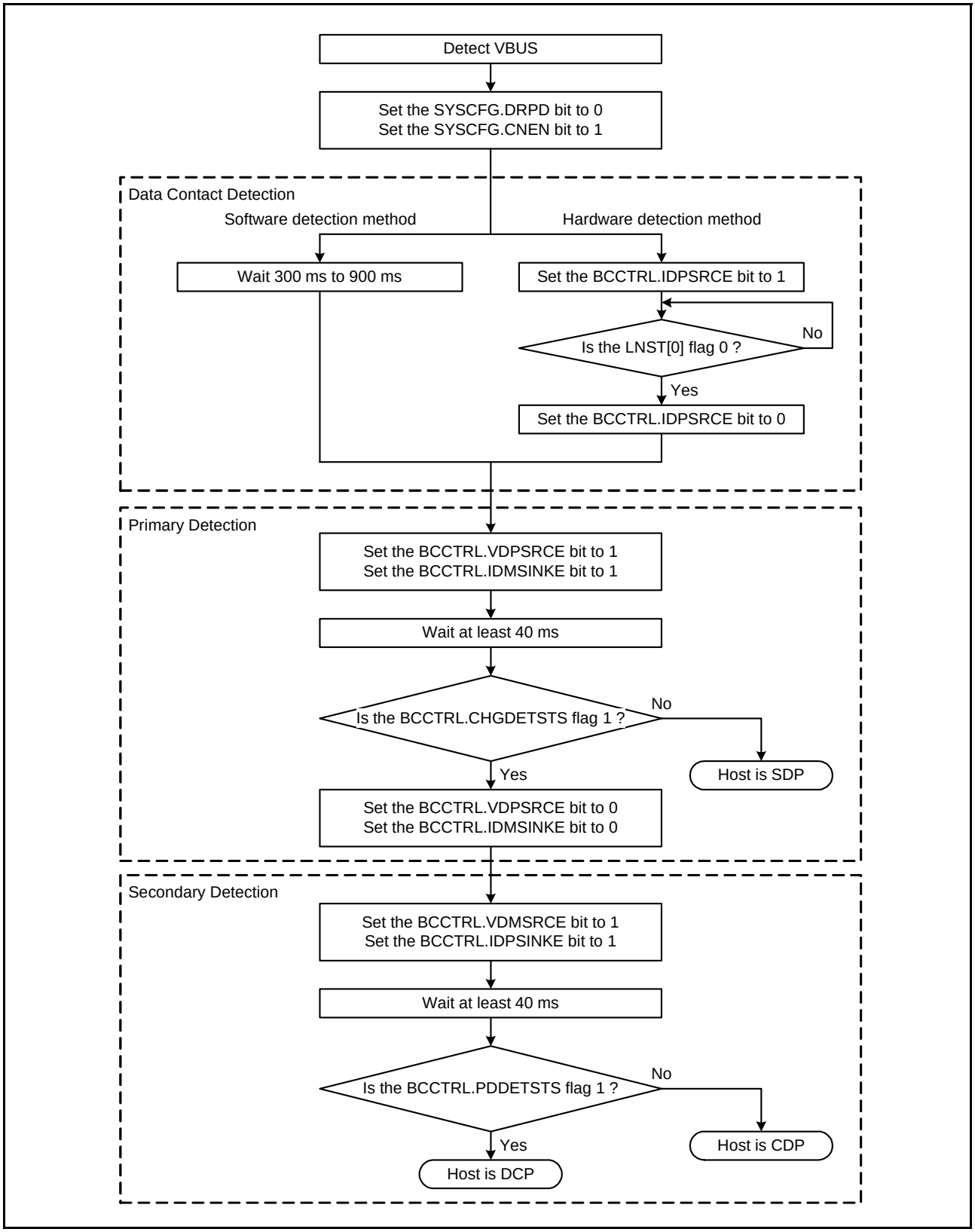


Figure 39.20 Processing as a Portable Device

39.3.15.2 Processing in Host Controller Operation

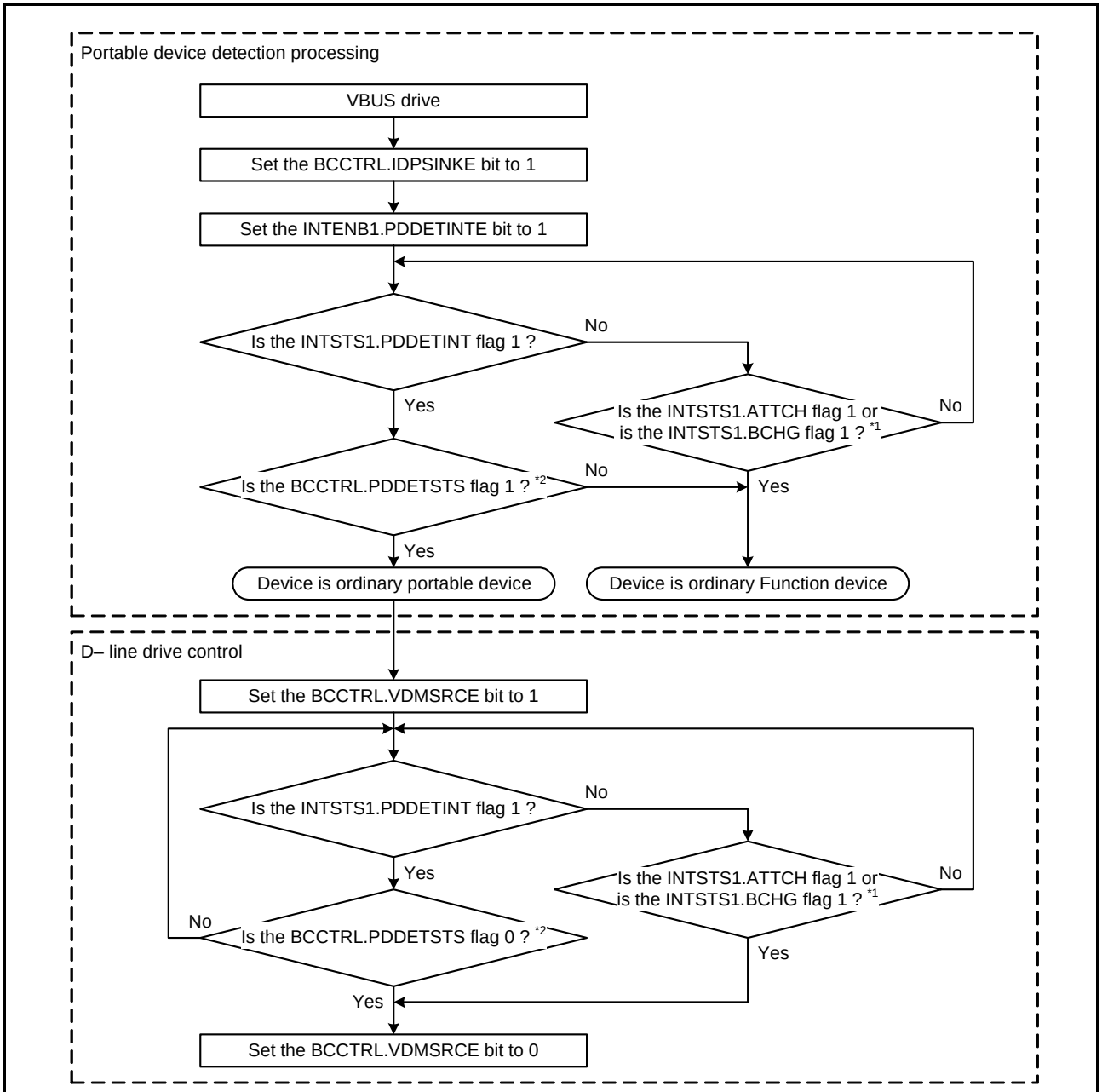
In host controller operation, driving the D– line is required for a Portable Device to implement Primary Detection. Two Primary Detection methods are provided: one for when the hardware has a Portable Device detection function and the other for when the hardware does not have the function or the function is not used even though the hardware has it. Figure 39.21 and Figure 39.22 show processing for the respective methods.

- When using the Portable Device detection function
  The D– line is driven when a Portable Device is detected.

(1) Start driving the VBUS.
(2) Set the BCCTRL.IDMSINKE bit to 1 to enable the Portable Device detection circuit.
(3) Read the BCCTRL.PDDETSTS flag in the Portable Device detection interrupt processing. If the PDDETSTS flag is 1, set the BCCTRL.VDMSRCE bit to 1.
(4) When the BCCTRL.PDDETSTS flag is read in the Portable Device detection interrupt processing after driving the D– line, if the PDDETSTS flag is 0, set the BCCTRL.VDMSRCE bit to 0.

- When not using the Portable Device detection function
  Regardless of whether or not a Portable Device was detected, drive the D– line after device disconnection is detected, and leave the D– line open after connection is detected. Software handles the timing of (1) and (2).

(1) After device disconnection is detected, start driving the D– line within 200 ms.
(2) After device connection is detected, stop driving the D– line within 10 ms.



Note 1. If the PHY clock is being supplied (LPSTS.SUSPENDM bit is 1), the determination is made by the device connection detection interrupt; if the clock is stopped (SUSPENDM bit is 0), the determination is made by the bus change interrupt and the SYSSTS0.LNST[1:0] flags.
Note 2. Read the flag several times to eliminate chattering.

Figure 39.21 Processing Flow When Using the Portable Device Detection Function

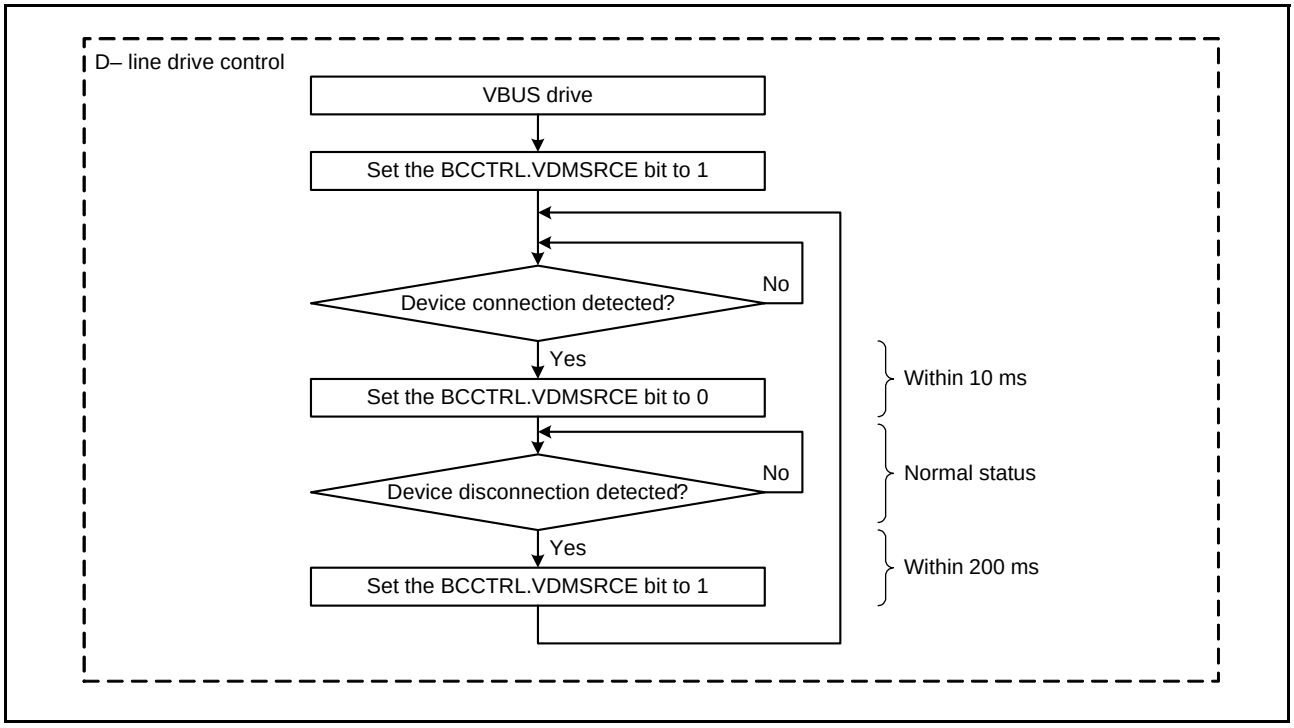


Figure 39.22 Processing Flow When Not Using Portable Device Detection Function

39.3.16 Link Power Management Processing

The Link Power Management standard defines the existing suspended state as the L2 state, and newly defines the L1 state as a state that can be entered into and exited from faster than the L2 state. Table 39.29 is a Comparison of the L1 State and L2 State.

Table 39.29 Comparison of the L1 State and L2 State

Item	L1 State	L2 State
Transition	LPM transaction	Idle for 3 ms
Exit caused by host	(1) Host Drive the K state for a period specified by the HIRD[3:0] bits. Drive period can be specified from 75 µs to 9.96 ms. (2) Function Exits within 10 µs	(1) Host Drive the K state for at least 20 ms (2) Function Exits within 10 ms
Exit caused by function	(1) Function Drive the K state for 50 µs (2) Host Drive the K state for 60 to 990 µs	(1) Function Drive the K state for 1 to 15 ms (2) Host Drive the K state for at least 20 ms
Signaling	Low- and full-speed idle	Low- and full-speed idle
Enabling remote wakeup	Specified by the LPM token	Specified by the SetFeature() request

39.3.16.1 Processing in Function Controller Operation

(1) Contents of the Descriptor

In function controller operation, the USBA must return its Descriptor when a GetDescriptor() request is received. The content of the Descriptor to be returned must be modified depending on whether the USBA accepts or denies transition to or exit from the L1 state according to the LPM transaction. Table 39.30 shows the Relationship Between LPM Support and the Descriptor.

Table 39.30 Relationship Between LPM Support and the Descriptor

LPM Support	*bcdUSB* Field	USB 2.0 Extension Descriptor		Response to Received LPM Request	Note
		Provided/ Not provided	LPM bit		
No	0200h	Not provided	—	No response	Normal operation when the LPM is not supported
	0201h	Provided	0	STALL	Setting when it is obvious that LPM is not supported. In this case, a STALL response must be returned.
Yes	0201h	Provided	1	ACK or NYET	Normal operation when the LPM is supported

The LPM bit in USB 2.0 Extension Descriptor declares whether the USBA accepts or denies transition to or exit from the L1 state. To provide USB 2.0 Extension Descriptor, the *bcdUSB* field of the Standard Device Descriptor must be set to a value of 0201h or larger.

When the LPM is not supported, the USB 2.0 Extension Descriptor is not provided and the *bcdUSB* field value must be 0200h. Even if an LPM token is received in this case, it must be ignored.

When the LPM is not clearly supported, it is also possible to set *bcdUSB* field value to 0201h and the LPM bit in USB 2.0 Extension Descriptor to 0 (no support for the LPM token). In this case, however, the LPM token cannot be ignored and a STALL response must be returned.

When the LPM token is supported, set *bcdUSB* field value to 0201h and set the LPM bit in USB 2.0 Extension Descriptor to 1 (supports the LPM token). This allows acknowledgment of returning a NYET or ACK response to the LPM token.

(2) Processing When Receiving an LPM Token

The following describes the procedure to enter and exit the L1 state in function controller operation.

(1) When the USBA receives an LPM token from the host, the settings of the PL1CTRL1.L1RESPEN, L1RESPMD[1:0], and L1NEGOMD bits determines whether to return an ACK, NYET, or STALL response, or ignore.

(2) When an LPM token is received and an ACK response is returned, if the LPM token is not retransmitted within 8 µs, the USBA enters the L1 state. The USBA processes retransmission of the LPM token and transition to the L1 state. The device state transition interrupt can be used to detect the transition to the L1 state.

(3) There are two types of processing for exiting the L1 state:

- When the host starts the resume:
  When the resume signal is detected, the resume interrupt is generated and the function starts processing to exit the L1 state.
- When the function outputs a remote wakeup signal:
  If the DVSTCTR0.WKUP bit is set to 1, a remote wakeup signal is output to the host. When exiting the L2 state, the DVSTCTR0.WKUP bit must be set to 0 by software, but the USBA sets the DVSTCTR0.WKUP bit to 0 when exiting from the L1 state.

(3) BESL Field Value Negotiation Function

The BESL field value included in the LPM token is the period that the outputs the resume signal when returning from the L1 state. The BESL field value can be adjusted to accommodate the target system. For example, a low HIRD threshold value must be set for a system that focuses on improving transfer efficiency, while a high HIRD threshold value must be set for a system that focuses on low power consumption.

As per the PL1CTRL1.L1NEGOMD and HIRDTHR[3:0] bit settings, an ACK response or NYET response is returned when the received BESL field value is compared to the HIRD threshold value, and a request can be made to the host to modify the BESL field value.

39.3.16.2 Processing in Host Controller Operation

(1) Processing When Transmitting an LPM Token

The following describes the procedure to enter and exit the L1 state in host controller operation.

(1) When the HL1CTRL1.L1REQ bit is set to 1, an LPM token is transmitted from the host to the function.

(2) If an ACK response is received from the function, a transition to the L1 state starts within 10 µs and is completed within 50 µs. In this case, if a transaction error is detected, the LPM token starts retransmitting within 8 µs. Retransmission can be performed up to two times. The USBA controls all of this processing.

(3) There are two types of processing for returning from the L1 state.

- When the host starts the resume:
  When the DVSTCTR0.RESUME bit is set to 1, the USBA starts outputting the resume signal and starts the processing to exit the L1 state.
- When the function outputs a remote wakeup signal:
  When the USBA detects a remote wakeup signal being output from the function, the USBA sets the DVSTCTR0.RESUME bit *1 to 1 and starts outputting the resume signal.

Note 1. Unlike exiting from the L2 state, the USBA sets the DVSTCTR0.RESUME bit to 0. When the RESUME bit becomes 0, the DVSTCTR0.UACT bit becomes 1, and the L1RSMEND interrupt request is generated.

39.3.17 USB External Connection Circuit

Figure 39.23 shows an example of the USB Connector's OTG Connection in Self Powered Mode.
The USBA controls the pull-up resistor of the D+ line, and the pull-down resistor of the D+ and D– lines. Set the SYSCFG.DPRPU bit and DRPD bit to specify pull-up or pull-down for each line.
In addition, in function control operation, if the DPRPU bit is set to 0 while communicating with the USB host, the pull-up resistor for the USB data line becomes invalid, and the USB host can be notified of a device disconnection.

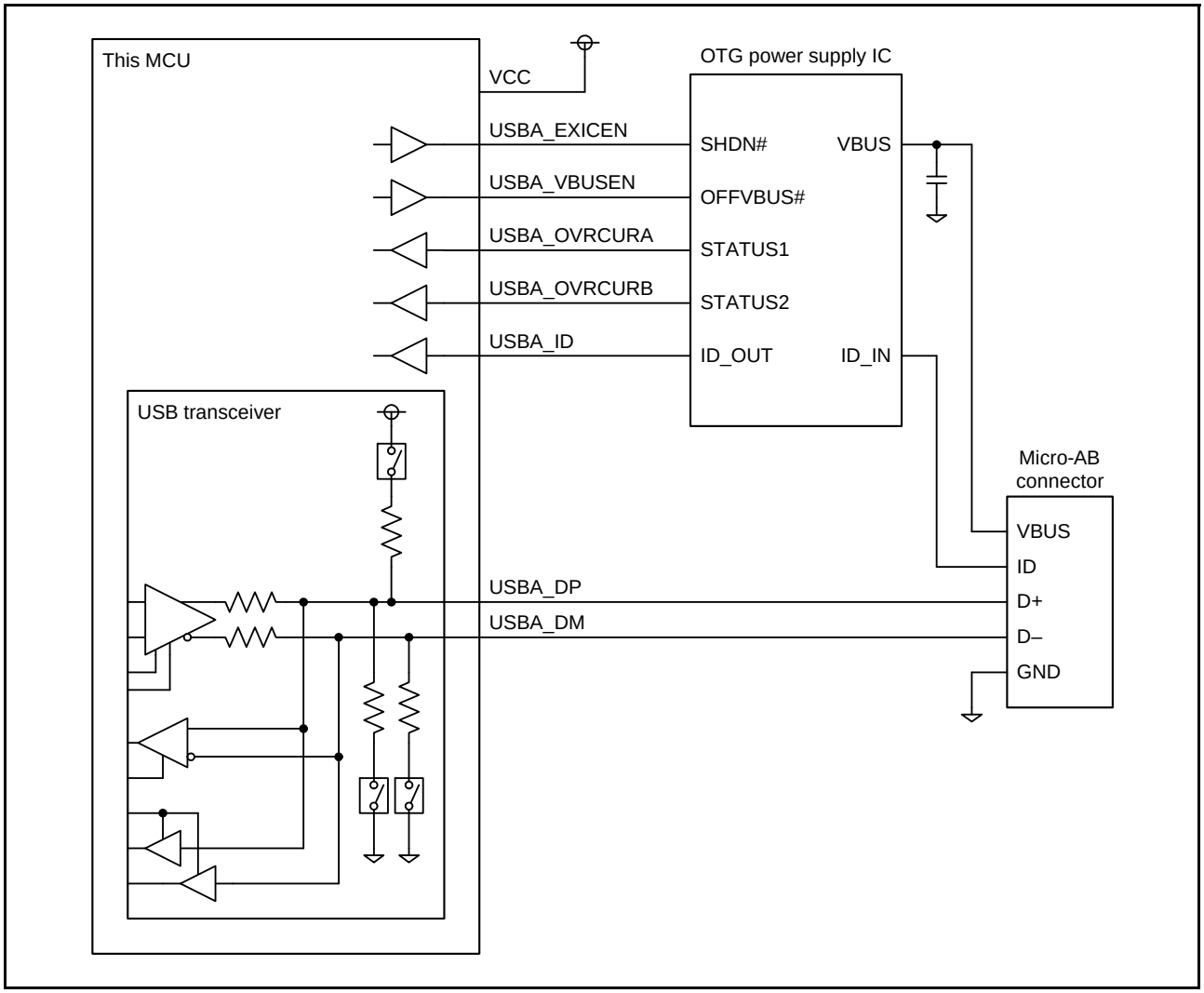


Figure 39.23 USB Connector's OTG Connection in Self Powered Mode

Figure 39.24 shows the USB Connector's Function Connection in Self Powered Mode.

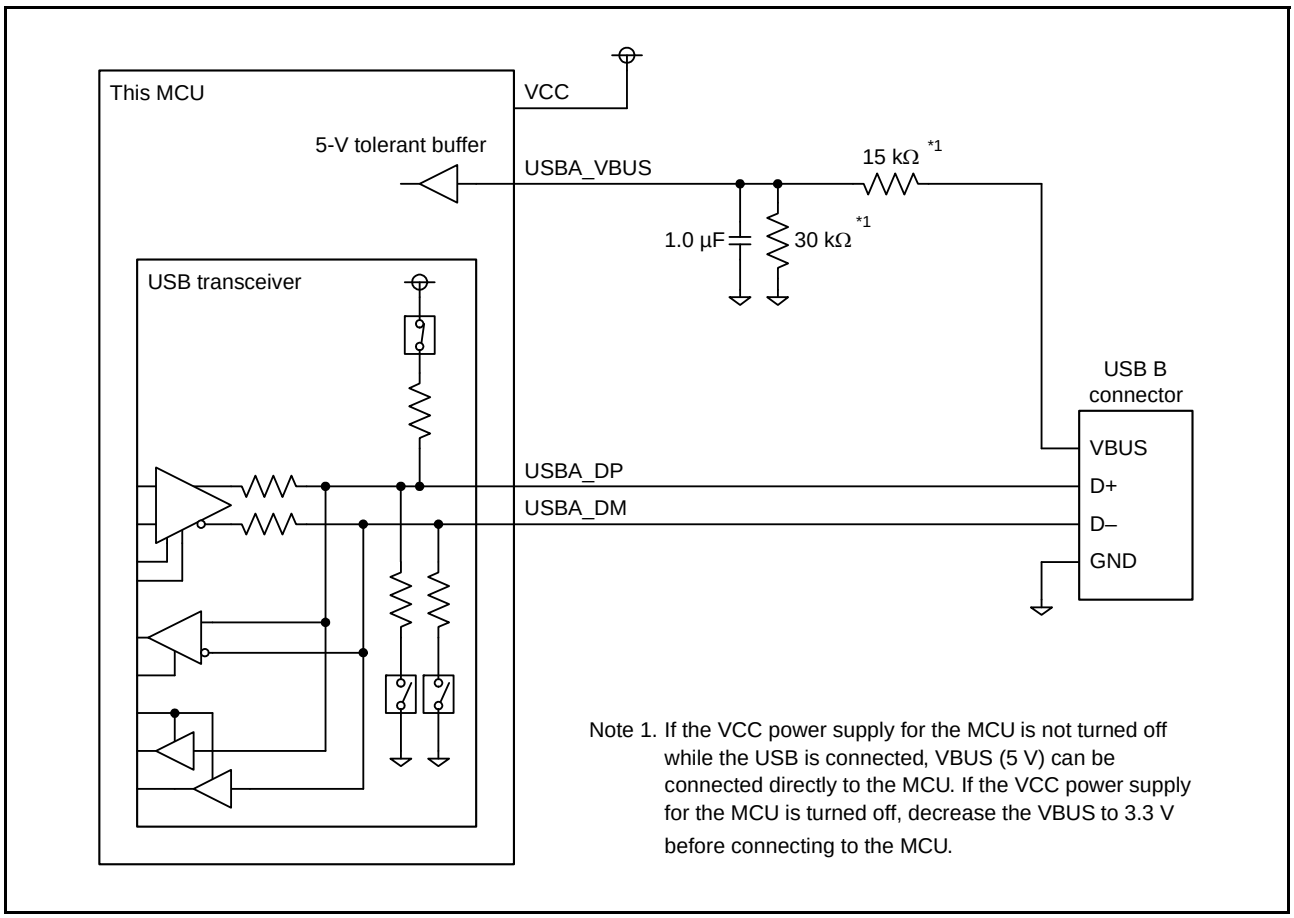


Figure 39.24 USB Connector's Function Connection in Self Powered Mode

Figure 39.25 shows the USB Connector's Function Connection in Bus Powered Mode.

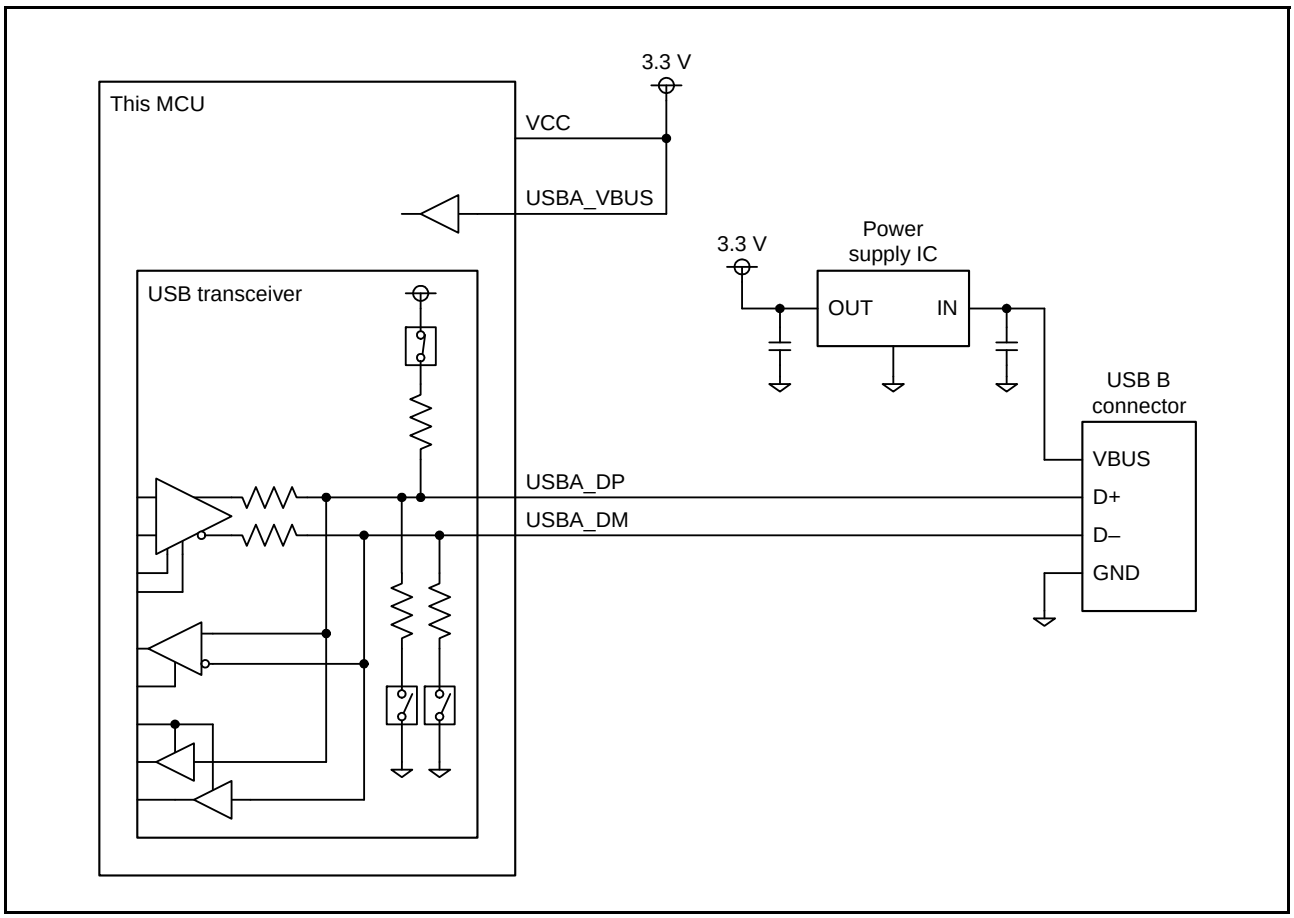


Figure 39.25 USB Connector's Function Connection in Bus Powered Mode

Figure 39.26 shows the USB Connector Host Connection.

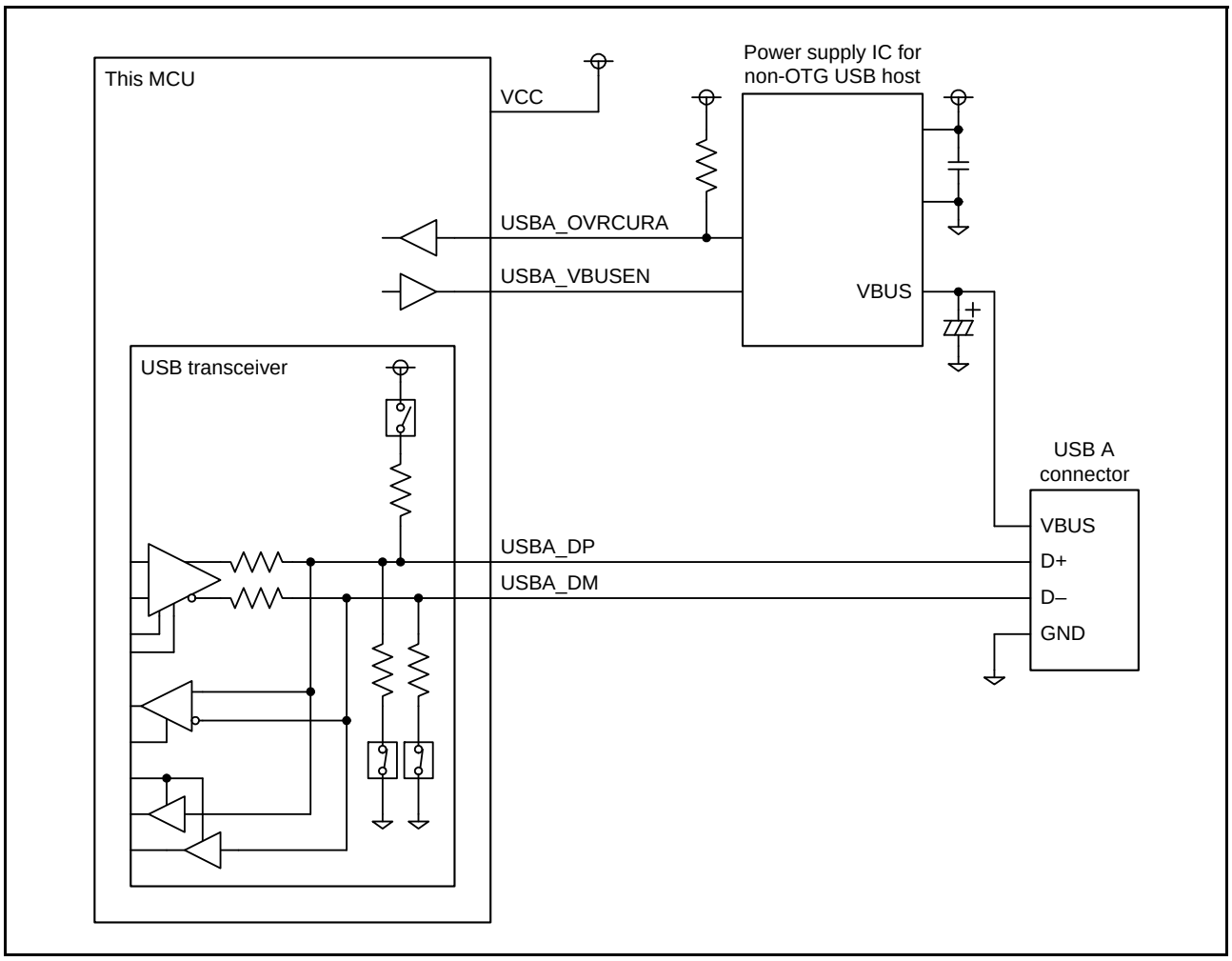


Figure 39.26 USB Connector Host Connection

39.4 Notes on Using the USBA

39.4.1 Setting the USBA Module Stop Function

Set the MSTPCRB.MSTPB12 bit to start or stop USBA operation. The USBA is stopped after a reset. Registers in the USBA can be accessed by setting the MSTPCRB.MSTPB12 bit to 1 (module-stop state is canceled). After enabling access to USBA registers, when operating the PHY circuit, configure settings for items such as the input system clock frequency, and then set the PHYSET.DIRPD bit to 0. Refer to section 11, Low Power Consumption for details.

39.4.2 Setting to Transition to Deep Software Standby Mode

Before transitioning to deep software standby mode, clear the DVSTCTR0.VBUSEN bit to 0.

40. Serial Communications Interface (SCIg, SCIh)

This MCU has nine independent serial communications interface (SCI) channels. The SCI consists of the SCIg module (SCI0 to SCI7) and the SCIh module (SCI12).
The SCIg module (SCI0 to SCI7) can handle both asynchronous and clock synchronous serial communications. Asynchronous serial data communications can be carried out with standard asynchronous communications chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communications Interface Adapter (ACIA). As an extended function in asynchronous communications mode, the SCI also supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards). The SCI is also supports simple SPI interfaces, and simple I^2C-bus interfaces when configured for single-master systems.
The SCIh module includes the functions of the SCIg module, and supports an extended serial communication protocol formed of Start Frames and Information Frames.
In this section, “PCLK” is used to refer to PCLKB.

40.1 Overview

Table 40.1 lists the specifications of the SCIg module, Table 40.2 lists the specifications of the SCIh module, and Table 40.3 lists the specifications of the individual SCI channels.
Figure 40.1 and Figure 40.2 show the block diagrams of the SCIg module, and Figure 40.3 shows the block diagram of the SCIh module.

Table 40.1 SCIg Specifications (1/2)

Item		Description
Serial communication modes		• Asynchronous • Clock synchronous • Smart card interface • Simple I^2C-bus • Simple SPI bus
Transfer speed		Bit rate specifiable with the on-chip baud rate generator.
Full-duplex communications		Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
I/O pins		Refer to Table 40.4 to Table 40.6.
Data transfer		Selectable as LSB first or MSB first transfer*1
Interrupt sources		Transmit end, transmit data empty, receive data full, and receive error Completion of generation of a start condition, restart condition, or stop condition (for simple I^2C mode)
Low power consumption function		Module stop state can be set for each channel.
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Start-bit detection	Low level or falling edge is selectable.
	Break detection	When a framing error occurs, a break can be detected by reading the RXDn pin level directly.
	Clock source	An internal or external clock can be selected. Transfer rate clock input from the TMR can be used. (SCI5, SCI6)
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.

Table 40.1 SCIg Specifications (2/2)

Item		Description
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception
		Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.
Simple I2C mode	Transfer format	I2C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (refer to section 40.2.11, Bit Rate Register (BRR) to set the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	SS input pin function	Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.
Event link function (supported by SCI5 only)		Error (receive error or error signal detection) event output
		Receive data full event output
		Transmit data empty event output
		Transmit end event output

Note 1. In simple I2C mode, only MSB first is available.

Table 40.2 SCIh Specifications (1/2)

Item	Description
Serial communication modes	• Asynchronous • Clock synchronous • Smart card interface • Simple I2C-bus • Simple SPI bus
Transfer speed	Bit rate specifiable with the on-chip baud rate generator.
Full-duplex communications	Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
I/O pins	Refer to Table 40.4 to Table 40.7.
Data transfer	Selectable as LSB first or MSB first transfer*1
Interrupt sources	Transmit end, transmit data empty, receive data full, and receive error Completion of generation of a start condition, restart condition, or stop condition (for simple I2C mode)
Low power consumption function	Module stop state can be set.

Table 40.2 SCIh Specifications (2/2)

Item		Description
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Start-bit detection	Low level or falling edge is selectable.
	Break detection	When a framing error occurs, a break can be detected by reading the RXDn pin level directly.
	Clock source	An internal or external clock can be selected. Transfer rate clock input from the TMR can be used. (SCI12)
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception
		Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.
Simple I^2C mode	Transfer format	I^2C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (refer to section 40.2.11, Bit Rate Register (BRR) to set the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	SS input pin function	Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.
Extended serial mode	Start Frame transmission	• Output of a low level as the Break Field over a specified width and generation of interrupts on completion • Detection of bus collisions and the generation of interrupts on detection
	Start Frame reception	• Detection of the Break Field low width and generation of an interrupt on detection • Comparison of Control Fields 0 and 1 and generation of an interrupt when the two match • Two kinds of data for comparison (primary and secondary) can be set in Control Field 1. • A priority interrupt bit can be set in Control Field 1. • Handling of Start Frames that do not include a Break Field • Handling of Start Frames that do not include a Control Field • Function for measuring bit rates
	I/O control function	• Selectable polarity for TXDX12 and RXDX12 signals • Selection of a digital filter for the RXDX12 signal • Half-duplex operation employing RXDX12 and TXDX12 signals multiplexed on the same pin • Selectable timing for the sampling of data received through RXDX12 • Signals received on RXDX12 can be passed though to SCIg when the extended serial mode control section is off.
	Timer function	• Usable as a reloading timer
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.

Note 1. In simple I^2C mode, only MSB first is available.

Table 40.3 Functions of SCI Channels

Item	SCI0 to SCI4, SCI7	SCI5	SCI6	SCI12
Asynchronous mode	Available	Available	Available	Available
Clock synchronous mode	Available	Available	Available	Available
Smart card interface mode	Available	Available	Available	Available
Simple I2C mode	Available	Available	Available	Available
Simple SPI mode	Available	Available	Available	Available
Extended serial mode	Not available	Not available	Not available	Available
TMR clock input	Not available	Available	Available	Available
Event link function	Not available	Available	Not available	Not available

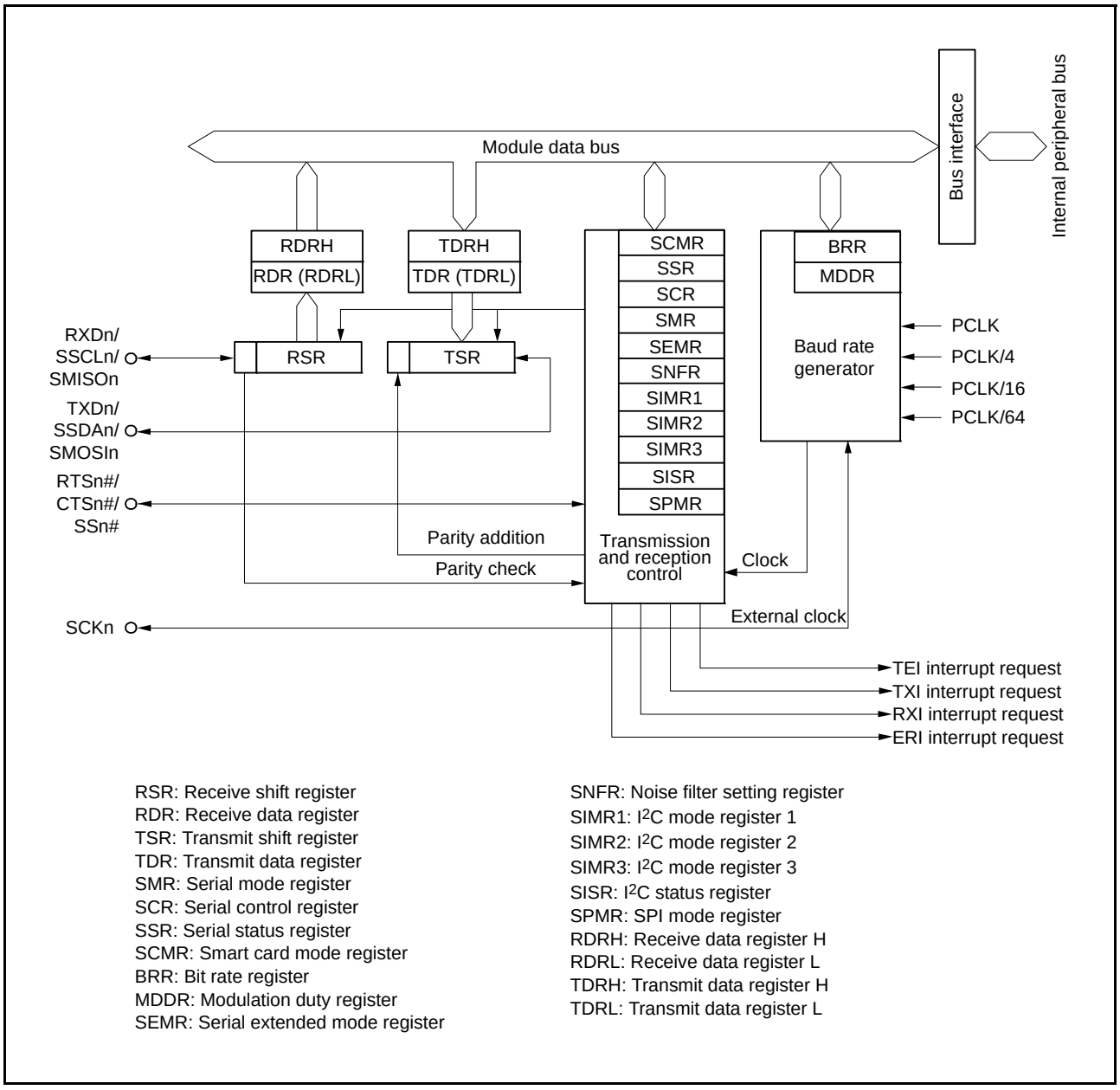


Figure 40.1 Block Diagram of SCIg (SCI0 to SCI4 and SCI7)

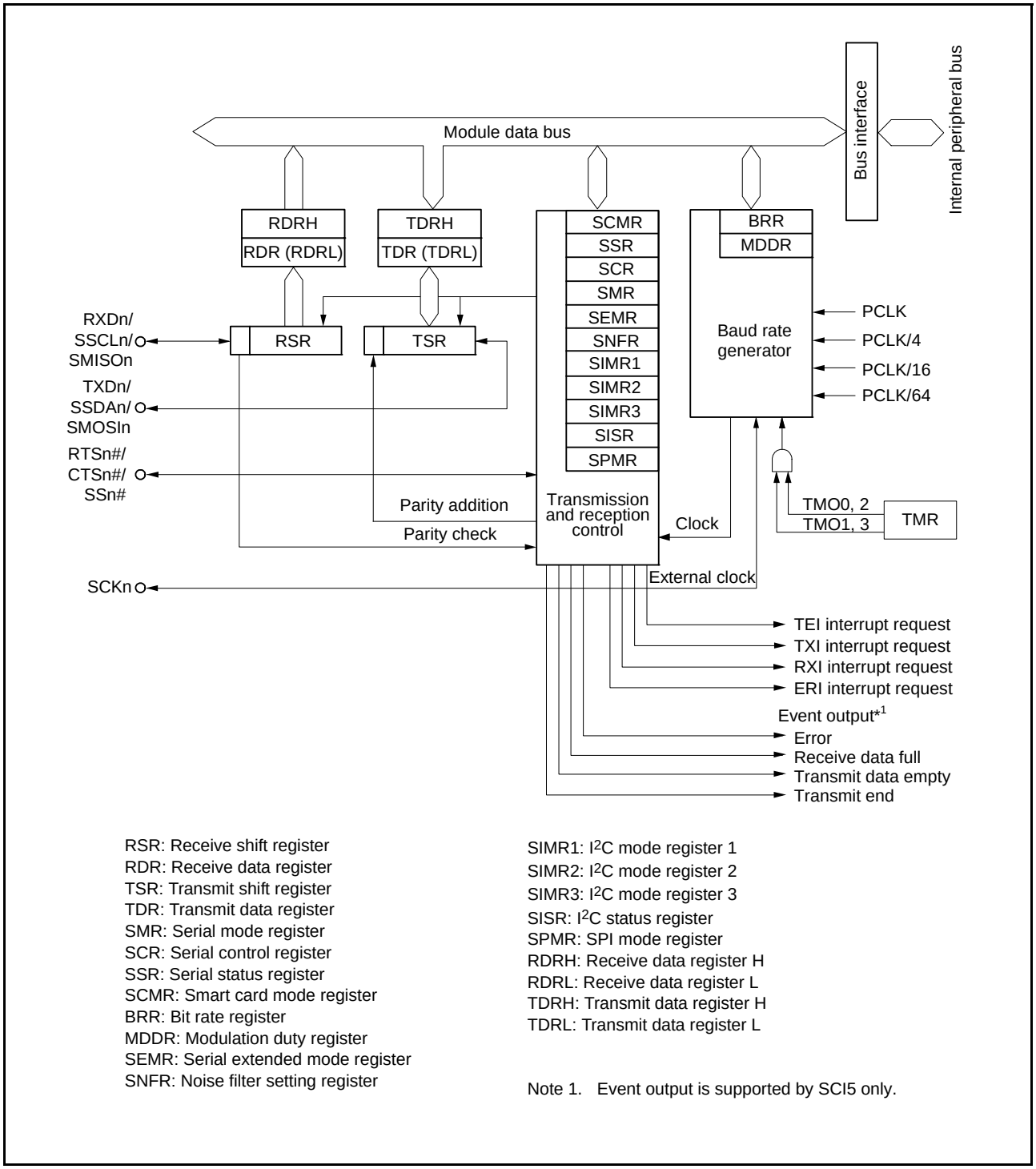


Figure 40.2 Block Diagram of SCIg (SCI5 and SCI6)

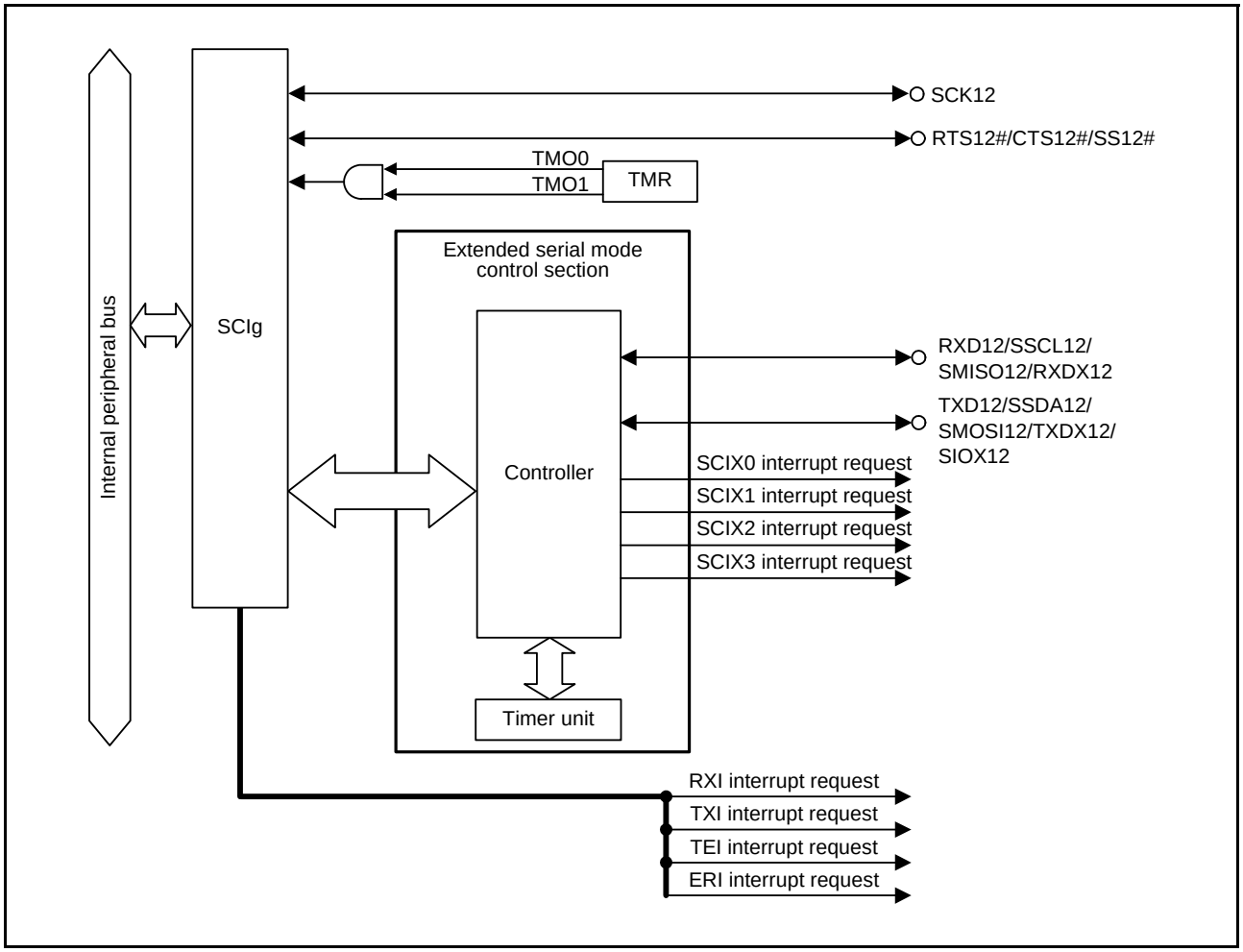


Figure 40.3 Block Diagram of SCIh (SCI12)

Table 40.4 to Table 40.7 list the pin configuration of the SCIs for the individual modes.

Table 40.4 SCI Pin Configuration in Asynchronous Mode and Clock Synchronous Mode

Channel	Pin Name	I/O	Function
SCI0	SCK0	I/O	SCI0 clock input/output
	RXD0	Input	SCI0 receive data input
	TXD0	Output	SCI0 transmit data output
	CTS0#/RTS0#	I/O	SCI0 transfer start control input/output
SCI1	SCK1	I/O	SCI1 clock input/output
	RXD1	Input	SCI1 receive data input
	TXD1	Output	SCI1 transmit data output
	CTS1#/RTS1#	I/O	SCI1 transfer start control input/output
SCI2	SCK2	I/O	SCI2 clock input/output
	RXD2	Input	SCI2 receive data input
	TXD2	Output	SCI2 transmit data output
	CTS2#/RTS2#	I/O	SCI2 transfer start control input/output
SCI3	SCK3	I/O	SCI3 clock input/output
	RXD3	Input	SCI3 receive data input
	TXD3	Output	SCI3 transmit data output
	CTS3#/RTS3#	I/O	SCI3 transfer start control input/output
SCI4	SCK4	I/O	SCI4 clock input/output
	RXD4	Input	SCI4 receive data input
	TXD4	Output	SCI4 transmit data output
	CTS4#/RTS4#	I/O	SCI4 transfer start control input/output
SCI5	SCK5	I/O	SCI5 clock input/output
	RXD5	Input	SCI5 receive data input
	TXD5	Output	SCI5 transmit data output
	CTS5#/RTS5#	I/O	SCI5 transfer start control input/output
SCI6	SCK6	I/O	SCI6 clock input/output
	RXD6	Input	SCI6 receive data input
	TXD6	Output	SCI6 transmit data output
	CTS6#/RTS6#	I/O	SCI6 transfer start control input/output
SCI7	SCK7	I/O	SCI7 clock input/output
	RXD7	Input	SCI7 receive data input
	TXD7	Output	SCI7 transmit data output
	CTS7#/RTS7#	I/O	SCI7 transfer start control input/output
SCI12	SCK12	I/O	SCI12 clock input/output
	RXD12	Input	SCI12 receive data input
	TXD12	Output	SCI12 transmit data output
	CTS12#/RTS12#	I/O	SCI12 transfer start control input/output

Table 40.5 SCI Pin Configuration in Simple I²C Mode

Channel	Pin Name	I/O	Function
SCI0	SSCL0	I/O	SCI0 I²C clock input/output
	SSDA0	I/O	SCI0 I²C data input/output
SCI1	SSCL1	I/O	SCI1 I²C clock input/output
	SSDA1	I/O	SCI1 I²C data input/output
SCI2	SSCL2	I/O	SCI2 I²C clock input/output
	SSDA2	I/O	SCI2 I²C data input/output
SCI3	SSCL3	I/O	SCI3 I²C clock input/output
	SSDA3	I/O	SCI3 I²C data input/output
SCI4	SSCL4	I/O	SCI4 I²C clock input/output
	SSDA4	I/O	SCI4 I²C data input/output
SCI5	SSCL5	I/O	SCI5 I²C clock input/output
	SSDA5	I/O	SCI5 I²C data input/output
SCI6	SSCL6	I/O	SCI6 I²C clock input/output
	SSDA6	I/O	SCI6 I²C data input/output
SCI7	SSCL7	I/O	SCI7 I²C clock input/output
	SSDA7	I/O	SCI7 I²C data input/output
SCI12	SSCL12	I/O	SCI12 I²C clock input/output
	SSDA12	I/O	SCI12 I²C data input/output

Table 40.6 SCI Pin Configuration in Simple SPI Mode (1/2)

Channel	Pin Name	I/O	Function
SCI0	SCK0	I/O	SCI0 clock input/output
	SMISO0	I/O	SCI0 slave transmit data input/output
	SMOSI0	I/O	SCI0 master transmit data input/output
	SS0#	Input	SCI0 chip select input
SCI1	SCK1	I/O	SCI1 clock input/output
	SMISO1	I/O	SCI1 slave transmit data input/output
	SMOSI1	I/O	SCI1 master transmit data input/output
	SS1#	Input	SCI1 chip select input
SCI2	SCK2	I/O	SCI2 clock input/output
	SMISO2	I/O	SCI2 slave transmit data input/output
	SMOSI2	I/O	SCI2 master transmit data input/output
	SS2#	Input	SCI2 chip select input
SCI3	SCK3	I/O	SCI3 clock input/output
	SMISO3	I/O	SCI3 slave transmit data input/output
	SMOSI3	I/O	SCI3 master transmit data input/output
	SS3#	Input	SCI3 chip select input
SCI4	SCK4	I/O	SCI4 clock input/output
	SMISO4	I/O	SCI4 slave transmit data input/output
	SMOSI4	I/O	SCI4 master transmit data input/output
	SS4#	Input	SCI4 chip select input

Table 40.6 SCI Pin Configuration in Simple SPI Mode (2/2)

Channel	Pin Name	I/O	Function
SCI5	SCK5	I/O	SCI5 clock input/output
	SMISO5	I/O	SCI5 slave transmit data input/output
	SMOSI5	I/O	SCI5 master transmit data input/output
	SS5#	Input	SCI5 chip select input
SCI6	SCK6	I/O	SCI6 clock input/output
	SMISO6	I/O	SCI6 slave transmit data input/output
	SMOSI6	I/O	SCI6 master transmit data input/output
	SS6#	Input	SCI6 chip select input
SCI7	SCK7	I/O	SCI7 clock input/output
	SMISO7	I/O	SCI7 slave transmit data input/output
	SMOSI7	I/O	SCI7 master transmit data
	SS7#	Input	SCI7 chip select input
SCI12	SCK12	I/O	SCI12 clock input/output
	SMISO12	I/O	SCI12 slave transmit data input/output
	SMOSI12	I/O	SCI12 master transmit data input/output
	SS12#	Input	SCI12 chip select input

Table 40.7 SCI Pin Configuration in Extended Serial Mode

Channel	Pin Name	I/O	Function
SCI12	RXDX12	Input	SCI12 receive data input
	TXDX12	Output	SCI12 transmit data output
	SIOX12	I/O	SCI12 transfer data input/output

40.2 Register Descriptions

40.2.1 Receive Shift Register (RSR)

RSR is a shift register which is used to receive serial data input from the RXDn pin and converts it into parallel data.
When one frame of data has been received, it is automatically transferred to the RDR register.
The RSR register cannot be directly accessed by the CPU.

40.2.2 Receive Data Register (RDR)

Address(es): SCI0.RDR 0008 A005h, SCI1.RDR 0008 A025h, SCI2.RDR 0008 A045h, SCI3.RDR 0008 A065h, SCI4.RDR 0008 A085h, SCI5.RDR 0008 A0A5h, SCI6.RDR 0008 A0C5h, SCI7.RDR 0008 A0E5h, SCI12.RDR 0008 B305h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

RDR is an 8-bit register that stores receive data.
When one frame of serial data has been received, the received serial data is transferred from RSR to RDR. Then the RSR register can receive the next data.
Since RSR and RDR function as a double buffer in this way, continuous receive operations can be performed.
Read RDR only once after a receive data full interrupt (RXI) has occurred. Note that if next one frame of data is received before reading receive data from RDR, an overrun error occurs.
RDR cannot be written to by the CPU.

40.2.3 Receive Data Register H, L, HL (RDRH, RDRL, RDRHL)

- Receive Data Register H (RDRH)

Address(es): SCI0.RDRH 0008 A010h, SCI1.RDRH 0008 A030h, SCI2.RDRH 0008 A050h, SCI3.RDRH 0008 A070h, SCI4.RDRH 0008 A090h, SCI5.RDRH 0008 A0B0h, SCI6.RDRH 0008 A0D0h, SCI7.RDRH 0008 A0F0h, SCI12.RDRH 0008 B310h

- Receive Data Register L (RDRL)

Address(es): SCI0.RDRL 0008 A011h, SCI1.RDRL 0008 A031h, SCI2.RDRL 0008 A051h, SCI3.RDRL 0008 A071h, SCI4.RDRL 0008 A091h, SCI5.RDRL 0008 A0B1h, SCI6.RDRL 0008 A0D1h, SCI7.RDRL 0008 A0F1h, SCI12.RDRL 0008 B311h

- Receive Data Register HL (RDRHL)

Address(es): SCI0.RDRHL 0008 A010h, SCI1.RDRHL 0008 A030h, SCI2.RDRHL 0008 A050h, SCI3.RDRHL 0008 A070h, SCI4.RDRHL 0008 A090h, SCI5.RDRHL 0008 A0B0h, SCI6.RDRHL 0008 A0D0h, SCI7.RDRHL 0008 A0F0h, SCI12.RDRHL 0008 B310h

	RDRHL															
	RDRH								RDRL							
	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RDRH and RDRL are 8-bit registers that store receive data. Use these registers when asynchronous mode and 9-bit data length are selected.

RDRL is the shadow register of RDR; i.e. access to RDRL is equivalent to access to RDR.

After one frame of data is received, the received data is transferred from the RSR register to these registers, thus allowing the RSR register to receive the next data.

The RSR, RDRH and RDRL registers have a double-buffered construction to enable continuous reception.

Read RDRH and RDRL should be performed only once in the order from RDRH to RDRL when a receive data full interrupt (RXI) request is issued. Note that an overrun error occurs when the next frame of data is received before the received data has been read from RDRL.

The CPU cannot write to the RDRH and RDRL registers. Bits 0 to 7 in RDRH are fixed to 0. These bits are read as 0.

The RDRHL register can be accessed in 16-bit units.

40.2.4 Transmit Data Register (TDR)

Address(es): SCI0.TDR 0008 A003h, SCI1.TDR 0008 A023h, SCI2.TDR 0008 A043h, SCI3.TDR 0008 A063h, SCI4.TDR 0008 A083h, SCI5.TDR 0008 A0A3h, SCI6.TDR 0008 A0C3h, SCI7.TDR 0008 A0E3h, SCI12.TDR 0008 B303h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

TDR is an 8-bit register that stores transmit data.

When the SCI detects that the TSR register is empty, it transfers the transmit data written in the TDR register to the TSR register and starts transmission.

The double-buffered structures of the TDR register and the TSR register enable continuous serial transmission. If the next transmit data has already been written to the TDR register when one frame of data is transmitted, the SCI transfers the written data to the TSR register to continue transmission.

The CPU is able to read from or write to the TDR register at any time. Only write transmit data to the TDR register once after each instance of the transmit data empty interrupt (TXI).

40.2.5 Transmit Data Register H, L, HL (TDRH, TDRL, TDRHL)

- Transmit Data Register H (TDRH)

Address(es): SCI0.TDRH 0008 A00Eh, SCI1.TDRH 0008 A02Eh, SCI2.TDRH 0008 A04Eh, SCI3.TDRH 0008 A06Eh, SCI4.TDRH 0008 A08Eh, SCI5.TDRH 0008 A0AEh, SCI6.TDRH 0008 A0CEh, SCI7.TDRH 0008 A0EEh, SCI12.TDRH 0008 B30Eh

- Transmit Data Register L (TDRL)

Address(es): SCI0.TDRL 0008 A00Fh, SCI1.TDRL 0008 A02Fh, SCI2.TDRL 0008 A04Fh, SCI3.TDRL 0008 A06Fh, SCI4.TDRL 0008 A08Fh, SCI5.TDRL 0008 A0AFh, SCI6.TDRL 0008 A0CFh, SCI7.TDRL 0008 A0EFh, SCI12.TDRL 0008 B30Fh

- Transmit Data Register HL (TDRHL)

Address(es): SCI0.TDRHL 0008 A00Eh, SCI1.TDRHL 0008 A02Eh, SCI2.TDRHL 0008 A04Eh, SCI3.TDRHL 0008 A06Eh, SCI4.TDRHL 0008 A08Eh, SCI5.TDRHL 0008 A0AEh, SCI6.TDRHL 0008 A0CEh, SCI7.TDRHL 0008 A0EEh, SCI12.TDRHL 0008 B30Eh

	TDRHL															
	TDRH								TDRL							
	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TDRH and TDRL are 8-bit registers that store transmit data. Use these registers when asynchronous mode and 9-bit data length are selected.

TDRL is the shadow register of TDR; i.e. access to TDRL is equivalent to access to TDR.

When empty space is detected in the TSR register, the transmit data stored in the TDRH and TDRL registers is transferred to TSR; i.e., transmitting is started.

The TSR, TDRH and TDRL registers have a double-buffered construction to realize continuous reception. When the next data to be transmitted is stored in the TDRL register after one frame of data has been transmitted, the transmitting operation is continued by transfer to the TSR register.

The CPU can read and write to the TDRH and TDRL registers. Bits 0 to 7 in RDRH are fixed to 1. These bits are read as 1. The write value should be 1.

Writing transmit data to the TDRH and TDRL registers should be performed only once in the order from TDRH to TDRL when a transmit data empty interrupt (TXI) request is issued.

The TDRHL register can be accessed in 16-bit units.

40.2.6 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data.

To perform serial data transmission, the SCI first automatically transfers transmit data from TDR to TSR, and then sends the data to the TXDn pin.

TSR cannot be directly accessed by the CPU.

40.2.7 Serial Mode Register (SMR)

Note: Some bits in SMR have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI0.SMR 0008 A000h, SCI1.SMR 0008 A020h, SCI2.SMR 0008 A040h, SCI3.SMR 0008 A060h, SCI4.SMR 0008 A080h, SCI5.SMR 0008 A0A0h, SCI6.SMR 0008 A0C0h, SCI7.SMR 0008 A0E0h, SCI12.SMR 0008 B300h

	b7	b6	b5	b4	b3	b2	b1	b0
	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK (n = 0)*1 0 1: PCLK/4 (n = 1)*1 1 0: PCLK/16 (n = 2)*1 1 1: PCLK/64 (n = 3)*1	R/W*4
b2	MP	Multi-Processor Mode	(Valid only in asynchronous mode) 0: Multi-processor communications function is disabled 1: Multi-processor communications function is enabled	R/W*4
b3	STOP	Stop Bit Length	(Valid only in asynchronous mode) 0: 1 stop bit 1: 2 stop bits	R/W*4
b4	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W*4
b5	PE	Parity Enable	(Valid only in asynchronous mode) • When transmitting 0: Parity bit addition is not performed 1: The parity bit is added • When receiving 0: Parity bit checking is not performed 1: The parity bit is checked	R/W*4
b6	CHR	Character Length	(Valid only in asynchronous mode*2) Selects in combination with the SCMR.CHR1 bit. CHR1 CHR 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*3	R/W*4
b7	CM	Communications Mode	0: Asynchronous mode or simple I2C mode 1: Clock synchronous mode or simple SPI mode	R/W*4

Note 1. n is the decimal notation of the value of n in the BRR register (refer to section 40.2.11, Bit Rate Register (BRR)).
Note 2. In other than asynchronous mode, this bit setting is invalid and a fixed data length of 8 bits is used.
Note 3. LSB first is fixed and the MSB (bit 7) in the TDR register is not transmitted in transmission.
Note 4. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relation between the settings of these bits and the baud rate, refer to section 40.2.11, Bit Rate Register (BRR).

MP Bit (Multi-Processor Mode)

Disables/enables the multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

STOP Bit (Stop Bit Length)

Selects the stop bit length in transmission.
In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

PM Bit (Parity Mode)

Selects the parity mode (even or odd) for transmission and reception.
The setting of the PM bit is invalid in multi-processor mode.

PE Bit (Parity Enable)

When this bit is set to 1, the parity bit is added to transmit data, and the parity bit is checked in reception.
Irrespective of the setting of the PE bit, the parity bit is not added or checked in multi-processor format.

CHR Bit (Character Length)

Selects the data length for transmission and reception.
Selects in combination with the CHR1 bit in SCMR.
In other than asynchronous mode, a fixed data length of 8 bits is used.

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI0.SMR 0008 A000h, SMCI1.SMR 0008 A020h, SMCI2.SMR 0008 A040h, SMCI3.SMR 0008 A060h, SMCI4.SMR 0008 A080h, SMCI5.SMR 0008 A0A0h, SMCI6.SMR 0008 A0C0h, SMCI7.SMR 0008 A0E0h, SMCI12.SMR 0008 B300h

	b7	b6	b5	b4	b3	b2	b1	b0
	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK (n = 0)*1 0 1: PCLK/4 (n = 1)*1 1 0: PCLK/16 (n = 2)*1 1 1: PCLK/64 (n = 3)*1	R/W*2
b3, b2	BCP[1:0]	Base Clock Pulse	Selects the number of base clock cycles in combination with the SCMR.BCP2 bit. Table 40.8 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*2
b4	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W*2
b5	PE	Parity Enable	When this bit is set to 1, a parity bit is added to transmit data, and the parity of received data is checked. Set this bit to 1 in smart card interface mode.	R/W*2
b6	BLK	Block Transfer Mode	0: Normal mode operation 1: Block transfer mode operation	R/W*2
b7	GM	GSM Mode	0: Normal mode operation 1: GSM mode operation	R/W*2

Note 1. n is the decimal notation of the value of n in BRR (refer to section 40.2.11, Bit Rate Register (BRR)).
Note 2. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relationship between the settings of these bits and the baud rate, refer to section 40.2.11, Bit Rate Register (BRR).

BCP[1:0] Bits (Base Clock Pulse)

These bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.

Set these bits in combination with the SCMR.BCP2 bit.

For details, refer to section 40.6.4, Receive Data Sampling Timing and Reception Margin.

Table 40.8 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits

SCMR.BCP2 Bit	SMR.BCP[1:0] Bits		Number of Base Clock Cycles for 1-Bit Transfer Period
0	0	0	93 clock cycles (S = 93)*1
0	0	1	128 clock cycles (S = 128)*1
0	1	0	186 clock cycles (S = 186)*1
0	1	1	512 clock cycles (S = 512)*1
1	0	0	32 clock cycles (S = 32)*1 (Initial Value)
1	0	1	64 clock cycles (S = 64)*1
1	1	0	372 clock cycles (S = 372)*1
1	1	1	256 clock cycles (S = 256)*1

Note 1. S is the value of S in BRR (refer to section 40.2.11, Bit Rate Register (BRR)).

PM Bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd).
For details on the usage of this bit in smart card interface mode, refer to section 40.6.2, Data Format (Except in Block Transfer Mode).

PE Bit (Parity Enable)

Set the PE bit to 1.
The parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

BLK Bit (Block Transfer Mode)

Setting this bit to 1 allows block transfer mode operation.
For details, refer to section 40.6.3, Block Transfer Mode.

GM Bit (GSM Mode)

Setting this bit to 1 allows GSM mode operation.
In GSM mode, the SSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, refer to section 40.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 40.6.8, Clock Output Control.

40.2.8 Serial Control Register (SCR)

Note: Some bits in the SCR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI0.SCR 0008 A002h, SCI1.SCR 0008 A022h, SCI2.SCR 0008 A042h, SCI3.SCR 0008 A062h, SCI4.SCR 0008 A082h, SCI5.SCR 0008 A0A2h, SCI6.SCR 0008 A0C2h, SCI7.SCR 0008 A0E2h, SCI12.SCR 0008 B302h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable	• For SCI0 to SCI4 and SCI7 (Asynchronous mode) b1 b0 0 0: On-chip baud rate generator The SCKn pin functions as I/O port. 0 1: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. 1 x: External clock The clock with a frequency 16 times the bit rate should be input from the SCKn pin. Input a clock signal with a frequency eight times the bit rate when the SEMR.ABCS bit is 1. (Clock synchronous mode) b1 b0 0 x: Internal clock The SCKn pin functions as the clock output pin. 1 x: External clock The SCKn pin functions as the clock input pin.	R/W*1
b1, b0	CKE[1:0]	Clock Enable	• For SCI5, SCI6, and SCI12 (Asynchronous mode) b1 b0 0 0: On-chip baud rate generator The SCKn pin is available for use as an I/O port according to the I/O port settings. 0 1: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. 1 x: External clock or TMR clock • The clock with a frequency 16 times the bit rate should be input from the SCKn pin. Input a clock signal with a frequency eight times the bit rate when the SEMR.ABCS bit is 1. • The TMR clock can be used. The SCKn pin is available for use as an I/O port according to the I/O port settings when the TMR clock is used. (Clock synchronous mode) b1 b0 0 x: Internal clock The SCKn pin functions as the clock output pin. 1 x: External clock The SCKn pin functions as the clock input pin.	R/W*1
b2	TEIE	Transmit End Interrupt Enable	0: A TEI interrupt request is disabled 1: A TEI interrupt request is enabled	R/W

Bit	Symbol	Bit Name	Description	R/W
b3	MPIE	Multi-Processor Interrupt Enable	(Valid in asynchronous mode when SMR.MP = 1) 0: Normal reception 1: When the data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags ORER and FER in SSR to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and normal reception is resumed.	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W*2
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*2
b6	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b7	TIE	Transmit Interrupt Enable	0: A TXI interrupt request is disabled 1: A TXI interrupt request is enabled	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0, while the SMR.CM bit is 1. After setting TE or RE to 1, only 0 can be written to TE and RE. While the SMR.CM bit is 0 and the SIMR1.IICM bit is 0, writing is enabled under any condition.

CKE[1:0] Bits (Clock Enable)

These bits select the clock source and SCKn pin function.

The combination of the settings of these bits and of the SEMR.ACS0 bit sets the internal TMR clock.

TEIE Bit (Transmit End Interrupt Enable)

Enables or disables a TEI interrupt request.

A TEI interrupt request is disabled by setting the TEIE bit to 0.

In simple I^2C mode, the TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STI). In this case, the TEIE bit can be used to enable or disable the STI.

MPIE Bit (Multi-Processor Interrupt Enable)

When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags ORER and FER in the SSR register to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and normal reception is resumed. For details, refer to section 40.4, Multi-Processor Communications Function.

When the data with the multi-processor bit set to 0 is received, the receive data is not transferred from the RSR to the RDR, a receive error is not detected, and setting the flags ORER and FER to 1 is disabled.

When the data with the multi-processor bit set to 1 is received, the MPB bit is set to 1, the MPIE bit is automatically cleared to 0, the RXI and ERI interrupt requests are enabled (if the SCR.RIE bit is set to 1), and setting the flags ORER and FER to 1 is enabled.

Set the MPIE bit to 0 if multi-processor communications function is not to be used.

RE Bit (Receive Enable)

Enables or disables serial reception.

When this bit is set to 1, serial reception is started by detecting the start bit in asynchronous mode or the synchronous clock input in clock synchronous mode. Note that the SMR register should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by setting the RE bit to 0, the ORER, FER, PER, and RDRF flags in the SSR register are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission is started by writing transmit data to TDR. Note that SMR should be set prior to setting the TE bit to 1 in order to designate the transmission format.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.

An RXI interrupt request is disabled by setting the RIE bit to 0.

An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in the SSR register and then setting the flag to 0, or setting the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables TXI interrupt request.

A TXI interrupt request is disabled by setting the TIE bit to 0.

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI0.SCR 0008 A002h, SMCI1.SCR 0008 A022h, SMCI2.SCR 0008 A042h, SMCI3.SCR 0008 A062h, SMCI4.SCR 0008 A082h, SMCI5.SCR 0008 A0A2h, SMCI6.SCR 0008 A0C2h, SMCI7.SCR 0008 A0E2h, SMCI12.SCR 0008 B302h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable	• When SMR.GM = 0 b1 b0 0 0: Output disabled (The SCKn pin is available for use as an I/O port according to the I/O port settings.) 0 1: Clock output 1 x: (Setting prohibited) • When SMR.GM = 1 b1 b0 0 0: Output fixed low x 1: Clock output 1 0: Output fixed high	R/W*1
b2	TEIE	Transmit End Interrupt Enable	This bit should be 0 in smart card interface mode.	R/W
b3	MPIE	Multi-Processor Interrupt Enable	This bit should be 0 in smart card interface mode.	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W*2
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*2
b6	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b7	TIE	Transmit Interrupt Enable	0: A TXI interrupt request is disabled 1: A TXI interrupt request is enabled	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written in TE and RE.

For details on interrupt requests, refer to section 40.12, Interrupt Sources.

CKE[1:0] Bits (Clock Enable)

These bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, refer to section 40.6.8, Clock Output Control.

TEIE Bit (Transmit End Interrupt Enable)

This bit should be 0 in smart card interface mode.

MPIE Bit (Multi-Processor Interrupt Enable)

This bit should be 0 in smart card interface mode.

RE Bit (Receive Enable)

Enables or disables serial reception.

When this bit is set to 1, serial reception is started by detecting the start bit. Note that the SMR register should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by setting the RE bit to 0, the ORER, FER, and PER flags in the SSR register are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission is started by writing transmit data to the TDR register. Note that the SMR register should be set prior to setting the TE bit to 1 in order to designate the transmission format.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.

An RXI interrupt request is disabled by setting the RIE bit to 0.

An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in the SSR register and then setting the flag to 0, or setting the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables TXI interrupt request.

A TXI interrupt request is disabled by setting the TIE bit to 0.

40.2.9 Serial Status Register (SSR)

Some bits in the SSR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI0.SSR 0008 A004h, SCI1.SSR 0008 A024h, SCI2.SSR 0008 A044h, SCI3.SSR 0008 A064h, SCI4.SSR 0008 A084h, SCI5.SSR 0008 A0A4h, SCI6.SSR 0008 A0C4h, SCI7.SSR 0008 A0E4h, SCI12.SSR 0008 B304h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	Sets the multi-processor bit for adding to the transmission frame 0: Data transmission cycles 1: ID transmission cycles	R/W
b1	MPB	Multi-Processor	Value of the multi-processor bit in the reception frame 0: Data transmission cycles 1: ID transmission cycles	R
b2	TEND	Transmit End Flag	0: A character is being transmitted. 1: Character transfer has been completed.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W) *1
b4	FER	Framing Error Flag	0: No framing error occurred 1: A framing error has occurred	R/(W) *1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R/(W) *1
b6	RDRF	Receive Data Full Flag	0: No valid data is held in the RDR register 1: Received data is held in the RDR register	R/(W) *2
b7	TDRE	Transmit Data Empty Flag	0: Data to be transmitted is held in the TDR register 1: No data is held in the TDR register	R/(W) *2

Note 1. Only 0 can be written to this bit, to clear the flag. To clear this flag, confirm that the flag is 1 and then set it to 0.
Note 2. Write 1 when writing is necessary.

MPB Bit (Multi-Processor)

Holds the value of the multi-processor bit in the reception frame. This bit does not change when the SCR.RE bit is 0.

TEND Flag (Transmit End Flag)

Indicates completion of transmission.

[Setting conditions]

- When the SCR.TE bit is set to 0 (serial transmission is disabled)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated at the time of transmission of the tail-end bit of a character being transmitted

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1
  When setting the TEND flag to 0 to complete the interrupt handling, refer to section 15.5.2, Level Detection.

PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
  Although receive data when the parity error occurs is transferred to RDR, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to PER after reading PER = 1
  When setting the PER flag to 0 to complete the interrupt handling, refer to section 15.5.2, Level Detection.
  Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

FER Flag (Framing Error Flag)

Indicates that a framing error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When the stop bit is 0
  In 2-stop-bit mode, only the first stop bit is checked whether it is 1 but the second stop bit is not checked. Note that although receive data when the framing error occurs is transferred to RDR, no RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to FER after reading FER = 1
  When setting the FER flag to 0 to complete the interrupt handling, refer to section 15.5.2, Level Detection.
  Even when the SCR.RE bit is set to 0, the FER flag is not affected and retains its previous value.

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from RDR
  In RDR, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed. Note that, in clock synchronous mode, serial transmission also cannot continue.

[Clearing condition]

- When 0 is written to ORER after reading ORER = 1
  When setting the ORER flag to 0 to complete the interrupt handling, refer to section 15.5.2, Level Detection.
  Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

RDRF Flag (Receive Data Full Flag)

Indicates whether the RDR register has received data.

[Setting condition]

- When data has been received normally, and transferred from RSR to RDR

[Clearing condition]

- When data is read from RDR

TDRE Flag (Transmit Data Empty Flag)

Indicates whether the TDR register has data to be transmitted.

[Setting condition]

- When data is transferred from TDR to TSR

[Clearing condition]

- When data is written to TDR

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI0.SSR 0008 A004h, SMCI1.SSR 0008 A024h, SMCI2.SSR 0008 A044h, SMCI3.SSR 0008 A064h, SMCI4.SSR 0008 A084h, SMCI5.SSR 0008 A0A4h, SMCI6.SSR 0008 A0C4h, SMCI7.SSR 0008 A0E4h, SMCI12.SSR 0008 B304h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	This bit should be set to 0 in smart card interface mode.	R/W
b1	MPB	Multi-Processor	This bit is not used in smart card interface mode. It should be set to 0.	R
b2	TEND	Transmit End Flag	0: A character is being transmitted. 1: Character transfer has been completed.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W) *1
b4	ERS	Error Signal Status Flag	0: Low error signal not responded 1: Low error signal responded	R/(W) *1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R/(W) *1
b6	RDRF	Receive Data Full Flag	0: No valid data is held in the RDR register 1: Received data is held in the RDR register	R/(W) *2
b7	TDRE	Transmit Data Empty Flag	0: Data to be transmitted is held in the TDR register 1: No data is held in the TDR register	R/(W) *2

Note 1. Only 0 can be written to this bit, to clear the flag. To clear this flag, confirm that the flag is 1 and then set it to 0.
Note 2. Write 1 when writing is necessary.

TEND Flag (Transmit End Flag)

With no error signal from the receiving side, this bit is set to 1 when further data for transfer is ready to be transferred to the TDR register.

[Setting conditions]

- When the SCR.TE bit = 0 (serial transmission is disabled)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period has elapsed after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated
  The set timing is determined by register settings as listed below.
  When SMR.GM = 0 and SMR.BLK = 0, 12.5 etu after the start of transmission
  When SMR.GM = 0 and SMR.BLK = 1, 11.5 etu after the start of transmission
  When SMR.GM = 1 and SMR.BLK = 0, 11.0 etu after the start of transmission
  When SMR.GM = 1 and SMR.BLK = 1, 11.0 etu after the start of transmission

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1
  When setting the TEND flag to 0 to complete the interrupt handling, refer to section 15.5.2, Level Detection.

PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
  Although receive data when the parity error occurs is transferred to RDR, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to PER after reading PER = 1
  When setting the PER flag to 0 to complete the interrupt handling, refer to section 15.5.2, Level Detection.
  Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

ERS Flag (Error Signal Status Flag)

[Setting condition]

- When a low error signal is sampled

[Clearing condition]

- When 0 is written to ERS after reading ERS = 1
  When setting the ERS flag to 0 to complete the interrupt handling, refer to section 15.5.2, Level Detection.
  Even when the SCR.RE bit is set to 0, the ERS flag is not affected and retains its previous value.

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from RDR
  In RDR, the receive data prior to an overrun error occurrence is retained, but data received following the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed.

[Clearing condition]

- When 0 is written to ORER after reading ORER = 1
  When setting the ORER flag to 0 to complete the interrupt handling, refer to section 15.5.2, Level Detection.
  Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

RDRF Flag (Receive Data Full Flag)

Indicates whether the RDR register has received data.

[Setting condition]

- When data has been received normally, and transferred from RSR to RDR

[Clearing condition]

- When data is read from RDR

TDRE Flag (Transmit Data Empty Flag)

Indicates whether the TDR register has data to be transmitted.

[Setting condition]

- When data is transferred from TDR to TSR

[Clearing condition]

- When data is written to TDR

40.2.10 Smart Card Mode Register (SCMR)

Address(es): SMCI0.SCMR 0008 A006h, SMCI1.SCMR 0008 A026h, SMCI2.SCMR 0008 A046h, SMCI3.SCMR 0008 A066h, SMCI4.SCMR 0008 A086h, SMCI5.SCMR 0008 A0A6h, SMCI6.SCMR 0008 A0C6h, SMCI7.SCMR 0008 A0E6h, SMCI12.SCMR 0008 B306h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCP2	—	—	CHR1	SDIR	SINV	—	SMIF
Value after reset:	1	1	1	1	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	SMIF	Smart Card Interface Mode Select	0: Non-smart card interface mode (Asynchronous mode, clock synchronous mode, simple SPI mode, or simple I2C mode) 1: Smart card interface mode	R/W*1
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b2	SINV	Transmitted/Received Data Invert	0: TDR contents are transmitted as they are. Receive data is stored as it is in RDR. 1: TDR contents are inverted before being transmitted. Receive data is stored in inverted form in RDR.	R/W*1
b3	SDIR	Transmitted/Received Data Transfer Direction	This bit can be used in the following modes. • Smart card interface mode • Asynchronous mode (multi-processor mode) • Clock synchronous mode • Simple SPI mode Set this bit to 1 if operation is to be in simple I2C mode. 0: Transfer with LSB first 1: Transfer with MSB first	R/W*1
b4	CHR1	Character Length 1	(Only valid in asynchronous mode)*2 Selects in combination with the SMR.CHR bit. CHR1 CHR 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*3	R/W*1
b6, b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7	BCP2	Base Clock Pulse 2	Selects the number of base clock cycles in combination with the SMR.BCP[1:0] bits. Table 40.9 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*1

Note 1. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).
Note 2. The setting is invalid and a fixed data length of 8 bits is used in modes other than asynchronous mode.
Note 3. LSB first should be selected and the value of MSB (b7) in the TDR register cannot be transmitted.

SMIF Bit (Smart Card Interface Mode Select)

When this bit is set to 1, smart card interface mode is selected.

When this bit is set to 0, non-smart card interface mode, i.e., asynchronous mode (including multi-processor mode), clock synchronous mode, simple SPI mode, or simple I2C mode is selected.

SINV Bit (Transmitted/Received Data Invert)

Inverts the transmit/receive data logic level. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the PM bit in the SMR register.

CHR1 bit (Character Length 1)

Selects the data length of transmit/receive data.

Selects in combination with the CHR bit in SMR.

A fixed data length of 8 bits is used in modes other than asynchronous mode.

BCP2 Bit (Base Clock Pulse 2)

Selects the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set this bit in combination with the SMR.BCP[1:0] bits.

Table 40.9 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits

SCMR.BCP2 Bit	SMR.BCP[1:0] Bits		Number of Base Clock Cycles for 1-Bit Transfer Period
0	0	0	93 clock cycles (S = 93)*1
0	0	1	128 clock cycles (S = 128)*1
0	1	0	186 clock cycles (S = 186)*1
0	1	1	512 clock cycles (S = 512)*1
1	0	0	32 clock cycles (S = 32)*1 (Initial Value)
1	0	1	64 clock cycles (S = 64)*1
1	1	0	372 clock cycles (S = 372)*1
1	1	1	256 clock cycles (S = 256)*1

Note 1. S is the value of S in BRR (refer to section 40.2.11, Bit Rate Register (BRR)).

40.2.11 Bit Rate Register (BRR)

Address(es): SCI0.BRR 0008 A001h, SCI1.BRR 0008 A021h, SCI2.BRR 0008 A041h, SCI3.BRR 0008 A061h, SCI4.BRR 0008 A081h, SCI5.BRR 0008 A0A1h, SCI6.BRR 0008 A0C1h, SCI7.BRR 0008 A0E1h, SCI12.BRR 0008 B301h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

The BRR register is an 8-bit register that adjusts the bit rate.
As each SCI channel has independent baud rate generator control, different bit rates can be set for each. Table 40.10 shows the relationship between the setting (N) in the BRR register and the bit rate (B) for normal asynchronous mode, multi-processor transfer, clock synchronous mode, smart card interface mode, simple SPI mode, and simple I²C mode. The BRR register is writable only when the TE and RE bits in the SCR register are 0.

Table 40.10 Relationship between N Setting in BRR and Bit Rate B

Mode	SEMR Settings		BRR Setting	Error
	BGDM Bit	ABCS Bit		
Asynchronous, multi-processor transfer	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \{\frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$
	0	1	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \{\frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$
	1	0		
	1	1	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \{\frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$
Clock synchronous, simple SPI			$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
Smart card interface			$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error\ (\%) = \{\frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1\} \times 100$
Simple I²C*1			$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	

B: Bit rate (bps)
N: BRR setting for on-chip baud rate generator ($0 \le N \le 255$)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in the table below.
Note 1. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I²C mode satisfy the I²C standard.

Table 40.11 Calculating Widths at High and Low Level for SCL

Mode	SCL	Formula (Result in Seconds)
I²C	Width at high level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{PCLK \times 10^6}$
	Width at low level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{PCLK \times 10^6}$

Table 40.12 Clock Source Settings

SMR.CKS[1:0] Bit Setting	Clock Source	n
0 0	PCLK	0
0 1	PCLK/4	1
1 0	PCLK/16	2
1 1	PCLK/64	3

Table 40.13 Base Clock Settings in Smart Card Interface Mode

SCMR.BCP2 Bit Setting	SMR.BCP[1:0] Bit Setting	Base Clock Cycles for 1-bit Period	S
0	0 0	93 clock cycles	93
0	0 1	128 clock cycles	128
0	1 0	186 clock cycles	186
0	1 1	512 clock cycles	512
1	0 0	32 clock cycles	32
1	0 1	64 clock cycles	64
1	1 0	372 clock cycles	372
1	1 1	256 clock cycles	256

Table 40.14 lists examples of N settings in BRR in normal asynchronous mode. Table 40.15 lists the maximum bit rate settable for each operating frequency. Examples of BRR (N) settings in clock synchronous mode and simple SPI mode are listed in Table 40.18. Examples of BRR (N) settings in smart card interface mode are listed in Table 40.20. Examples of BRR (N) settings in simple I^2C mode are listed in Table 40.22. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, refer to section 40.6.4, Receive Data Sampling Timing and Reception Margin. Table 40.16 and Table 40.19 list the maximum bit rates with external clock input.

When either the asynchronous mode base clock select bit (ABCS) or the baud rate generator double-speed mode select bit (BGDM) in the serial extended mode register (SEMR) is set to 1 in asynchronous mode, the bit rate becomes twice that listed in Table 40.14. When both of those registers are set to 1, the bit rate becomes four times the listed value.

Table 40.14 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	70	0.03	3	75	0.48	3	79	–0.12	3	86	0.31
150	2	181	0.16	2	207	0.16	2	223	0.00	2	233	0.16	2	255	0.00
300	2	90	0.16	2	103	0.16	2	111	0.00	2	116	0.16	2	127	0.00
600	1	181	0.16	1	207	0.16	1	223	0.00	1	233	0.16	1	255	0.00
1200	1	90	0.16	1	103	0.16	1	111	0.00	1	116	0.16	1	127	0.00
2400	0	181	0.16	0	207	0.16	0	223	0.00	0	233	0.16	0	255	0.00
4800	0	90	0.16	0	103	0.16	0	111	0.00	0	116	0.16	0	127	0.00
9600	0	45	–0.93	0	51	0.16	0	55	0.00	0	58	–0.69	0	63	0.00
19200	0	22	–0.93	0	25	0.16	0	27	0.00	0	28	1.02	0	31	0.00
31250	0	13	0.00	0	15	0.00	0	16	1.20	0	17	0.00	0	19	–1.70
38400	—	—	—	0	12	0.16	0	13	0.00	0	14	–2.34	0	15	0.00

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	20			25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	–0.25	3	110	–0.02	3	132	0.13	3	145	0.33	3	177	–0.25
150	3	64	0.16	3	80	0.47	3	97	–0.35	3	106	0.39	3	129	0.16
300	2	129	0.16	2	162	–0.15	2	194	0.16	2	214	–0.07	3	64	0.16
600	2	64	0.16	2	80	0.47	2	97	–0.35	2	106	0.39	2	129	0.16
1200	1	129	0.16	1	162	–0.15	1	194	0.16	1	214	–0.07	2	64	0.16
2400	1	64	0.16	1	80	0.47	1	97	–0.35	1	106	0.39	1	129	0.16
4800	0	129	0.16	0	162	–0.15	0	194	0.16	0	214	–0.07	1	64	0.16
9600	0	64	0.16	0	80	0.47	0	97	–0.35	0	106	0.39	0	129	0.16
19200	0	32	–1.36	0	40	–0.76	0	48	–0.35	0	53	–0.54	0	64	0.16
31250	0	19	0.00	0	24	0.00	0	29	0.00	0	32	0.00	0	39	0.00
38400	0	15	1.73	0	19	1.73	0	23	1.73	0	26	–0.54	0	32	–1.36

Bit Rate (bps)	Operating Frequency PCLK (MHz)					
	50			60		
	n	N	Error (%)	n	N	Error (%)
110	3	221	–0.02			
150	3	162	–0.15	3	194	0.16
300	3	80	0.47	3	97	–0.35
600	2	162	–0.15	3	48	–0.35
1200	2	80	0.47	2	97	–0.35
2400	1	162	–0.15	2	48	–0.35
4800	1	80	0.47	1	97	–0.35
9600	0	162	–0.15	1	48	–0.35
19200	0	80	0.47	0	97	–0.35
31250	0	49	0.00	0	59	0.00
38400	0	40	–0.76	0	48	–0.35

Note: This is an example when the ABCS and BGDM bits in SEMR are 0.
When either the ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS and BGDM bits in SEMR are set to 1, the bit rate increases four times.

Table 40.15 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode)

	SEMR Settings						SEMR Settings				
PCLK (MHz)	BGDM Bit	ABCS Bit	n	N	Maximum Bit Rate (bps)	PCLK (MHz)	BGDM Bit	ABCS Bit	n	N	Maximum Bit Rate (bps)
8	0	0	0	0	250000	19.6608	0	0	0	0	614400
		1	0	0	500000			1	0	0	1228800
	1	0	0	0			1	0	0	0	
		1	0	0	1000000			1	0	0	2457600
9.8304	0	0	0	0	307200	20	0	0	0	0	625000
		1	0	0	614400			1	0	0	1250000
	1	0	0	0			1	0	0	0	
		1	0	0	1228800			1	0	0	2500000
10	0	0	0	0	312500	25	0	0	0	0	781250
		1	0	0	625000			1	0	0	1562500
	1	0	0	0			1	0	0	0	
		1	0	0	1250000			1	0	0	3125000
12	0	0	0	0	375000	30	0	0	0	0	937500
		1	0	0	750000			1	0	0	1875000
	1	0	0	0			1	0	0	0	
		1	0	0	1500000			1	0	0	3750000
12.288	0	0	0	0	384000	33	0	0	0	0	1031250
		1	0	0	768000			1	0	0	2062500
	1	0	0	0			1	0	0	0	
		1	0	0	1536000			1	0	0	4125000
14	0	0	0	0	437500	40	0	0	0	0	1250000
		1	0	0	875000			1	0	0	2500000
	1	0	0	0			1	0	0	0	
		1	0	0	1750000			1	0	0	5000000
16	0	0	0	0	500000	50	0	0	0	0	1562500
		1	0	0	1000000			1	0	0	3125000
	1	0	0	0			1	0	0	0	
		1	0	0	2000000			1	0	0	6250000
17.2032	0	0	0	0	537600	60	0	0	0	0	1875000
		1	0	0	1075200			1	0	0	3750000
	1	0	0	0			1	0	0	0	
		1	0	0	2150400			1	0	0	7500000
18	0	0	0	0	562500						
		1	0	0	1125000						
	1	0	0	0							
		1	0	0	2250000						

Table 40.16 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

		Maximum Bit Rate (bps)	
PCLK (MHz)	External Input Clock (MHz)	SEMR.ABCS Bit = 0	SEMR.ABCS Bit = 1
8	2.0000	125000	250000
9.8304	2.4576	153600	307200
10	2.5000	156250	312500
12	3.0000	187500	375000
12.288	3.0720	192000	384000
14	3.5000	218750	437500
16	4.0000	250000	500000
17.2032	4.3008	268800	537600
18	4.5000	281250	562500
19.6608	4.9152	307200	614400
20	5.0000	312500	625000
25	6.2500	390625	781250
30	7.5000	468750	937500
33	8.2500	515625	1031250
40	10.0000	625000	1250000
50	12.5000	781250	1562500
60	15.0000	937500	1875000

Table 40.17 Maximum Bit Rate with TMR Clock Input (Asynchronous Mode)

		Maximum Bit Rate (bps)	
PCLK (MHz)	TMR Clock (MHz)	SEMR.ABCS Bit = 0	SEMR.ABCS Bit = 1
8	4	250000	500000
9.8304	4.9152	307200	614400
10	5	312500	625000
12	6	375000	750000
12.288	6.144	384000	768000
14	7	437500	875000
16	8	500000	1000000
17.2032	8.6016	537600	1075200
18	9	562500	1125000
19.6608	9.8304	614400	1228800
20	10	625000	1250000
25	12.5	781250	1562500
30	15	937500	1875000
33	16.5	1031250	2062500
40	20	1250000	2500000
50	25	1562500	3125000
60	30	1875000	3750000

Table 40.18 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)																			
	8		10		16		20		25		30		33		40		50		60	
	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N
110																				
250	3	124	—	—	3	249														
500	2	249	—	—	3	124	—	—			3	233								
1 k	2	124	—	—	2	249	—	—	3	97	3	116	3	128	3	155	3	194	3	233
2.5 k	1	199	1	249	2	99	2	124	2	155	2	187	2	205	2	249	3	77	3	93
5 k	1	99	1	124	1	199	1	249	2	77	2	93	2	102	2	124	2	155	3	46
10 k	0	199	0	249	1	99	1	124	1	155	1	187	1	205	1	249	2	77	2	93
25 k	0	79	0	99	0	159	0	199	0	249	1	74	1	82	1	99	1	124	1	149
50 k	0	39	0	49	0	79	0	99	0	124	0	149	0	164	1	49	1	61	1	74
100 k	0	19	0	24	0	39	0	49	0	62	0	74	0	82	0	99	0	124	0	149
250 k	0	7	0	9	0	15	0	19	0	24	0	29	0	32	0	39	0	49	0	59
500 k	0	3	0	4	0	7	0	9	—	—	0	14	—	—	0	19	0	24	0	29
1 M	0	1			0	3	0	4	—	—	—	—	—	—	0	9	—	—	0	14
2.5 M			0	0*1			0	1	—	—	0	2	—	—	0	3	0	4	0	5
5 M							0	0*1							0	1			0	2
7.5 M											0	0*1							0	1

Space: Setting prohibited.

—: Can be set, but an error will occur.

Note 1. Continuous transmission or reception is not possible. After transmitting/receiving one frame of data, there is an interval of a 1-bit period before starting transmitting/receiving the next frame of data. The output of the synchronization clock is stopped for a 1-bit period. For this reason, it takes 9 bits worth of time to transfer one frame (8 bits) of data, and the average transfer rate is 8/9 times the bit rate.

Table 40.19 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode)

PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Mbps)
8	1.3333	1.3333
10	1.6667	1.6667
12	2.0000	2.0000
14	2.3333	2.3333
16	2.6667	2.6667
18	3.0000	3.0000
20	3.3333	3.3333
25	4.1667	4.1667
30	5.0000	5.0000
33	5.5000	5.5000
40	6.6667	6.6667
50	8.3333	8.3333
60	10.0000	10.0000

Table 40.20 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372)

Bit Rate (bps)	PCLK (MHz)	n	N	Error (%)
9600	7.1424	0	0	0.00
	10.00	0	1	30
	10.7136	0	1	25
	13.00	0	1	8.99
	14.2848	0	1	0.00
	16.00	0	1	12.01
	18.00	0	2	15.99
	20.00	0	2	6.66
	25.00	0	3	12.49
	30.00	0	3	5.01
	33.00	0	4	7.59
	40.00	0	5	–6.66
	50.00	0	6	0.01
	60.00	0	7	5.01

Table 40.21 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 32)

PCLK (MHz)	Maximum Bit Rate (bps)	n	N
10.00	156250	0	0
10.7136	167400	0	0
13.00	203125	0	0
16.00	250000	0	0
18.00	281250	0	0
20.00	312500	0	0
25.00	390625	0	0
30.00	468750	0	0
33.00	515625	0	0
40.00	625000	0	0
50.00	781250	0	0
60.00	937500	0	0

Table 40.22 BRR Settings for Various Bit Rates (Simple I²C Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	8			10			16			20			25		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	0	24	0.0	0	31	–2.3	1	12	–3.8	1	15	–2.3	1	19	–2.3
25 k	0	9	0.0	0	12	–3.8	1	4	0.0	1	6	–10.7	1	7	–2.3
50 k	0	4	0.0	0	6	–10.7	1	2	–16.7	1	3	–21.9	1	3	–2.3
100 k	0	2	–16.7	0	3	–21.9	0	4	0.0	0	6	–10.7	1	1	–2.3
250 k	0	0	0.0	0	1	–37.5	0	1	0.0	0	2	–16.7	0	3	–21.9
350 k										0	1	–10.7	0	2	–25.6

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	30			33			40			50			60		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	1	23	–2.3	1	25	–0.8	0	124	0.00	2	9	–2.3	1	46	–0.27
25 k	1	9	–6.3	1	10	–6.3	0	40	0.00	2	3	–2.3	0	74	0.00
50 k	1	4	–6.3	1	5	–14.1	0	24	0.00	2	1	–2.3	0	37	–1.32
100 k	1	2	–21.9	1	2	–14.1	0	12	–3.85	1	3	–2.3	0	18	–1.32
250 k	0	3	–6.3	0	4	–17.5	0	4	0.00	0	6	–10.7	0	7	–6.25
350 k	0	2	–10.7	0	2	–1.8	0	3	–10.71	0	4	–10.7	0	4	7.14

Table 40.23 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple I²C Mode)

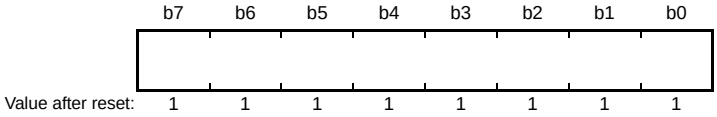
Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	8			10			16			20		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	0	24	43.75/50.00	0	31	44.80/51.20	1	12	45.50/52.00	1	15	44.80/51.20
25 k	0	9	17.50/20.00	0	12	18.20/20.80	1	4	17.50/20.00	1	6	19.60/22.40
50 k	0	4	8.75/10.00	0	6	9.80/11.20	1	2	10.50/12.00	1	3	11.20/12.80
100 k	0	2	5.25/6.00	0	3	5.60/6.40	0	4	4.37/5.00	0	6	4.90/5.60
250 k	0	0	1.75/2.00	0	1	2.80/3.20	0	1	1.75/2.00	0	2	2.10/2.40
350 k										0	1	1.40/1.60

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	25			30			33			40		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	1	19	44.80/51.20	1	23	44.80/51.20	1	25	44.12/50.42	1	32	46.20/52.80
25 k	1	7	17.92/20.48	1	9	18.66/21.33	1	10	18.66/21.33	1	12	18.20/20.80
50 k	1	3	8.96/10.24	1	4	9.33/10.66	1	5	10.18/11.63	1	6	9.80/11.20
100 k	1	1	4.48/5.12	1	2	5.60/6.40	1	2	5.09/5.81	0	13	4.90/5.60
250 k	0	3	2.24/2.56	0	3	1.86/2.13	0	4	2.12/2.42	0	4	1.75/2.00
350 k	0	2	1.68/1.92	0	2	1.40/1.60	0	2	1.27/1.45	0	3	1.40/1.60

Bit Rate (bps)	Operating Frequency PCLK (MHz)					
	50			60		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	2	9	44.80/51.20	3	2	44.80/51.20
25 k	2	3	17.92/20.48	2	4	18.67/21.33
50 k	2	1	8.96/10.24	1	9	9.33/10.67
100 k	1	3	4.48/5.12	1	4	4.67/5.33
250 k	0	6	1.96/2.24	0	7	1.87/2.13
350 k	0	4	1.40/1.60	0	5	1.40/1.60

40.2.12 Modulation Duty Register (MDDR)

Address(es): SCI0.MDDR 0008 A012h, SCI1.MDDR 0008 A032h, SCI2.MDDR 0008 A052h, SCI3.MDDR 0008 A072h, SCI4.MDDR 0008 A092h, SCI5.MDDR 0008 A0B2h, SCI6.MDDR 0008 A0D2h, SCI7.MDDR 0008 A0F2h, SCI12.MDDR 0008 B312h



The MDDR register corrects the bit rate adjusted by the BRR register.
When the SEMR.BRME bit is set to 1, the bit rate generated by the on-chip baud rate generator is evenly corrected according to the settings of the MDDR register (M/256). The relationship between the MDDR register setting (M) and the bit rate (B) is given in Table 40.24.
The range of the value that can be set in the MDDR register is from 80h to FFh. A value other than these cannot be set.
The MDDR register is writable only when the TE and RE bits in the SCR register are 0.

Table 40.24 Relationship between MDDR Setting (M) and Bit Rate (B) When Bit Rate Modulation Function is Used

Mode	SEMR Settings		BRR Setting	Error
	BGDM Bit	ABCS Bit		
Asynchronous multi-processor communication	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	$Error\ (\%) = \{\frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\} \times 100$
	0	1	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times (256/M) \times B} - 1$	$Error\ (\%) = \{\frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\} \times 100$
	1	0		
	1	1	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times (256/M) \times B} - 1$	$Error\ (\%) = \{\frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\} \times 100$
Clock synchronous mode, simple SPI mode*1			$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times (256/M) \times B} - 1$	
Smart card interface mode			$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times (256/M) \times B} - 1$	$Error\ (\%) = \{\frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (256/M) \times (N+1)} - 1\} \times 100$
Simple I^2C*2			$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	

B: Bit rate (bps)
M: MDDR setting (128 ≤ MDDR ≤ 256)
N: BRR setting for baud rate generator (0 ≤ N ≤ 255)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in Table 40.12 and Table 40.13, section 40.2.11, Bit Rate Register (BRR).

Note 1. Do not use this function in clock synchronous mode and in the highest speed settings in simple SPI mode (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).
Note 2. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I^2C mode satisfy the I^2C standard.

40.2.13 Serial Extended Mode Register (SEMR)

Address(es): SCI0.SEMR 0008 A007h, SCI1.SEMR 0008 A027h, SCI2.SEMR 0008 A047h, SCI3.SEMR 0008 A067h, SCI4.SEMR 0008 A087h, SCI5.SEMR 0008 A0A7h, SCI6.SEMR 0008 A0C7h, SCI7.SEMR 0008 A0E7h, SCI12.SEMR 0008 B307h

	b7	b6	b5	b4	b3	b2	b1	b0
	RXDESEL	BGDM	NFEN	ABCS	—	BRME	—	ACS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ACS0	Asynchronous Mode Clock Source Select	(Valid only in asynchronous mode) 0: External clock input 1: Logical AND of two compare matches output from TMR (valid for SCI5, SCI6, and SCI12 only) Available compare match output varies per SCI channel.	R/W*1
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	BRME	Bit Rate Modulation Enable	0: Bit rate modulation function is disabled. 1: Bit rate modulation function is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	ABCS	Asynchronous Mode Base Clock Select	(Valid only in asynchronous mode) 0: Selects 16 base clock cycles for 1-bit period. 1: Selects 8 base clock cycles for 1-bit period.	R/W*1
b5	NFEN	Digital Noise Filter Function Enable	(In asynchronous mode) 0: Noise cancellation function for the RXDn input signal is disabled. 1: Noise cancellation function for the RXDn input signal is enabled. (in simple I^2C mode) 0: Noise cancellation function for the SSCLn and SSDAn input signals is disabled. 1: Noise cancellation function for the SSCLn and SSDAn input signals is enabled. The NFEN bit should be 0 in any mode other than above.	R/W*1
b6	BGDM	Baud Rate Generator Double-Speed Mode Select	(Only valid the CKE[1] bit in SCR is 0 in asynchronous mode). 0: Baud rate generator outputs the clock with normal frequency. 1: Baud rate generator outputs the clock with doubled frequency.	R/W
b7	RXDESEL	Asynchronous Start Bit Edge Detection Select	(Valid only in asynchronous mode) 0: The low level on the RXDn pin is detected as the start bit. 1: A falling edge on the RXDn pin is detected as the start bit.	R/W*1

Note 1. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

The SEMR register is used to select a clock source for 1-bit period in asynchronous mode or a detection method of the start bit.

ACS0 Bit (Asynchronous Mode Clock Source Select)

Selects the clock source in the asynchronous mode.

The ACS0 bit is valid in asynchronous mode (SMR.CM bit = 0) and when an external clock input is selected (SCR.CKE[1:0] bits = 10b or 11b). This bit is used to select an external clock input or the logical AND of compare matches output from the internal TMR.

Set the ACS0 bit to 0 in other than asynchronous mode.

For SCI5, SCI6, and SCI12, the TMOn output (n = 0 to 3) of TMR units 0 and 1 can be set as the serial transfer base clock. Refer to Table 40.25 for details.

These bits for the other SCI channels than SCI5, SCI6, and SCI12 are reserved. The write values to these bits for other than SCI5, SCI6, and SCI12 should be 0.

Table 40.25 Correspondence between SCI Channels and Compare Match Outputs

SCI	TMR	Compare Match Output
SCI5	Unit 0	TMO0, TMO1
SCI6	Unit 1	TMO2, TMO3
SCI12	Unit 0	TMO0, TMO1

Figure 40.4 shows a setting example of when TMO0 and TMO1 in the TMR unit 0 are selected for output.

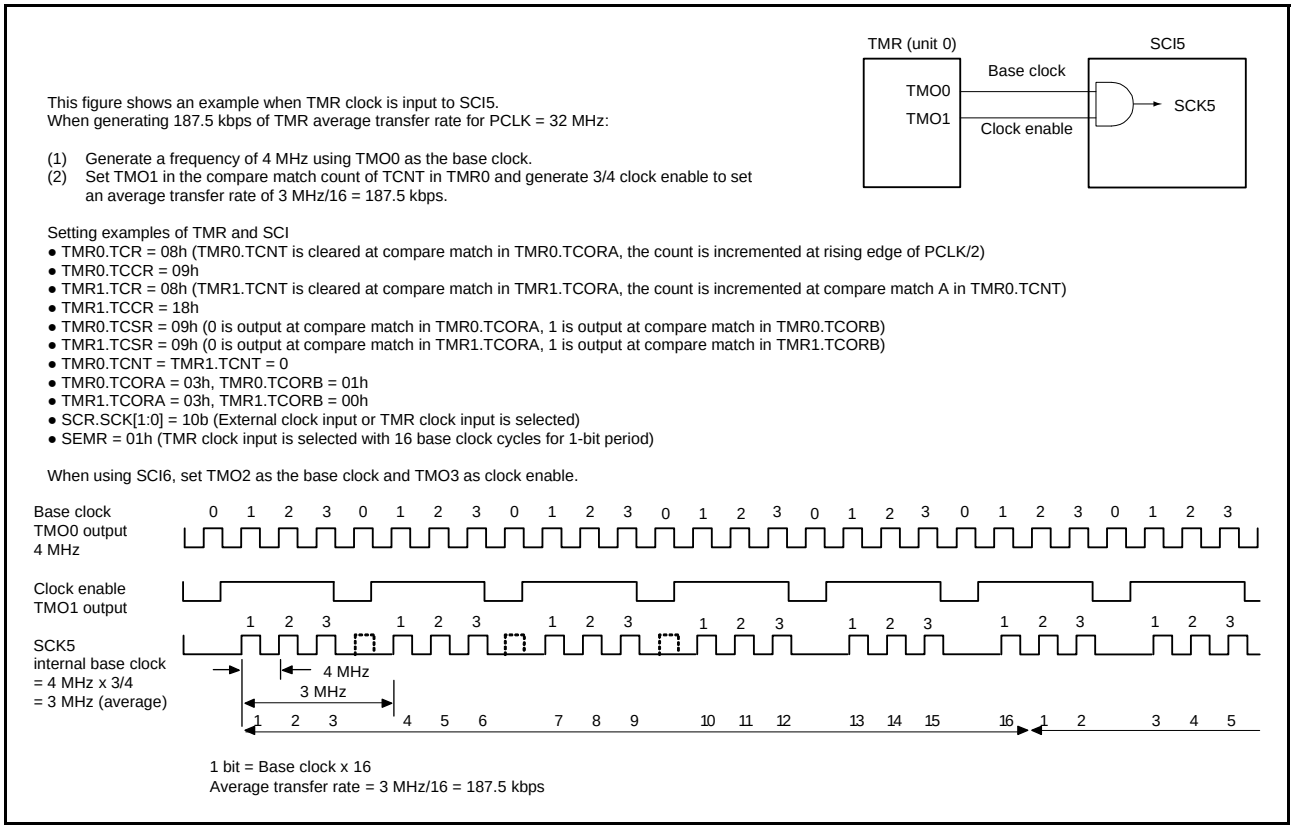


Figure 40.4 Example of Average Transfer Rate Setting When TMR Clock is Input

BRME bit (Bit Rate Modulation Enable)

Enables and disables the bit rate modulation function. The bit rate generated by on-chip baud rate generator is evenly corrected when this function is enabled.

NFEN Bit (Digital Noise Filter Function Enable)

This bit enables or disables the digital noise filter function.
When the function is enabled, noise cancellation is applied to the RXDn input signal in asynchronous mode, and noise cancellation is applied to the SSDAn and SSCLn input signals in simple I^2C mode.
In any mode other than above, set the NFEN bit to 0 to disable the digital noise filter function.
When the function is disabled, input signals are transferred as is, as internal signals.

BGDM bit (Baud Rate Generator Double-Speed Mode Select)

Selects the cycle of output clock for the baud rate generator.
This bit is valid when the on-chip baud rate generator is selected as the clock source (SCR.CKE[1] = 0) in asynchronous mode (SMR.CM = 0). For the clock output from the baud rate generator, either normal or doubled frequency can be selected. The base clock is generated by the clock output from the baud rate generator. When the BGDM bit is set to 1, the base clock cycle is halved and the bit rate is doubled.
Set this bit to 0 in modes other than asynchronous mode.

RXDESEL Bit (Asynchronous Start Bit Edge Detection Select)

Selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, data receiving operation depends on the settings of this bit. Set this bit to 1 when reception should be stopped while a break occurs or when reception should be started without retaining the RXDn pin input at high level for the period of one data frame or longer after completion of the break.
Set this bit to 0 in modes other than asynchronous mode.

40.2.14 Noise Filter Setting Register (SNFR)

Address(es): SCI0.SNFR 0008 A008h, SCI1.SNFR 0008 A028h, SCI2.SNFR 0008 A048h, SCI3.SNFR 0008 A068h, SCI4.SNFR 0008 A088h, SCI5.SNFR 0008 A0A8h, SCI6.SNFR 0008 A0C8h, SCI7.SNFR 0008 A0E8h, SCI12.SNFR 0008 B308h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	NFCS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	NFCS[2:0]	Noise Filter Clock Select	In asynchronous mode, the standard setting for the base clock is as follows. b2 b0 0 0 0: The clock signal divided by 1 is used with the noise filter. In simple I2C mode, the standard settings for the clock source of the on-chip baud rate generator selected by the SMR.CKS[1:0] bits are given below. b2 b0 0 0 1: The clock signal divided by 1 is used with the noise filter. 0 1 0: The clock signal divided by 2 is used with the noise filter. 0 1 1: The clock signal divided by 4 is used with the noise filter. 1 0 0: The clock signal divided by 8 is used with the noise filter. Settings other than above are prohibited.	R/W*1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR are 0 (serial reception and transmission disabled).

NFCS[2:0] Bits (Noise Filter Clock Select)

These bits select the sampling clock for the digital noise filter. To use the noise filter in asynchronous mode, set these bits to 000b. In simple I2C mode, set the bits to a value in the range from 001b to 100b.

40.2.15 I²C Mode Register 1 (SIMR1)

Address(es): SCI0.SIMR1 0008 A009h, SCI1.SIMR1 0008 A029h, SCI2.SIMR1 0008 A049h, SCI3.SIMR1 0008 A069h, SCI4.SIMR1 0008 A089h, SCI5.SIMR1 0008 A0A9h, SCI6.SIMR1 0008 A0C9h, SCI7.SIMR1 0008 A0E9h, SCI12.SIMR1 0008 B309h

	b7	b6	b5	b4	b3	b2	b1	b0
	IICDL[4:0]					—	—	IICM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICM	Simple I²C Mode Select	SMIF IICM 0 0: Asynchronous mode, Multi-processor mode, Clock synchronous mode (in asynchronous mode, synchronous, or simple SPI mode) 0 1: Simple I²C mode 1 0: Smart card interface mode 1 1: Setting prohibited.	R/W*1
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b3	IICDL[4:0]	SSDA Output Delay Select	(Cycles below are of the clock signal from the on-chip baud rate generator.) b7 b3 0 0 0 0 0: No output delay 0 0 0 0 1: 0 to 1 cycle 0 0 0 1 0: 1 to 2 cycles 0 0 0 1 1: 2 to 3 cycles 0 0 1 0 0: 3 to 4 cycles 0 0 1 0 1: 4 to 5 cycles : : 1 1 1 1 0: 29 to 30 cycles 1 1 1 1 1: 30 to 31 cycles	R/W*1

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR are 0 (both serial transmission and reception are disabled).

SIMR1 is used to select simple I²C mode and the number of delay stages for the SSDA output.

IICM Bit (Simple I²C Mode Select)

In conjunction with the SMIF bit in the SCMR register, this bit selects the operating mode.

IICDL[4:0] Bits (SSDA Output Delay Select)

These bits are used to set a delay for output on the SSDAn pin relative to the falling edge of the output on the SSCLn pin. The available delay settings range from no delay to 31 cycles, with the clock signal from the on-chip baud rate generator as the base. The signal obtained by frequency-dividing PCLK by the divisor set in the SMR.CKS[1:0] bits is supplied as the clock signal from the on-chip baud rate generator. Set these bits to 00000b unless operation is in simple I²C mode. In simple I²C mode, set the bits to a value in the range from 00001b to 11111b.

40.2.16 I²C Mode Register 2 (SIMR2)

Address(es): SCI0.SIMR2 0008 A00Ah, SCI1.SIMR2 0008 A02Ah, SCI2.SIMR2 0008 A04Ah, SCI3.SIMR2 0008 A06Ah, SCI4.SIMR2 0008 A08Ah, SCI5.SIMR2 0008 A0AAh, SCI6.SIMR2 0008 A0CAh, SCI7.SIMR2 0008 A0EAh, SCI12.SIMR2 0008 B30Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	IICACKT	—	—	—	IICCSC	IICINTM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICINTM	I²C Interrupt Mode Select	0: Use ACK/NACK interrupts. 1: Use reception and transmission interrupts.	R/W*1
b1	IICCSC	Clock Synchronization	0: No synchronization with the clock signal 1: Synchronization with the clock signal	R/W*1
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	IICACKT	ACK Transmission Data	0: ACK transmission 1: NACK transmission and reception of ACK/NACK	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR are 0 (serial reception and transmission disabled).

SIMR2 is used to select how reception and transmission are controlled in simple I²C mode.

IICINTM Bit (I²C Interrupt Mode Select)

This bit selects the sources of interrupt requests in simple I²C mode.

IICCSC Bit (Clock Synchronization)

Set the IICCSC bit to 1 if the internally generated SSCLn clock signal is to be synchronized when the SSCLn pin has been placed at the low level in the case of a wait inserted by the other device, etc.

The SSCLn clock signal is not synchronized if the IICCSC bit is 0. The SSCLn clock signal is generated in accord with the rate selected in the BRR regardless of the level being input on the SSCLn pin.

Set the IICCSC bit to 1 except during debugging.

IICACKT Bit (ACK Transmission Data)

Transmitted data contains ACK bits. Set this bit to 1 when ACK and NACK bits are received.

40.2.17 I²C Mode Register 3 (SIMR3)

Address(es): SCI0.SIMR3 0008 A00Bh, SCI1.SIMR3 0008 A02Bh, SCI2.SIMR3 0008 A04Bh, SCI3.SIMR3 0008 A06Bh, SCI4.SIMR3 0008 A08Bh, SCI5.SIMR3 0008 A0ABh, SCI6.SIMR3 0008 A0CBh, SCI7.SIMR3 0008 A0EBh, SCI12.SIMR3 0008 B30Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	IICSCLS[1:0]		IICSDAS[1:0]		IICSTIF	IICSTPREQ	IICRSTAREQ	IICSTAREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICSTAREQ	Start Condition Generation	0: A start condition is not generated. 1: A start condition is generated.*1, *3, *4, *5	R/W
b1	IICRSTAREQ	Restart Condition Generation	0: A restart condition is not generated. 1: A restart condition is generated.*2, *3, *4, *5	R/W
b2	IICSTPREQ	Stop Condition Generation	0: A stop condition is not generated. 1: A stop condition is generated.*2, *3, *4, *5	R/W
b3	IICSTIF	Issuing of Start, Restart, or Stop Condition Completed Flag	0: There are no requests for generating conditions or a condition is being generated. 1: A start, restart, or stop condition is completely generated.	R/W
b5, b4	IICSDAS[1:0]	SSDA Output Select	b5 b4 0 0: Serial data output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSDAn pin. 1 1: Place the SSDAn pin in the high-impedance state.	R/W
b7, b6	IICSCLS[1:0]	SSCL Output Select	b7 b6 0 0: Serial clock output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSCLn pin. 1 1: Place the SSCLn pin in the high-impedance state.	R/W

Note 1. Only generate a start condition after checking the bus state and confirming that it is free.
Note 2. Generate a restart or stop condition after checking the bus state and confirming that it is busy.
Note 3. Do not set more than one from among the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits to 1 at a given time.
Note 4. Execute the generation of a condition after the value of the IICSTIF flag is 0.
Note 5. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

SIMR3 is used to control the simple I²C mode start and stop conditions, and to hold the SSDAn and SSCLn pins at fixed levels.

IICSTAREQ Bit (Start Condition Generation)

When a start condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTAREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the start condition

IICRSTAREQ Bit (Restart Condition Generation)

When a restart condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICRSTAREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the restart condition

IICSTPREQ Bit (Stop Condition Generation)

When a stop condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTPREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the stop condition

IICSTIF Flag (Issuing of Start, Restart, or Stop Condition Completed Flag)

After generating a condition, this bit indicates that the generation is completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after setting the IICSTIF flag to 0.

When the IICSTIF flag is 1 while an interrupt request is enabled by setting the SCR.TEIE bit, an STI request is output.

[Setting condition]

- Completion of the generation of a start, restart, or stop condition (however, in cases where this conflicts with any of the conditions for the flag becoming 0 listed below, the other condition takes precedence)

[Clearing conditions]

- Writing 0 to the bit (confirm that the IICSTIF flag is 0 before doing so)
- Writing 0 to the SIMR1.IICM bit (when operation is not in simple I^2C mode)
- Writing 0 to the SCR.TE bit

IICSDAS[1:0] Bits (SSDA Output Select)

These bits control output from the SSDAn pin.

Set the IICSDAS[1:0] and IICSCLS[1:0] bits to the same value during normal operations.

IICSCLS[1:0] Bits (SSCL Output Select)

These bits control output from the SSCLn pin.

Set the IICSCLS[1:0] and IICSDAS[1:0] bits to the same value during normal operations.

40.2.18 I²C Status Register (SISR)

Address(es): SCI0.SISR 0008 A00Ch, SCI1.SISR 0008 A02Ch, SCI2.SISR 0008 A04Ch, SCI3.SISR 0008 A06Ch, SCI4.SISR 0008 A08Ch, SCI5.SISR 0008 A0ACh, SCI6.SISR 0008 A0CCh, SCI7.SISR 0008 A0ECh, SCI12.SISR 0008 B30Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IICACKR
Value after reset:	0	0	x	x	0	x	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	IICACKR	ACK Reception Data Flag	0: ACK received 1: NACK received	R/W*1
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	—	Reserved	The read value is undefined.	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	—	Reserved	The read value is undefined.	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit, to clear the flag.

SISR is used to monitor state in relation to simple I²C mode.

IICACKR Flag (ACK Reception Data Flag)

Received ACK and NACK bits can be read from this bit.

The IICACKR flag is updated at the rising of SSCLn clock for the ACK/NACK receiving bit.

40.2.19 SPI Mode Register (SPMR)

Address(es): SCI0.SPMR 0008 A00Dh, SCI1.SPMR 0008 A02Dh, SCI2.SPMR 0008 A04Dh, SCI3.SPMR 0008 A06Dh, SCI4.SPMR 0008 A08Dh, SCI5.SPMR 0008 A0ADh, SCI6.SPMR 0008 A0CDh, SCI7.SPMR 0008 A0EDh, SCI12.SPMR 0008 B30Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	CKPH	CKPOL	—	MFF	—	MSS	CTSE	SSE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SSE	SSn# Pin Function Enable	0: SSn# pin function is disabled. 1: SSn# pin function is enabled.	R/W*1
b1	CTSE	CTS Enable	0: CTS function is disabled (RTS output function is enabled). 1: CTS function is enabled.	R/W*1
b2	MSS	Master Slave Select	0: Transmission is through the TXDn pin and reception is through the RXDn pin (master mode). 1: Reception is through the TXDn pin and transmission is through the RXDn pin (slave mode).	R/W*1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	MFF	Mode Fault Flag	0: No mode fault error 1: Mode fault error	R/W*2
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	CKPOL	Clock Polarity Select	0: Clock polarity is not inverted. 1: Clock polarity is inverted.	R/W*1
b7	CKPH	Clock Phase Select	0: Clock is not delayed. 1: Clock is delayed.	R/W*1

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR are 0 (both serial transmission and reception are disabled).
Note 2. Only 0 can be written to these bits, which clears the flag.

SPMR is used to select the extension settings in asynchronous and clock synchronous modes.

SSE Bit (SSn# Pin Function Enable)

Set this bit to 1 if the SSn# pin is to be used in control of transmission and reception (in simple SPI mode). Set this bit to 0 in any other mode. Furthermore, even for usage in simple SPI mode, the SSn# pin on the master side is not required to control reception and transmission when master mode (SCR.CKE[1:0] = 00b and MSS = 0) is selected and there is a single master, so the setting for the SSE bit is 0. Do not set both the SSE and CTSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

CTSE Bit (CTS Enable)

Set this bit to 1 if the SSn# pin is to be used for inputting of the CTS control signal to control of transmission and reception. The RTS signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, and simple I^2C mode. Do not set both the CTSE and SSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

MSS Bit (Master Slave Select)

This bit selects between master and slave operation in simple SPI mode. The functions of the TXDn and RXDn pins are reversed when the MSS bit is set to 1, so that data is received through the TXDn pin and transmitted through the RXDn pin.
Set this bit to 0 in modes other than simple SPI mode.

MFF Flag (Mode Fault Flag)

This bit indicates mode fault errors.
In a multi-master configuration, determine the mode fault error occurrence by reading the MFF flag.
[Setting condition]

- Input on the SSn# pin being at the low level during master operation in simple SPI mode (SSE bit = 1 and MSS bit = 0)

[Clearing condition]

- Writing 0 to the bit after it was read as 1

CKPOL Bit (Clock Polarity Select)

This bit selects the polarity of the clock signal output through the SCKn pin. Refer to Figure 40.56 for details.
Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

CKPH Bit (Clock Phase Select)

This bit selects the phase of the clock signal output through the SCKn pin. Refer to Figure 40.56 for details.
Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

40.2.20 Extended Serial Module Enable Register (ESMER)

Address(es): SCI12.ESMER 0008 B320h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ESME
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESME	Extended Serial Mode Enable	0: The extended serial mode is disabled. 1: The extended serial mode is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ESME Bit (Extended Serial Mode Enable)

When the ESME bit is 1, the facilities of the extended serial mode control section are enabled.
When the ESME bit is 0, the extended serial mode control section is initialized.

Table 40.26 Settings of the ESME Bit and Timer Operation Mode

ESME Bit	Timer Mode	Break Field Low Width Determination Mode	Break Field Low Width Output Mode
0	Available*1	Not available	Not available
1	Available	Available	Available

Note 1. Operation is only possible with PCLK selected.

40.2.21 Control Register 0 (CR0)

Address(es): SCI12.CR0 0008 B321h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	BRME	RXDSF	SFSF	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	SFSF	Start Frame Status Flag	0: Start Frame detection function is disabled. 1: Start Frame detection function is enabled.	R
b2	RXDSF	RXDX12 Input Status Flag	0: RXDX12 input is enabled. 1: RXDX12 input is disabled.	R
b3	BRME	Bit Rate Measurement Enable	0: Measurement of bit rate is disabled. 1: Measurement of bit rate is enabled.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

40.2.22 Control Register 1 (CR1)

Address(es): SCI12.CR1 0008 B322h

	b7	b6	b5	b4	b3	b2	b1	b0
	PIBS[2:0]			PIBE	CF1DS[1:0]		CF0RE	BFE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFE	Break Field Enable	0: Break Field detection is disabled. 1: Break Field detection is enabled.	R/W
b1	CF0RE	Control Field 0 Reception Enable	0: Reception of Control Field 0 is disabled. 1: Reception of Control Field 0 is enabled.	R/W
b3, b2	CF1DS[1:0]	Control Field 1 Data Register Select	b3 b2 0 0: Selects comparison with the value in PCF1DR. 0 1: Selects comparison with the value in SCF1DR. 1 0: Selects comparison with the values in PCF1DR and SCF1DR. 1 1: Setting prohibited.	R/W
b4	PIBE	Priority Interrupt Bit Enable	0: The priority interrupt bit is disabled. 1: The priority interrupt bit is enabled.	R/W
b7 to b5	PIBS[2:0]	Priority Interrupt Bit Select	b7 b5 0 0 0: 0th bit of Control Field 1 0 0 1: 1st bit of Control Field 1 0 1 0: 2nd bit of Control Field 1 0 1 1: 3rd bit of Control Field 1 1 0 0: 4th bit of Control Field 1 1 0 1: 5th bit of Control Field 1 1 1 0: 6th bit of Control Field 1 1 1 1: 7th bit of Control Field 1	R/W

40.2.23 Control Register 2 (CR2)

Address(es): SCI12.CR2 0008 B323h

	b7	b6	b5	b4	b3	b2	b1	b0
	RTS[1:0]		BCCS[1:0]		—	DFCS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DFCS[2:0]	RXDX12 Signal Digital Filter Clock Select	b2 b0 0 0 0: Filter is disabled. 0 0 1: Filter clock is SCI base clock*1, *2 0 1 0: Filter clock is PCLK/8 0 1 1: Filter clock is PCLK/16 1 0 0: Filter clock is PCLK/32 1 0 1: Filter clock is PCLK/64 1 1 0: Filter clock is PCLK/128 1 1 1: Setting prohibited	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	BCCS[1:0]	Bus Collision Detection Clock Select	• When SEMR.BGDM = 0 or SEMR.BGDM = 1 and SMR.CKS[1:0] = a value other than 00b b5 b4 0 0: SCI base clock 0 1: SCI base clock frequency divided by 2 1 0: SCI base clock frequency divided by 4 1 1: Setting prohibited • When SEMR.BGDM = 1 and SMR.CKS[1:0] = 00b b5 b4 0 0: SCI base clock frequency divided by 2 0 1: SCI base clock frequency divided by 4 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b7, b6	RTS[1:0]	RXDX12 Reception Sampling Timing Select	• When SCI12.SEMR.ABCS = 0 b7 b6 0 0: Rising edge of the 8th cycle of SCI base clock 0 1: Rising edge of the 10th cycle of SCI base clock 1 0: Rising edge of the 12th cycle of SCI base clock 1 1: Rising edge of the 14th cycle of SCI base clock • When SCI12.SEMR.ABCS = 1 b7 b6 0 0: Rising edge of the 4th cycle of SCI base clock 0 1: Rising edge of the 5th cycle of SCI base clock 1 0: Rising edge of the 6th cycle of SCI base clock 1 1: Rising edge of the 7th cycle of SCI base clock	R/W

Note: The period of the SCI base clock is 1/16 of a single bit period when the SCI12.SEMR.ABCS is 0, and 1/8 of a single bit period when the SCI12.SEMR.ABCS is 1.
Note 1. To use the SCI base clock, set the SCI12.SCR.TE bit to 1.
Note 2. The SCI base clock divided by 2 is the filter clock when the SEMR.BGDM bit is 1 and the SMR.CKS[1:0] bits are 00b.

40.2.24 Control Register 3 (CR3)

Address(es): SCI12.CR3 0008 B324h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SDST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SDST	Start Frame Detection Start	0: Detection of Start Frame is not performed. 1: Detection of Start Frame is performed.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SDST Bit (Start Frame Detection Start)

Detection of a Start Frame begins when this bit is set to 1. The bit is read as 0.

40.2.25 Port Control Register (PCR)

Address(es): SCI12.PCR 0008 B325h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SHARPS	—	—	RXDXPS	TXDXPS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXDXPS	TXDX12 Signal Polarity Select	0: The polarity of TXDX12 signal is not inverted for output. 1: The polarity of TXDX12 signal is inverted for output.	R/W
b1	RXDXPS	RXDX12 Signal Polarity Select	0: The polarity of RXDX12 signal is not inverted for input. 1: The polarity of RXDX12 signal is inverted for input.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SHARPS	TXDX12/RXDX12 Pin Multiplexing Select	0: The TXDX12 and RXDX12 pins are independent. 1: The TXDX12 and RXDX12 signals are multiplexed on the same pin.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SHARPS Bit (TXDX12/RXDX12 Pin Multiplexing Select)

When this bit is set to 1, the TXDX12 and RXDX12 signals are multiplexed on the same pin so that half-duplex communications become possible.

40.2.26 Interrupt Control Register (ICR)

Address(es): SCI12.ICR 0008 B326h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AEDIE	BCDIE	PIBDIE	CF1MIE	CF0MIE	BFDIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFDIE	Break Field Low Width Detected Interrupt Enable	0: Interrupts on detection of the low width for a Break Field are disabled. 1: Interrupts on detection of the low width for a Break Field are enabled.	R/W
b1	CF0MIE	Control Field 0 Match Detected Interrupt Enable	0: Interrupts on detection of a match with Control Field 0 are disabled. 1: Interrupts on detection of a match with Control Field 0 are enabled.	R/W
b2	CF1MIE	Control Field 1 Match Detected Interrupt Enable	0: Interrupts on detection of a match with Control Field 1 are disabled. 1: Interrupts on detection of a match with Control Field 1 are enabled.	R/W
b3	PIBDIE	Priority Interrupt Bit Detected Interrupt Enable	0: Interrupts on detection of the priority interrupt bit are disabled. 1: Interrupts on detection of the priority interrupt bit are enabled.	R/W
b4	BCDIE	Bus Collision Detected Interrupt Enable	0: Interrupts on detection of a bus collision are disabled. 1: Interrupts on detection of a bus collision are enabled.	R/W
b5	AEDIE	Valid Edge Detected Interrupt Enable	0: Interrupts on detection of a valid edge are disabled. 1: Interrupts on detection of a valid edge are enabled.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

40.2.27 Status Register (STR)

Address(es): SCI12.STR 0008 B327h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AEDF	BCDF	PIBDF	CF1MF	CF0MF	BFDF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFDF	Break Field Low Width Detection Flag	[Setting conditions] • Detection of the low width for a Break Field • Completion of the output of the low width for a Break Field • Underflow of the timer [Clearing condition] • Writing 1 to the BFDCL bit in STCR	R
b1	CF0MF	Control Field 0 Match Flag	[Setting condition] • A match between the value received in Control Field 0 and the set value. [Clearing condition] • Writing 1 to the CF0MCL bit in STCR	R
b2	CF1MF	Control Field 1 Match Flag	[Setting condition] • A match between the data received in Control Field 1 and the set values. [Clearing condition] • Writing 1 to the CF1MCL bit in STCR	R
b3	PIBDF	Priority Interrupt Bit Detection Flag	[Setting condition] • Detection of the priority interrupt bit [Clearing condition] • Writing 1 to the PIBDCL bit in STCR	R
b4	BCDF	Bus Collision Detected Flag	[Setting condition] • Detection of the bus collision [Clearing condition] • Writing 1 to the BCDCL bit in STCR	R
b5	AEDF	Valid Edge Detection Flag	[Setting condition] • Detection of a valid edge [Clearing condition] • Writing 1 to the AEDCL bit in STCR	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

40.2.28 Status Clear Register (STCR)

Address(es): SCI12.STCR 0008 B328h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AEDCL	BCDCL	PIBDCL	CF1MCL	CF0MCL	BFDCL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFDCL	BFDF Clear	Setting this bit to 1 clears the STR.BFDF flag. This bit is read as 0.	R/W
b1	CF0MCL	CF0MF Clear	Setting this bit to 1 clears the STR.CF0MF flag. This bit is read as 0.	R/W
b2	CF1MCL	CF1MF Clear	Setting this bit to 1 clears the STR.CF1MF flag. This bit is read as 0.	R/W
b3	PIBDCL	PIBDF Clear	Setting this bit to 1 clears the STR.PIBDF flag. This bit is read as 0.	R/W
b4	BCDCL	BCDF Clear	Setting this bit to 1 clears the STR.BCDF flag. This bit is read as 0.	R/W
b5	AEDCL	AEDF Clear	Setting this bit to 1 clears the STR.AEDF flag. This bit is read as 0.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

40.2.29 Control Field 0 Data Register (CF0DR)

Address(es): SCI12.CF0DR 0008 B329h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

The CF0DR register is an 8-bit readable and writable register that holds a value for comparison with Control Field 0.

40.2.30 Control Field 0 Compare Enable Register (CF0CR)

Address(es): SCI12.CF0CR 0008 B32Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	CF0CE7	CF0CE6	CF0CE5	CF0CE4	CF0CE3	CF0CE2	CF0CE1	CF0CE0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CF0CE0	Control Field 0 Bit 0 Compare Enable	0: Comparison with bit 0 of Control Field 0 is disabled. 1: Comparison with bit 0 of Control Field 0 is enabled.	R/W
b1	CF0CE1	Control Field 0 Bit 1 Compare Enable	0: Comparison with bit 1 of Control Field 0 is disabled. 1: Comparison with bit 1 of Control Field 0 is enabled.	R/W
b2	CF0CE2	Control Field 0 Bit 2 Compare Enable	0: Comparison with bit 2 of Control Field 0 is disabled. 1: Comparison with bit 2 of Control Field 0 is enabled.	R/W
b3	CF0CE3	Control Field 0 Bit 3 Compare Enable	0: Comparison with bit 3 of Control Field 0 is disabled. 1: Comparison with bit 3 of Control Field 0 is enabled.	R/W
b4	CF0CE4	Control Field 0 Bit 4 Compare Enable	0: Comparison with bit 4 of Control Field 0 is disabled. 1: Comparison with bit 4 of Control Field 0 is enabled.	R/W
b5	CF0CE5	Control Field 0 Bit 5 Compare Enable	0: Comparison with bit 5 of Control Field 0 is disabled. 1: Comparison with bit 5 of Control Field 0 is enabled.	R/W
b6	CF0CE6	Control Field 0 Bit 6 Compare Enable	0: Comparison with bit 6 of Control Field 0 is disabled. 1: Comparison with bit 6 of Control Field 0 is enabled.	R/W
b7	CF0CE7	Control Field 0 Bit 7 Compare Enable	0: Comparison with bit 7 of Control Field 0 is disabled. 1: Comparison with bit 7 of Control Field 0 is enabled.	R/W

40.2.31 Control Field 0 Receive Data Register (CF0RR)

Address(es): SCI12.CF0RR 0008 B32Bh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

CF0RR is a readable register that holds the value received in Control Field 0. Writing to this register from the CPU or DTC is not possible.

40.2.32 Primary Control Field 1 Data Register (PCF1DR)

Address(es): SCI12.PCF1DR 0008 B32Ch

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

PCF1DR is an 8-bit readable and writable register that holds the 8-bit primary value for comparison with Control Field 1.

40.2.33 Secondary Control Field 1 Data Register (SCF1DR)

Address(es): SCI12.SCF1DR 0008 B32Dh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

PCF1DR is an 8-bit readable and writable register that holds the 8-bit secondary value for comparison with Control Field 1.

40.2.34 Control Field 1 Compare Enable Register (CF1CR)

Address(es): SCI12.CF1CR 0008 B32Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	CF1CE7	CF1CE6	CF1CE5	CF1CE4	CF1CE3	CF1CE2	CF1CE1	CF1CE0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CF1CE0	Control Field 1 Bit 0 Compare Enable	0: Comparison with bit 0 of Control Field 1 is disabled. 1: Comparison with bit 0 of Control Field 1 is enabled.	R/W
b1	CF1CE1	Control Field 1 Bit 1 Compare Enable	0: Comparison with bit 1 of Control Field 1 is disabled. 1: Comparison with bit 1 of Control Field 1 is enabled.	R/W
b2	CF1CE2	Control Field 1 Bit 2 Compare Enable	0: Comparison with bit 2 of Control Field 1 is disabled. 1: Comparison with bit 2 of Control Field 1 is enabled.	R/W
b3	CF1CE3	Control Field 1 Bit 3 Compare Enable	0: Comparison with bit 3 of Control Field 1 is disabled. 1: Comparison with bit 3 of Control Field 1 is enabled.	R/W
b4	CF1CE4	Control Field 1 Bit 4 Compare Enable	0: Comparison with bit 4 of Control Field 1 is disabled. 1: Comparison with bit 4 of Control Field 1 is enabled.	R/W
b5	CF1CE5	Control Field 1 Bit 5 Compare Enable	0: Comparison with bit 5 of Control Field 1 is disabled. 1: Comparison with bit 5 of Control Field 1 is enabled.	R/W
b6	CF1CE6	Control Field 1 Bit 6 Compare Enable	0: Comparison with bit 6 of Control Field 1 is disabled. 1: Comparison with bit 6 of Control Field 1 is enabled.	R/W
b7	CF1CE7	Control Field 1 Bit 7 Compare Enable	0: Comparison with bit 7 of Control Field 1 is disabled. 1: Comparison with bit 7 of Control Field 1 is enabled.	R/W

40.2.35 Control Field 1 Receive Data Register (CF1RR)

Address(es): SCI12.CF1RR 0008 B32Fh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

CF1RR is a readable register that holds the value received in Control Field 1. Writing to this register from the CPU or DTC is not possible.

40.2.36 Timer Control Register (TCR)

Address(es): SCI12.TCR 0008 B330h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TCST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TCST	Timer Count Start	0: Stops the timer counting 1: Starts the timer counting	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

40.2.37 Timer Mode Register (TMR)

Address(es): SCI12.TMR 0008 B331h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	TCSS[2:0]			TWRC	—	TOMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOMS[1:0]	Timer Operating Mode Select*1	b1 b0 0 0: Timer mode 0 1: Break Field low width determination mode 1 0: Break Field low width output mode 1 1: Setting prohibited	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	TWRC	Counter Write Control	0: Data is written to the reload register and counter 1: Data is written to the reload register only	R/W
b6 to b4	TCSS[2:0]	Timer Count Clock Source Select*1	b6 b4 0 0 0: PCLK 0 0 1: PCLK/2 0 1 0: PCLK/4 0 1 1: PCLK/8 1 0 0: PCLK/16 1 0 1: PCLK/32 1 1 0: PCLK/64 1 1 1: PCLK/128	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Rewrite the TOMS[1:0] and TCSS[2:0] bits only when the timer is stopped (TCST = 0).

TWRC Bit (Counter Write Control)

This bit determines whether a value written to TPRE or TCNT is written to the reload register only or is written to both the reload register and the counter.

40.2.38 Timer Prescaler Register (TPRE)

Address(es): SCI12.TPRE 0008 B332h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

TPRE consists of an 8-bit reload register, a read buffer, and a counter, each of which has FFh as its initial value. The counter counts down in synchronization with the counter clock selected by the TMR.TCSS[2:0] bits, and is reloaded with the value in the reload register when it underflows. Underflows of this register provide the clock source to drive counting by the TCNT register. The reload register and read buffer are allocated to the same address. Data is written to the reload register in writing, and the counter value that has been transferred to the read buffer is returned in reading.
It takes one PCLK cycle to load a value from the reload register to the counter.

40.2.39 Timer Count Register (TCNT)

Address(es): SCI12.TCNT 0008 B333h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

TCNT consists of an 8-bit reload register, a read buffer, and a counter, each of which has FFh as its initial value. This down-counter counts underflows of the TPRE register until the TCNT register underflows, and is then reloaded with the value from the reload register. The reload register and read buffer are allocated to the same address. Data is written to the reload register in writing, and the counter value that has been transferred to the read buffer is returned in reading.
It takes one PCLK cycle to load a value from the reload register to the counter.

40.3 Operation in Asynchronous Mode

Figure 40.5 shows the general format for asynchronous serial communications.
One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level).
In asynchronous serial communications, the communications line is usually held in the mark state (high level).
The SCI monitors the communications line. When the SCI detects a low, it regards that as a start bit and starts serial communication.
Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

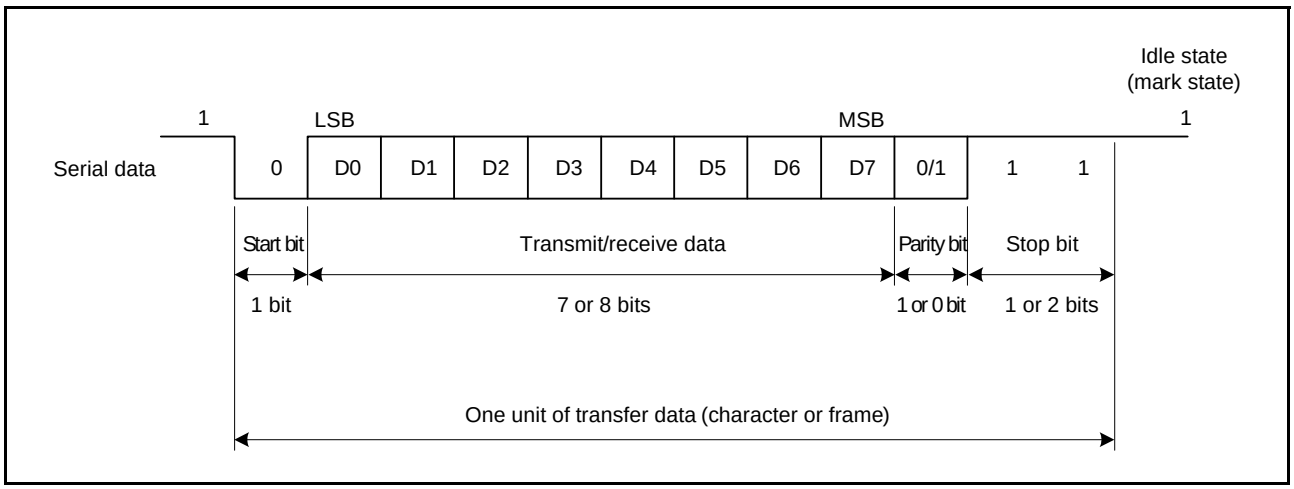


Figure 40.5 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, 2 Stop Bits)

40.3.1 Serial Data Transfer Format

Table 40.27 lists the serial data transfer formats that can be used in asynchronous mode.
Any of 18 transfer formats can be selected according to the SMR and SCMR setting. For details of multi-processor function, refer to section 40.4, Multi-Processor Communications Function.

Table 40.27 Serial Transfer Formats (Asynchronous Mode)

SCMR Setting	SMR Setting				Serial Transfer Format and Frame Length
CHR1	CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	0	0	0	S 9-bit data STOP
0	0	0	0	1	S 9-bit data STOP STOP
0	0	1	0	0	S 9-bit data P STOP
0	0	1	0	1	S 9-bit data P STOP STOP
1	0	0	0	0	S 8-bit data STOP
1	0	0	0	1	S 8-bit data STOP STOP
1	0	1	0	0	S 8-bit data P STOP
1	0	1	0	1	S 8-bit data P STOP STOP
1	1	0	0	0	S 7-bit data STOP
1	1	0	0	1	S 7-bit data STOP STOP
1	1	1	0	0	S 7-bit data P STOP
1	1	1	0	1	S 7-bit data P STOP STOP
0	0	—	1	0	S 9-bit data MPB STOP
0	0	—	1	1	S 9-bit data MPB STOP STOP
1	0	—	1	0	S 8-bit data MPB STOP
1	0	—	1	1	S 8-bit data MPB STOP STOP
1	1	—	1	0	S 7-bit data MPB STOP
1	1	—	1	1	S 7-bit data MPB STOP STOP

S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multi-processor bit

40.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a base clock with a frequency of 16 times*1 the bit rate.
In reception, the SCI samples the falling edge of the start bit using the base clock, and performs internal synchronization. Since receive data is sampled at the rising edge of the 8th pulse*1 of the base clock, data is latched at the middle of each bit, as shown in Figure 40.6. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\ F - \frac{D - 0.5}{N}\ (1 + F) \right| \times 100\ [\%] \cdots \text{Formula (1)}$$

M: Reception margin
N: Ratio of bit rate to clock (N = 16 when ABCS in SEMR = 0, N = 8 when ABCS in SEMR = 1)
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 13)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

M = {0.5 - 1/(2 × 16)} × 100 (%) = 46.875%

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

Note 1. This is an example when the ABCS bit in the SEMR register is 0. When the ABCS bit is 1, a frequency of 8 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 4th pulse of the base clock.

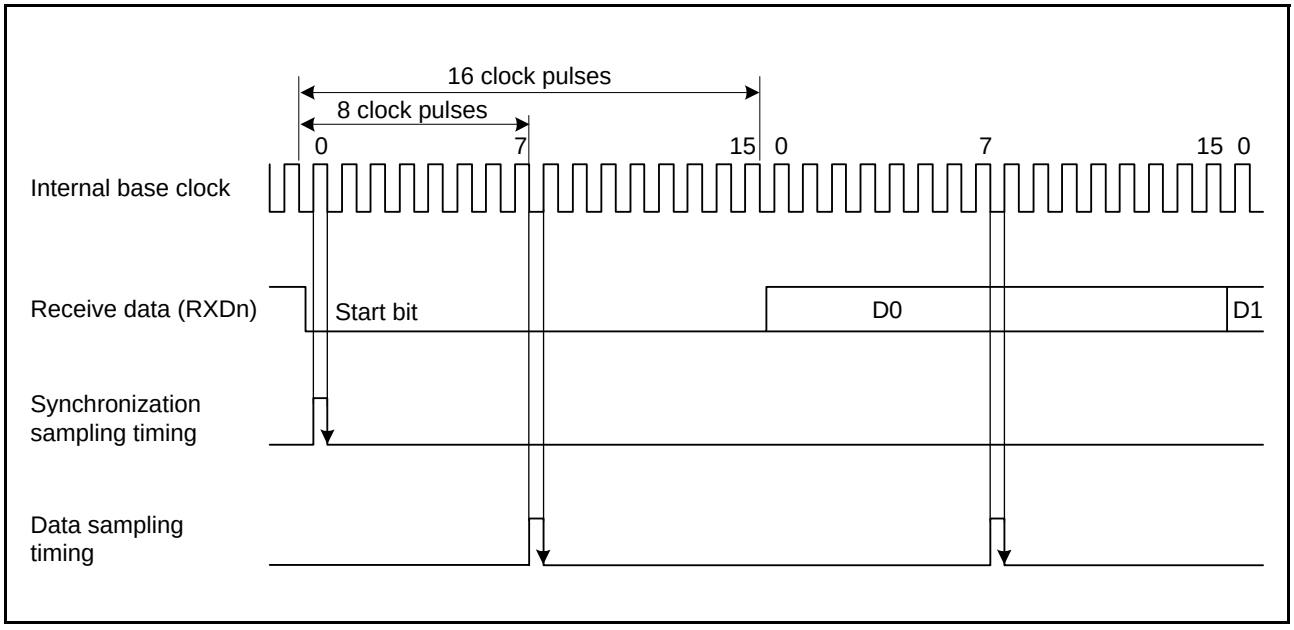


Figure 40.6 Receive Data Sampling Timing in Asynchronous Mode

40.3.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input to the SCKn pin can be selected as the SCI's transfer clock, according to the setting of the CM bit in the SMR register and the CKE[1:0] bits in the SCR register.

When an external clock is input to the SCKn pin, the clock frequency should be 16 times the bit rate (when SEMR.ABCS bit = 0) and 8 times the bit rate (when SEMR.ABCS bit = 1). In addition, when an external clock is specified, the base clock of TMR0 and TMR1 can be selected by the SCIn.SEMR.ACS0 bit (n = 5, 6, 12).

When the SCI is operated on an internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in Figure 40.7.

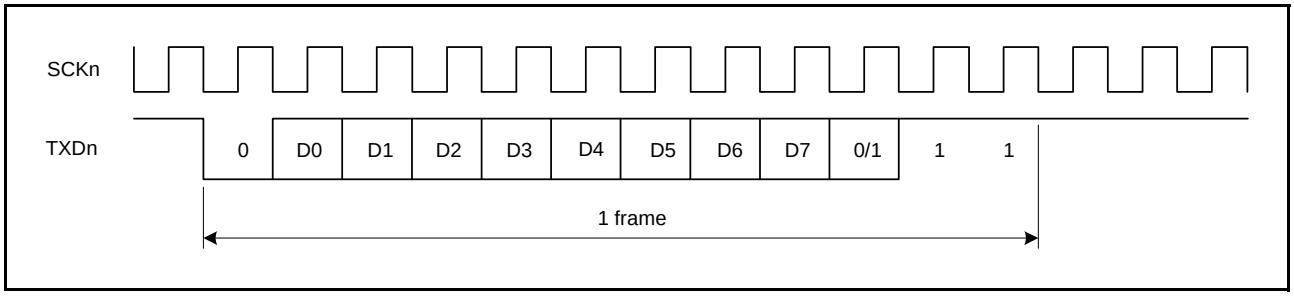


Figure 40.7 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode: SMR.CHR = 0, PE = 1, MP = 0, STOP = 1)

40.3.4 Double-Speed Mode

The output clock frequency of the on-chip baud rate generator is doubled by setting the SEMR.BGDM bit to 1, enabling high-speed communication at a doubled bit rate. If the SEMR.ABCS bit is set to 1 under the above condition, the number of base clock cycles changes from 16 to 8, so the bit rate becomes four times faster than the initial state.

As shown by Formula (1) in section 40.3.2, Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, setting the SEMR.ABCS bit to 1 changes the number of cycles to 8, and the sampling interval becomes longer. This causes the reception margin to decrease. Therefore, setting the SEMR.BGDM bit to 1 and the SEMR.ABCS bit to 0 is recommended instead of setting the SEMR.BGDM bit to 0 and the SEMR.ABCS bit to 1 for high-speed operation at a doubled bit rate.

40.3.5 CTS and RTS Functions

The CTS function is the use of input on the CTSn# pin in transmission control. Setting the SPMR.CTSE bit to 1 enables the CTS function. When the CTS function is enabled, placing the low level on the CTSn# pin causes transmission to start.
Applying the high level to the CTS# pin while transmission is in progress does not affect transmission of the current frame, which continues.
In the RTS function, by using the function of output on the RTSn# pin, a low level is output when reception becomes possible. Conditions for output of the low and high level are shown below.

[Conditions for low-level output]
When the following conditions are all satisfied

- The SCR.RE bit is 1
- Reception is not in progress
- There are no received data yet to be read
- The ORER, FER, and PER flags in the SSR are all 0

[Condition for high-level output]
When the conditions for low-level output are not satisfied

Note that either one of CTS and RTS can be selected.

40.3.6 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, start by writing the initial value 00h to the SCR register and then continue through the procedure for SCI given in Figure 40.8. Whenever the operating mode or transfer format is changed, the SCR register must be initialized before the change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied even during initialization.

Note that setting the SCR.RE bit to 0 initializes neither the ORER, FER, PER, and RDRF flags in the SSR register nor registers RDR, RDRH, and RDRL.

Moreover, note that changing the value of the SCR.TE bit from 1 to 0 or 0 to 1 while the SCR.TIE bit is 1 leads to the generation of a transmit data empty interrupt (TXI) request.

In addition, note that setting bits TIE, TE, and TEIE in the SCR register to 1 simultaneously leads to the generation of a transmit end interrupt (TEI) request before the generation of a TXI interrupt request.

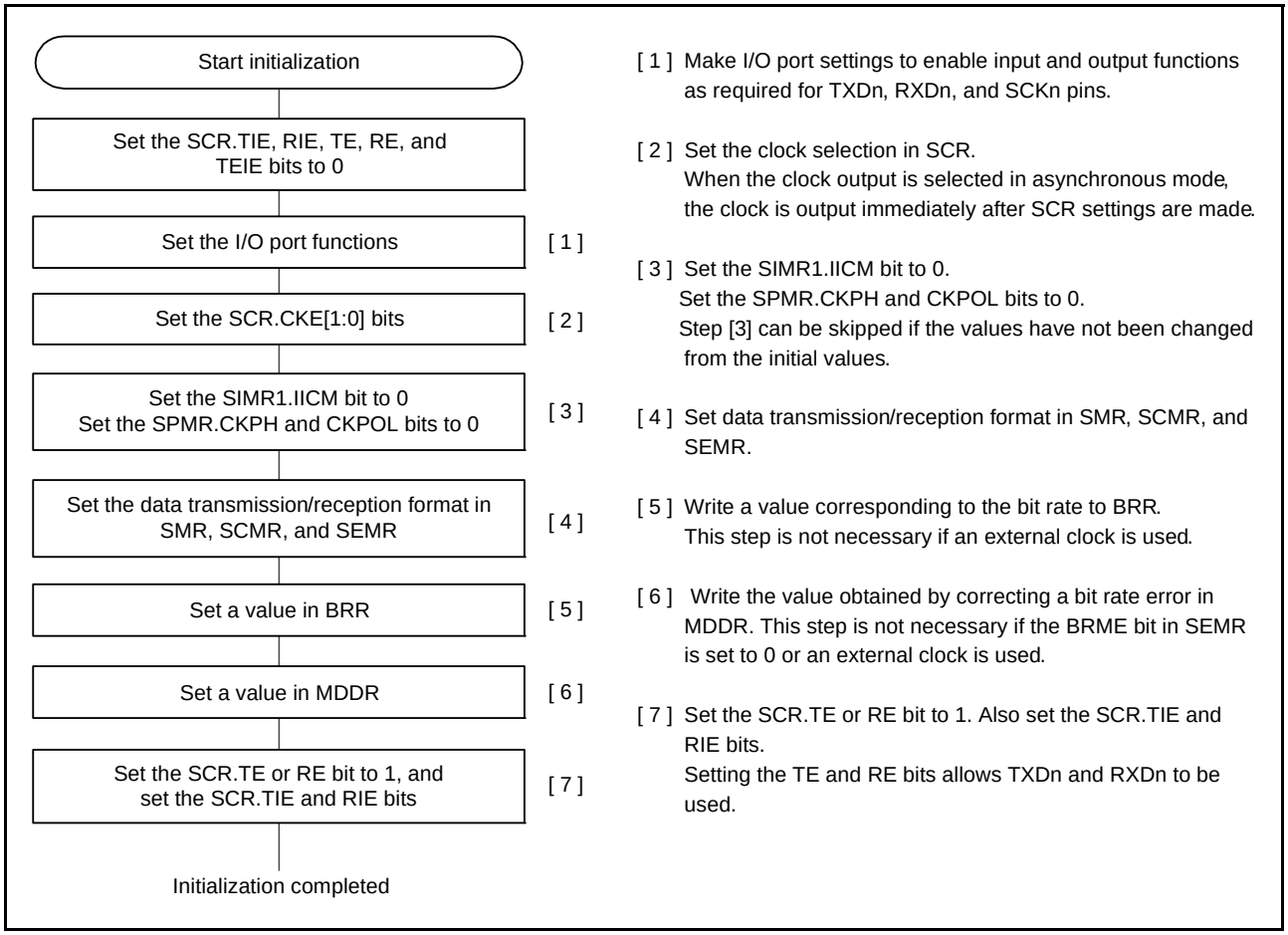


Figure 40.8 Sample SCI Initialization Flowchart (Asynchronous Mode)

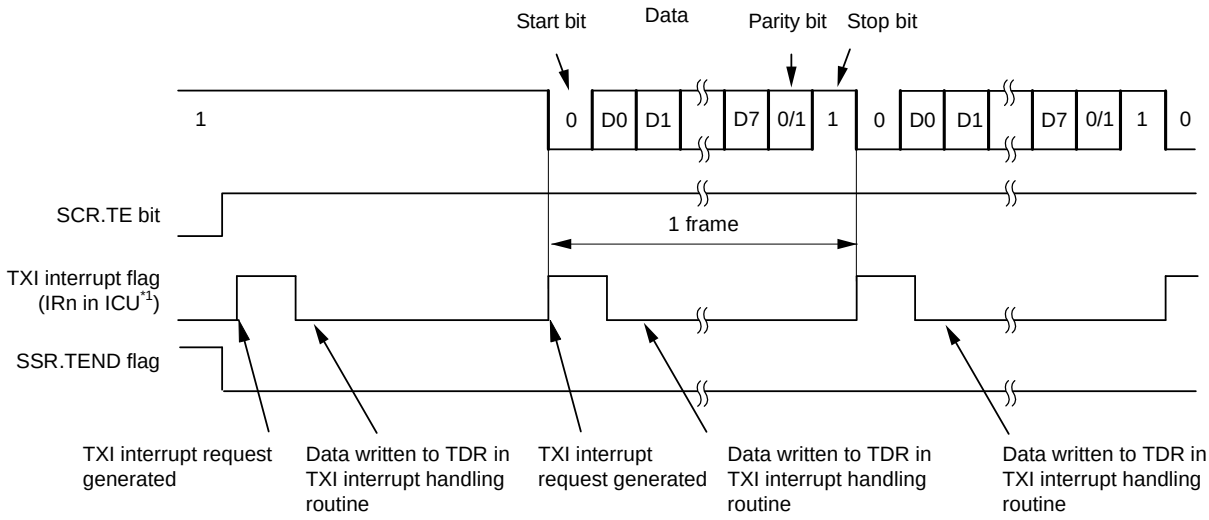
40.3.7 Serial Data Transmission (Asynchronous Mode)

Figure 40.9 to Figure 40.11 show an example of the operation for serial transmission in asynchronous mode.
In serial transmission, the SCI operates as described below.

1. The SCI transfers data from the TDR register*1 to the TSR register when data is written to the TDR register*1 in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the SCR.TE bit is set to 1 after the SCR.TIE bit is set to 1 or when these 2 bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the SPMR.CTSE bit is set to 0 (CTS function is disabled) and a low level on the CTSn# pin causes data transfer from the TDR register*1 to the TSR register. If the SCR.TIE bit is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to the TDR register*1 in the TXI interrupt handling routine before transmission of the current transmit data is completed. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (a TEI interrupt request is enabled) after the last of the data to be transmitted are written to the TDR register*1, *2 from the handling routine for TXI requests.
3. Data is sent from the TXDn pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks for updating of (writing to) the TDR register*3 at the time of stop bit output.
5. When the TDR register*3 is updated, setting of the SPMR.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn# pin cause the next transfer of the next transmit data from the TDR register*1 to the TSR register and sending of the stop bit, after which serial transmission of the next frame starts.
6. If the TDR register*3 is not updated, the SSR.TEND flag is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If the SCR.TEIE bit is 1 at this time, the SSR.TEND flag is set to 1 and a TEI interrupt request is generated.

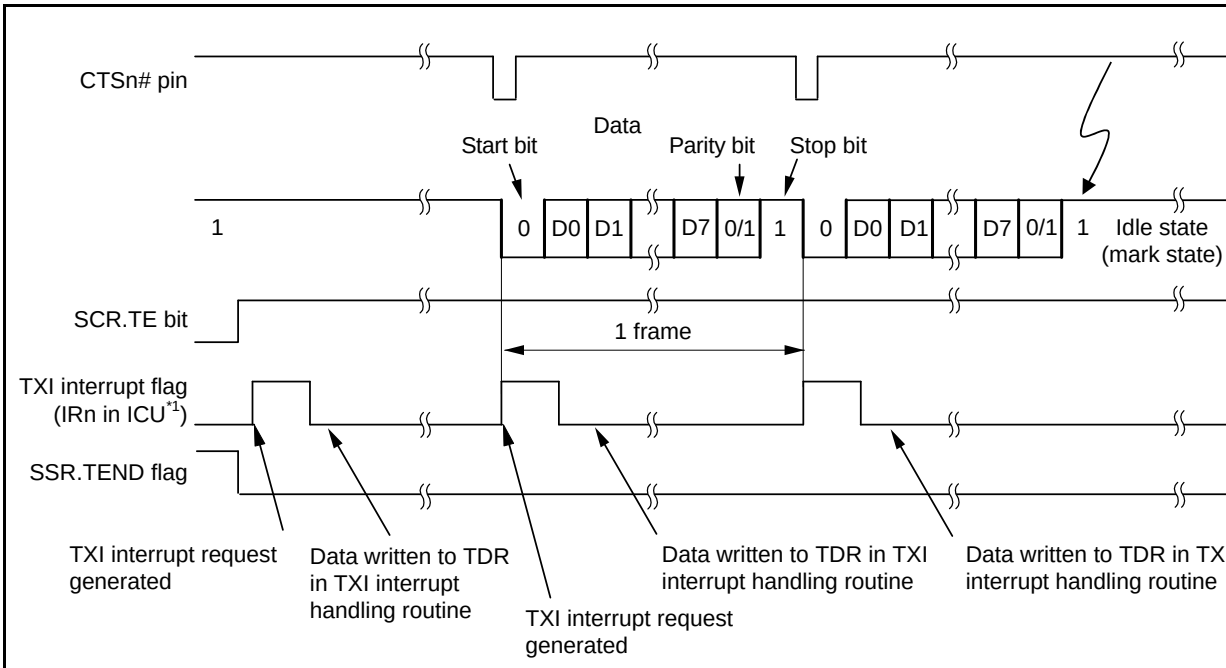
Note 1. Write data not to the TDR register but to the TDRH and TDRL registers when 9-bit data length is selected.
Note 2. Write data in the order from the TDRH register to the TDRL register when 9-bit data length is selected.
Note 3. The SCI checks for updating of the TDRL register only and does not check for updating of the TDRH register when 9-bit data length is selected.

Figure 40.12 shows a sample flowchart for serial transmission in asynchronous mode.



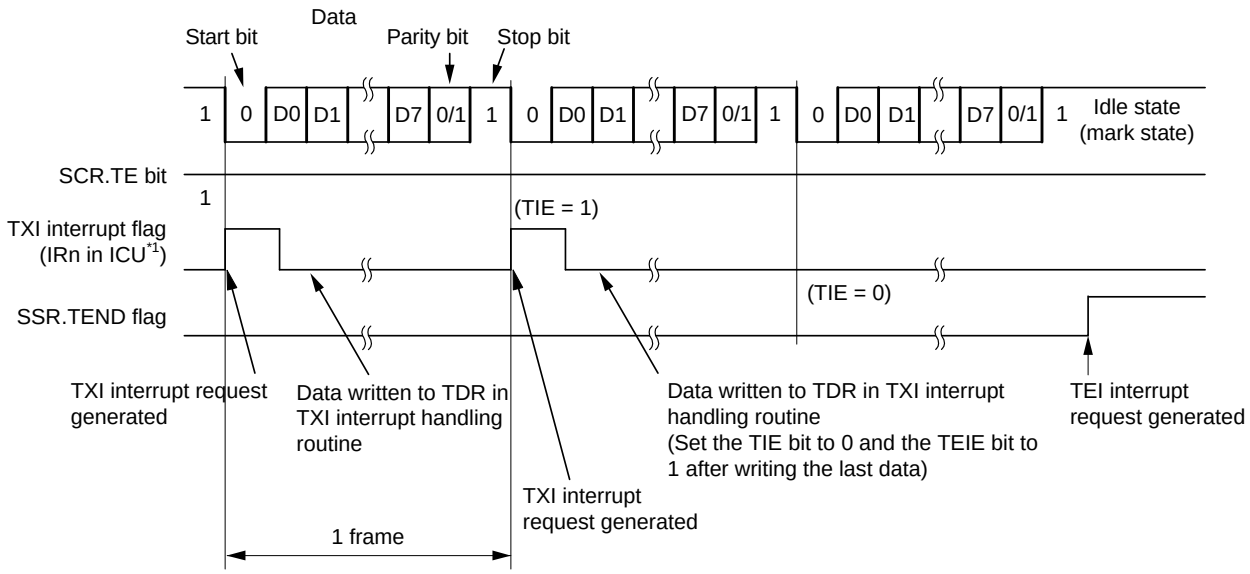
Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 40.9 Example of Operation for Serial Transmission in Asynchronous Mode (1) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, at the Beginning of Transmission)



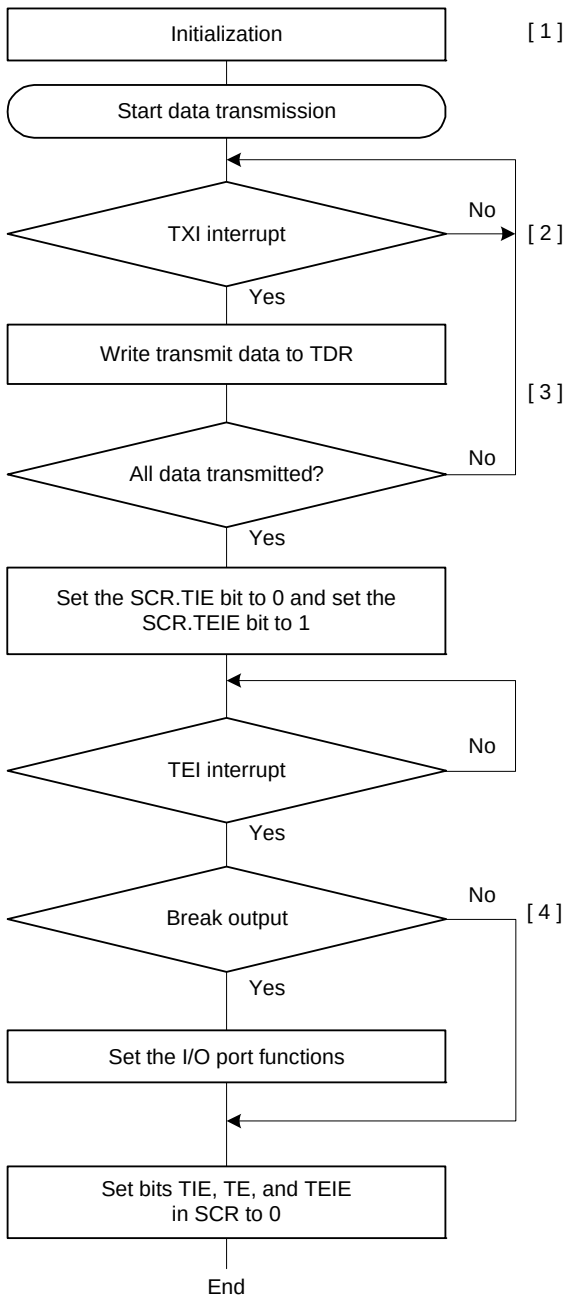
Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 40.10 Example of Operation for Serial Transmission in Asynchronous Mode (2) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Used, at the Beginning of Transmission)



Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 40.11 Example of Operation for Serial Transmission in Asynchronous Mode (3) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, from the Middle of Transmission until Transmission Completion)



[1] SCI initialization:
Set data transmission.
After the TE bit in SCR is set to 1, 1 is output for a frame, and transmission is enabled.

[2] Transmit data write to TDR by a TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (TXI) request is generated.
Write transmit data to TDR once in the TXI interrupt handling routine.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR once using a TXI interrupt request.
Transmit data can also be written to TDR by activating the DMAC or DTC.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the I/O port function corresponding to the TXDn pin (corresponding to output of the low level), and after switching the TXDn pin to the general I/O port function, set the TE bit in the SCR to 0.

Note: The TDR register becomes the TDRH and TDRL registers when 9-bit data length is selected. Write data in the order from TDRH to TDRL.

Figure 40.12 Example of Serial Transmission Flowchart in Asynchronous Mode

40.3.8 Serial Data Reception (Asynchronous Mode)

Figure 40.13 and Figure 40.14 show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the SCI operates as described below.

1. When the value of the SCR.RE bit becomes 1, the output signal on the RTSn# pin goes to the low level.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in the RSR register, and checks the parity bit and stop bit.
3. If an overrun error occurs, the SSR.ORER flag is set to 1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to the RDR register*1.
4. If a parity error is detected, the SSR.PER flag is set to 1 and receive data is transferred to the RDR register*1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, the SSR.FER flag is set to 1 and receive data is transferred to the RDR register*1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to the RDR register*1. If the SCR.RIE bit is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register*1 in this RXI interrupt handling routine before reception of the next receive data is completed. Reading the received data that have been transferred to the RDR register*1 causes the RTSn# pin to output the low level.

Note 1. Read data not in the RDR register but in the RDRH and RDRL registers when 9-bit data length is selected.
Note 2. The SCI checks for reading of the RDRL register only and does not check for reading of the RDRH register when 9-bit data length is selected.

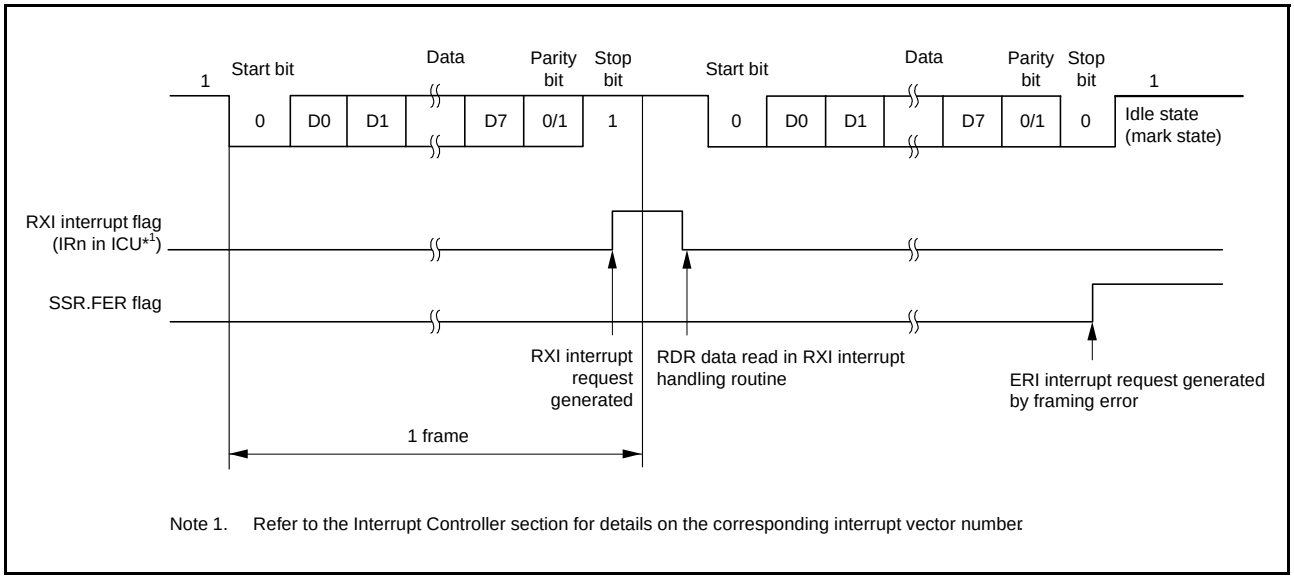
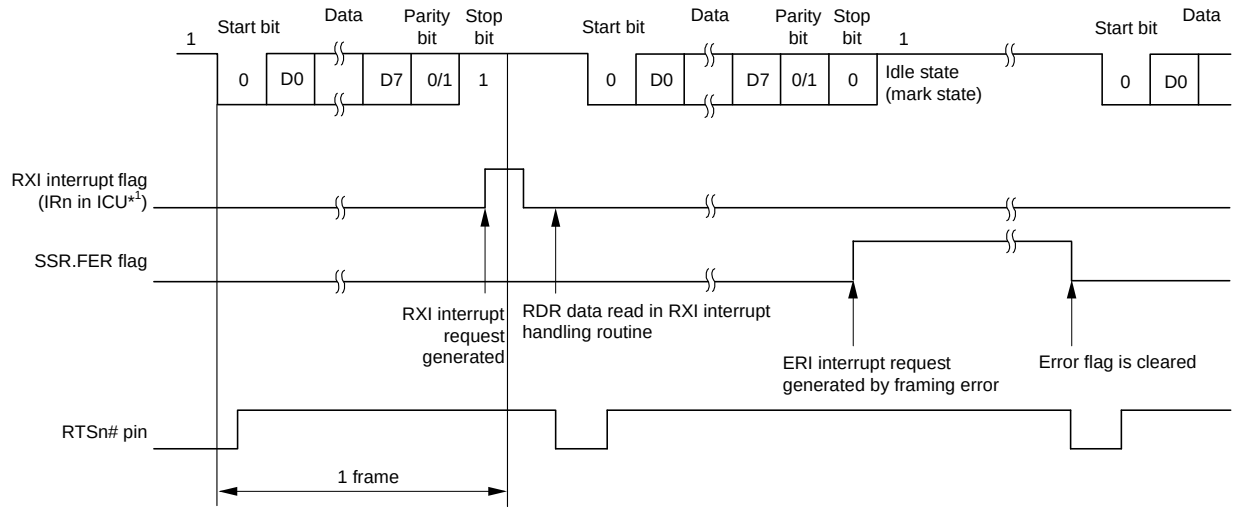


Figure 40.13 Example of SCI Operation for Serial Reception in Asynchronous Mode (1) (When RTS Function is Not Used) (Example with 8-Bit Data, Parity, 1 Stop Bit)



Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 40.14 Example of SCI Operation for Serial Reception in Asynchronous Mode (2) (When RTS Function is Used) (Example with 8-Bit Data, Parity, 1 Stop Bit)

Table 40.28 lists the states of the flags in the SSR status register and receive data handling when a receive error is detected.

If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, and PER flags to 0 before resuming reception. Moreover, be sure to read the RDR (or the RDRL) during overrun error processing. When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR (or the RDRL) register because received data which has not yet been read may be left in RDR (or the RDRL).

Figure 40.15 and Figure 40.16 show samples of flowcharts for serial data reception.

Table 40.28 Flags in the SSR Status Register and Receive Data Handling

Flags in the SSR Status Register				
ORER	**FER**	**PER**	**Receive Data**	**Receive Error Type**
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR*1	Framing error
0	0	1	Transferred to RDR*1	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR*1	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error

Note 1. Read data not in RDR but in the RDRH and RDRL registers when 9-bit data length is selected.

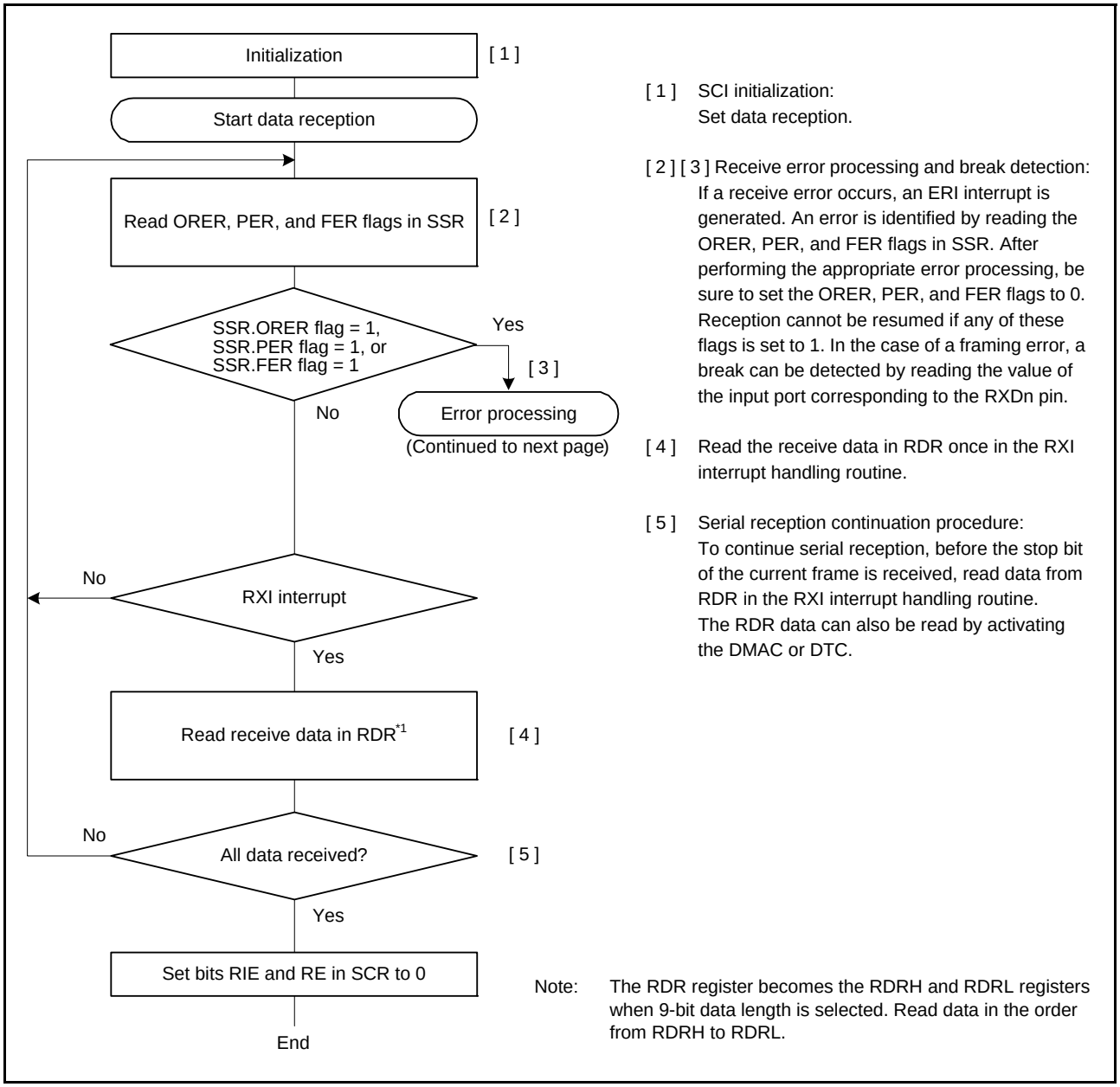


Figure 40.15 Example Flowchart of Serial Reception in Asynchronous Mode (1)

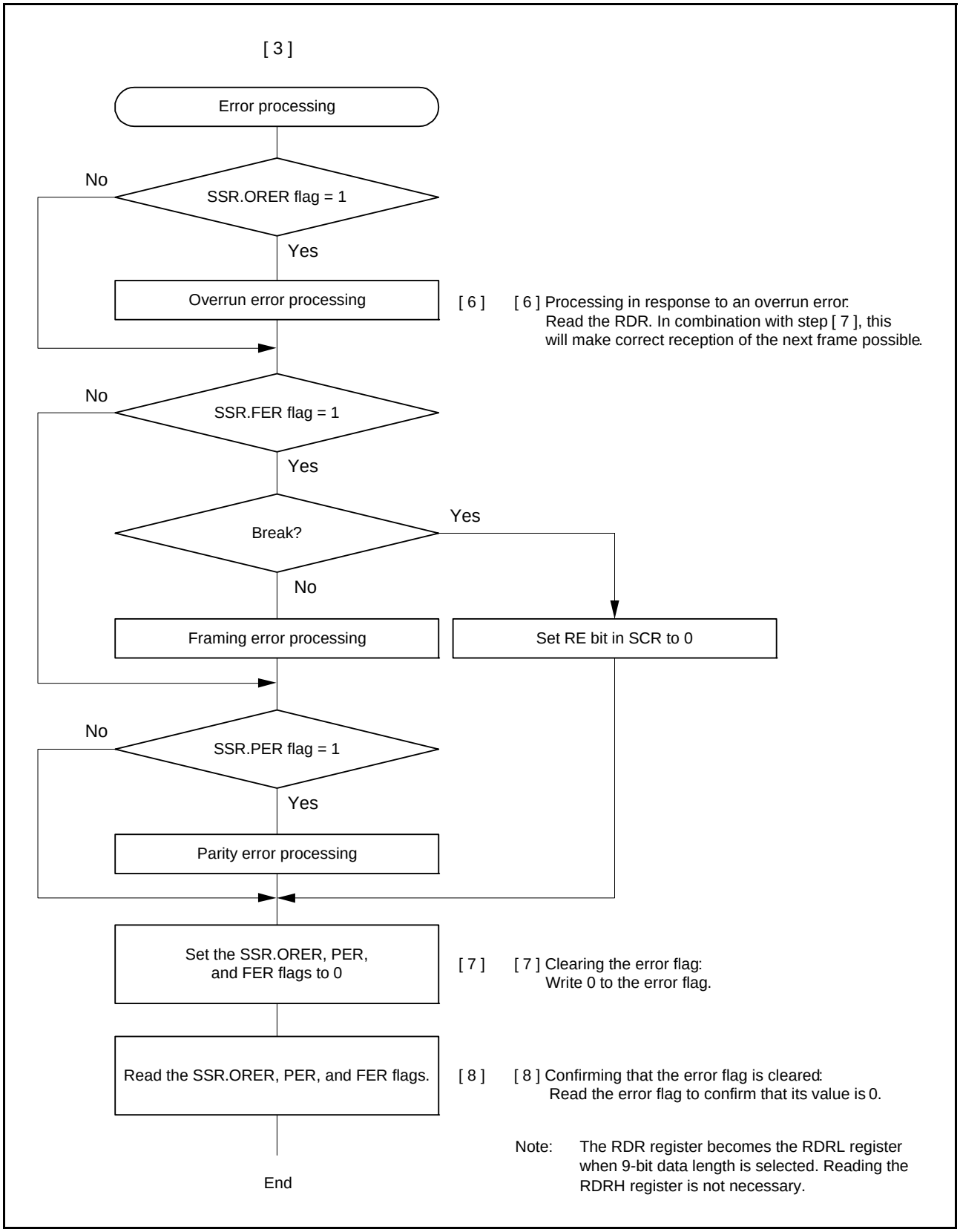


Figure 40.16 Example Flowchart of Serial Reception in Asynchronous Mode (2)

40.4 Multi-Processor Communications Function

Using the multi-processor communication functions enables to transmit and receive data by sharing a communication line between multiple processors by using asynchronous serial communication in which the multi-processor bit is added. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station. The multi-processor bit is used to distinguish between the ID transmission cycle and the data transmission cycle. When the multi-processor bit is set to 1, it indicates the ID transmission cycle and when the multi-processor bit is set to 0, it indicates the data transmission cycle. Figure 40.17 shows an example of communication between processors by using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits the communication data in which the multi-processor bit set to 0 is added to the transmit data. Upon receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself and if the two match, receives the communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until again receiving the communication data in which the multi-processor bit is set to 1.

For supporting this function, the SCI provides the SCR.MPIE bit. When the MPIE bit is set to 1, transfer of receive data from the RSR register to the RDR register (the RDRH and RDRL registers when 9-bit data length is selected), detection of a receive error, and setting the respective status flags ORER and FER in the SSR register are disabled until reception of data in which the multi-processor bit is set to 1. Upon receiving a reception character in which the multi-processor bit is set to 1, the SSR.MPB bit is set to 1 and the SCR.MPIE bit is automatically cleared, thus returning to a normal reception operation. During this time, an RXI interrupt is generated if the SCR.RIE bit is 1.

When the multi-processor format is specified, specification of the parity bit is disabled. Apart from this, there is no difference from the operation in the normal asynchronous mode. A clock which is used for the multi-processor communication is also the same as the clock used in the normal asynchronous mode.

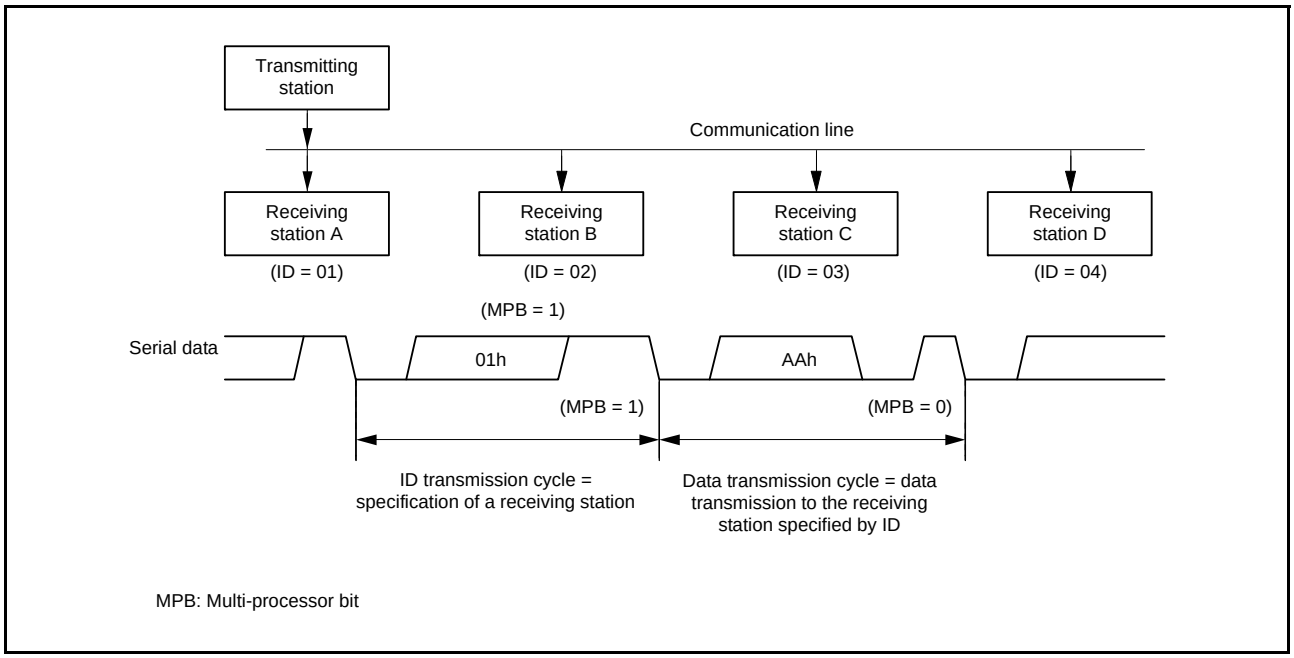
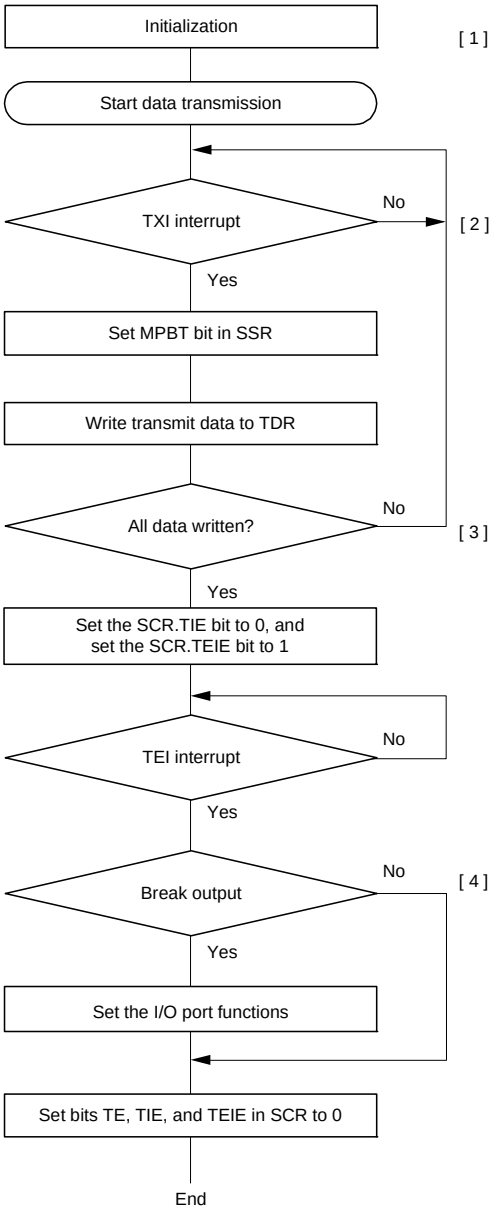


Figure 40.17 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)

40.4.1 Multi-Processor Serial Data Transmission

Figure 40.18 is a sample flowchart of multi-processor data transmission. In the ID transmission cycle, the ID should be transmitted with the SSR.MPBT bit set to 1. In the data transmission cycle, the data should be transmitted with the MPBT bit set to 0. The other operations are the same as the operations in asynchronous mode.



[1] SCI initialization:
Set data transmission.
After the TE bit in SCR is set to 1, 1 is output for a frame, and transmission is enabled.

[2] TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (TXI) request is generated.
Set the MPBT bit in SSR to 0 or 1, and write transmit data to TDR once in the TXI interrupt handling routine.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR once using a TXI interrupt request.
Transmit data can also be written to TDR by activating the DMAC or DTC.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR.

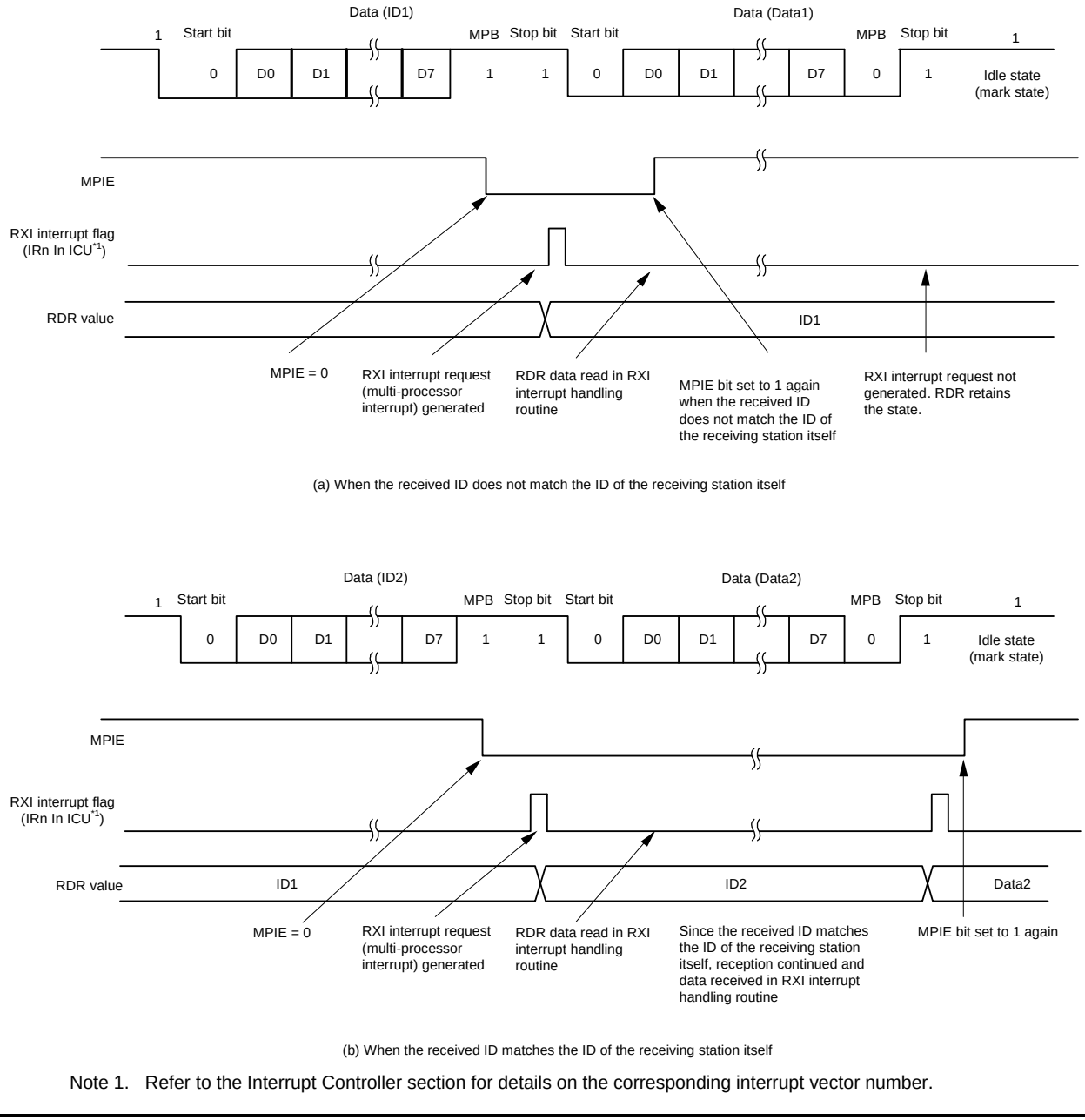
[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the I/O port function corresponding to the TXDn pin (corresponding to output of the low level), and after switching the TXDn pin to the general I/O port function, set the TE bit in the SCR to 0.

Note: The TDR register becomes the TDRH and TDRL registers when 9-bit data length is selected. Write data in the order from TDRH to TDRL.

Figure 40.18 Example of Multi-Processor Serial Transmission Flowchart

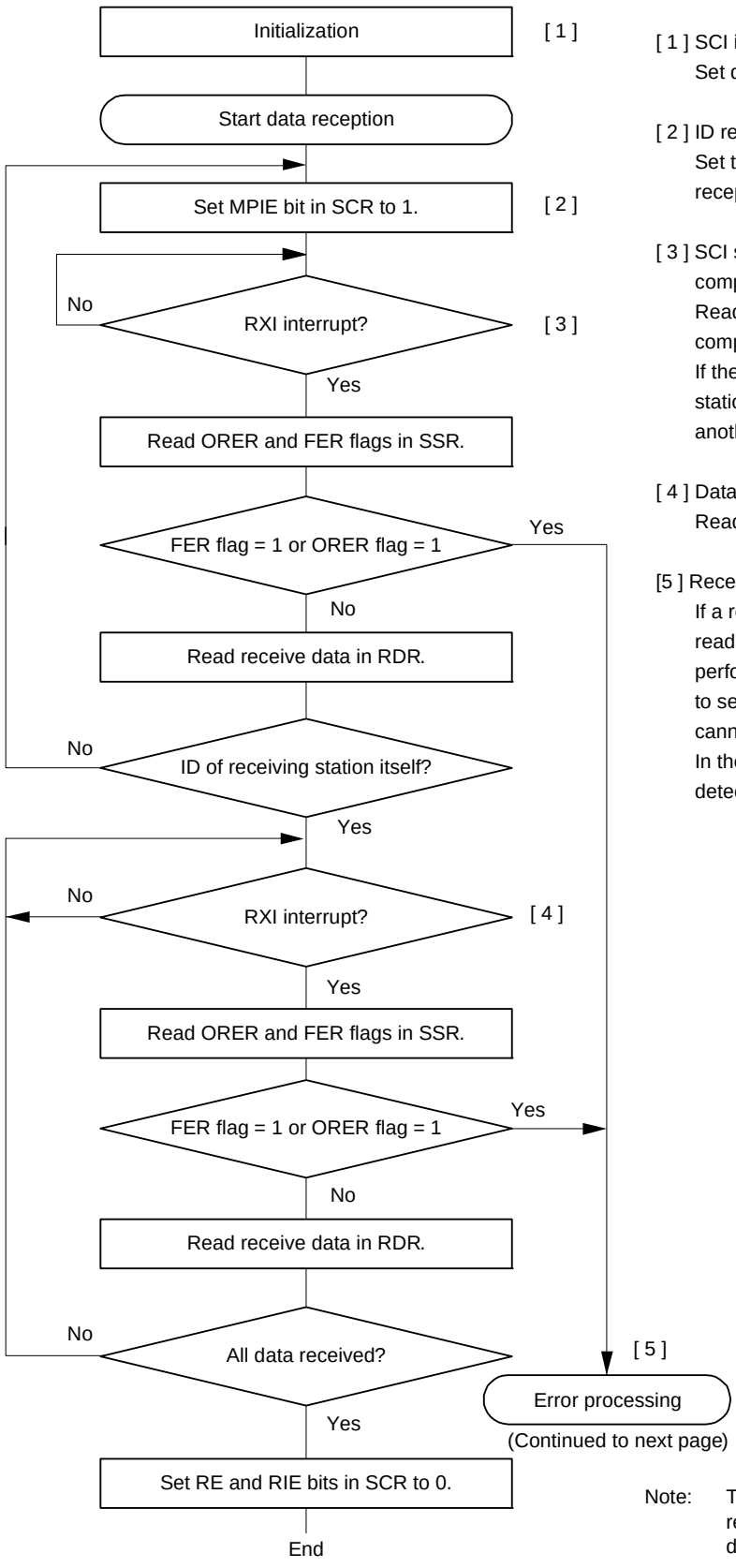
40.4.2 Multi-Processor Serial Data Reception

Figure 40.20 and Figure 40.21 are sample flowcharts of multi-processor data reception. When the SCR.MPIE bit is set to 1, reading the communication data is skipped until reception of the communication data in which the multi-processor bit is set to 1. When the communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to RDR (the RDRH and RDRL registers when 9-bit data length is selected). During this time, the RXI interrupt request is generated. The other operations are the same as the operations in asynchronous mode.
Figure 40.19 is the example of operation for reception.



Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 40.19 Example of SCI Reception (8-Bit Data/Multi-Processor Bit/1 Stop Bit)



[1] SCI initialization:
Set data reception.

[2] ID reception cycle:
Set the MPIE bit in SCR to 1 and wait for ID reception.

[3] SCI status confirmation and reception and comparison of ID:
Read data in RDR at the first RXI interrupt, and compare it with the ID of the receiving station itself. If the ID does not match the ID of the receiving station itself, set the MPIE bit to 1 again, and wait for another RXI interrupt RXI interrupt request.

[4] Data reception at an RXI interrupt:
Read data in RDR once in the RXI interrupt routine.

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading the ORER and FER flags in SSR. After performing the appropriate error processing, be sure to set the ORER and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading the value of the RXDn pin.

Note: The RDR register becomes the RDRH and RDRL registers when 9-bit data length is selected. Read data in the order from RDRH to RDRL.

Figure 40.20 Example of Multi-Processor Serial Reception Flowchart (1)

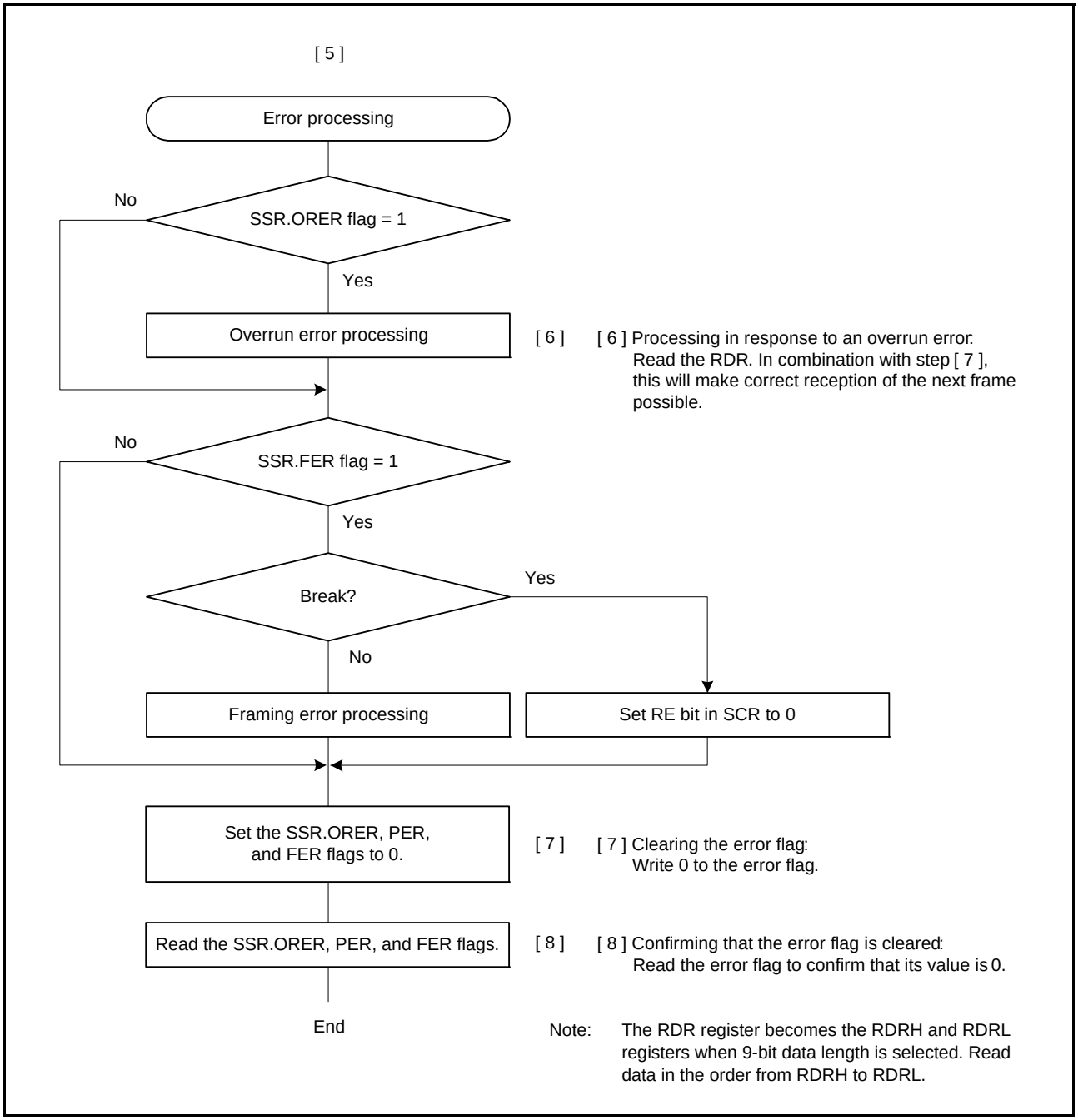


Figure 40.21 Example of Multi-Processor Serial Reception Flowchart (2)

40.5 Operation in Clock Synchronous Mode

Figure 40.22 shows the data format for clock synchronous serial data communications.
In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In clock synchronous mode, no parity bit can be added.
In data transmission, the SCI outputs data from one falling edge of the synchronization clock to the next. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the communication line holds the last bit output state.
Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.

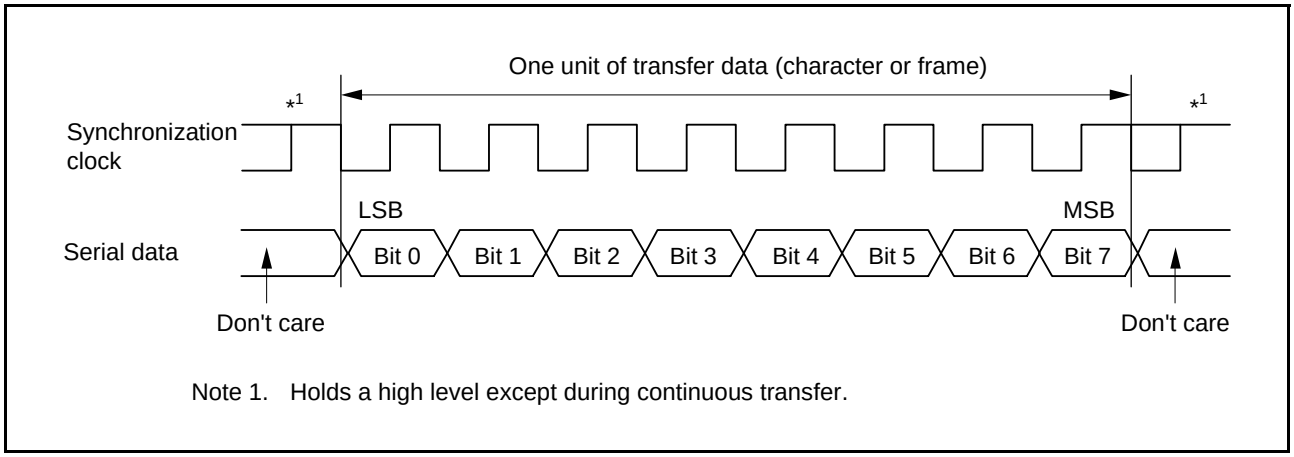


Figure 40.22 Data Format in Clock Synchronous Serial Communications (LSB First)

40.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCKn pin can be selected, according to the setting of the SCR.CKE[1:0] bits.
When the SCI is operated on an internal clock, the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is held high. However, when only data reception is performed while the CTS function is disabled, the synchronization clock output is started at the same time when the SCR.RE bit set to 1. The synchronization clock is stopped at the high level when an overrun error occurs or the SCR.RE bit is set to 0.
When only data reception is performed and the CTS function is enabled, the clock output is not started even when the SCR.RE bit set to 1 if the CTSn# pin input is high when the SCR.RE bit is 0. The synchronization clock output is started when the SCR.RE bit is set to 1 and the CTSn# pin input is low. After that, if the CTSn# pin input is high on completion of the frame reception, the synchronization clock output is stopped at the high level. If the CTSn# pin input continues to be low, the synchronization clock is stopped at the high level when an overrun error occurs or the SCR.RE bit is set to 0.

40.5.2 CTS and RTS Functions

In the CTS function, CTSn# pin input is used to control reception/transmission start when the clock source is the internal clock. Setting the SPMR.CTSE bit to 1 enables the CTS function.
When the CTS function is enabled, placing the low level on the CTSn# pin causes reception/transmission to start. Applying the high level to the CTS# pin while reception/transmission are in progress does not affect reception/transmission of the current frame, which continues.
In the RTS function, RTSn# pin output is used to request reception/transmission start when the clock source is an external synchronizing clock. A low level is output when serial communications become possible. Conditions for output of the low and high level are shown below.

[Conditions for low-level output]
When the following conditions are all satisfied

- The SCR.RE or SCR.TE bit is 1
- Transmission or reception of data is not in progress
- There are no received data yet to be read (when the SCR.RE bit is 1)
- Transmit data has been written (when the SCR.TE bit is 1)
- The SSR.ORER flag is 0

[Condition for high-level output]
The conditions for low-level output have not been satisfied.

40.5.3 SCI Initialization (Clock Synchronous Mode)

Before transmitting and receiving data, start by writing the initial value 00h to the SCR register and then continue through the procedure for SCI given in Figure 40.23. Whenever the operating mode or transfer format is changed, the SCR register must be initialized before the change is made.

Note that setting the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in the SSR register nor the RDR register.

Moreover, note that switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 while the SCR.TIE bit is 1 leads to the generation of a TXI interrupt request.

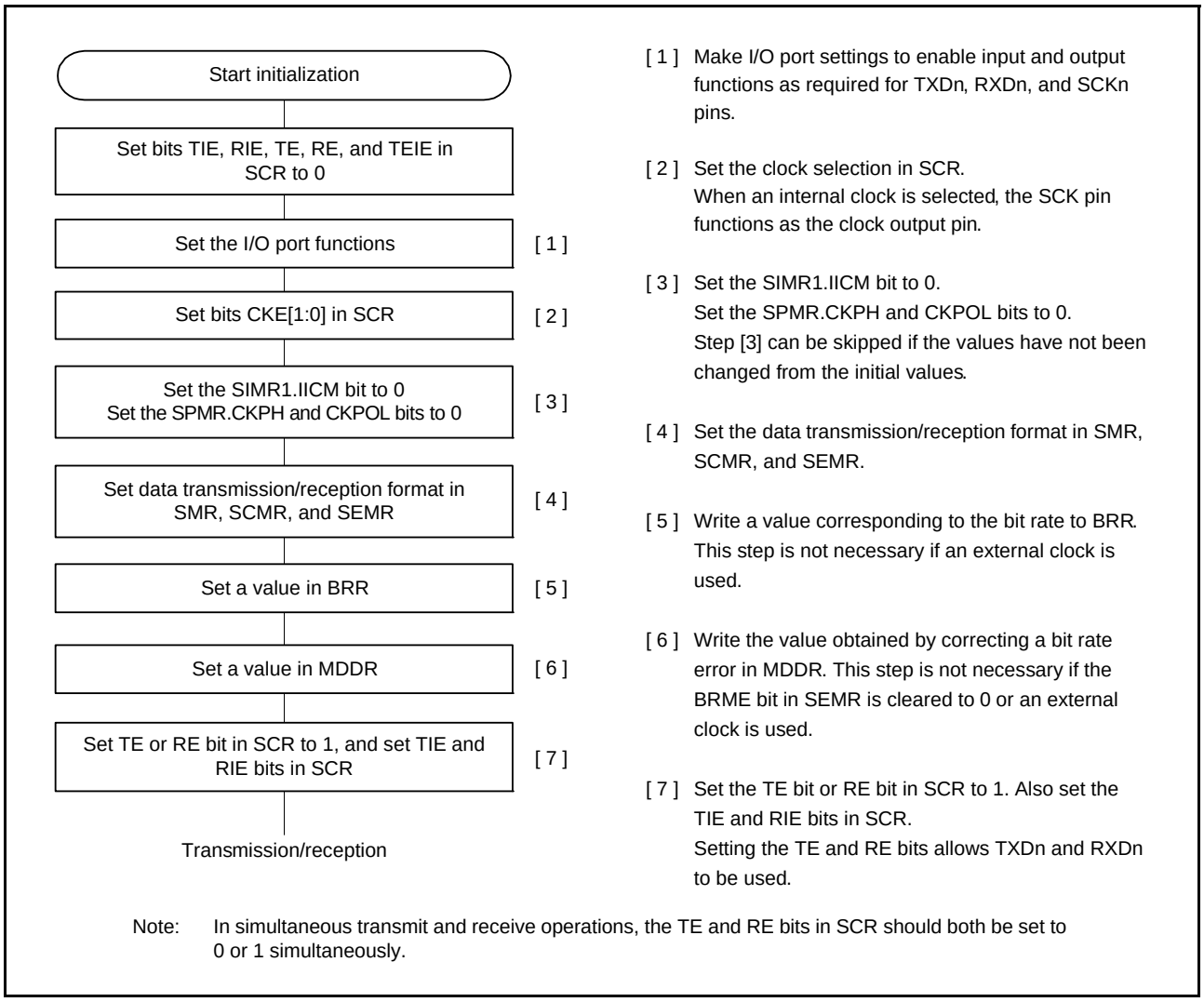


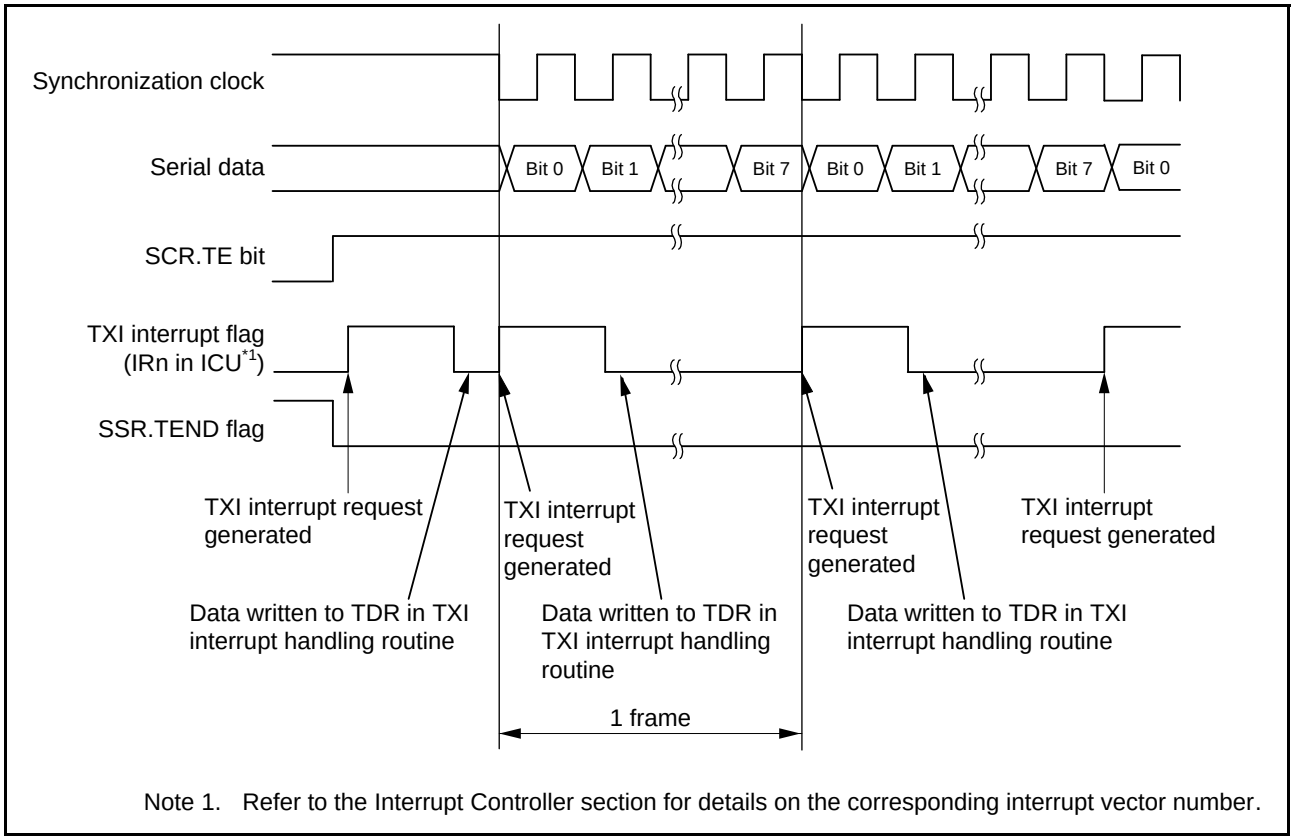
Figure 40.23 Example of SCI Initialization Flowchart (Clock Synchronous Mode)

40.5.4 Serial Data Transmission (Clock Synchronous Mode)

Figure 40.23, Figure 40.24, and Figure 40.25 show an example of the operation for serial transmission in clock synchronous mode.
In serial data transmission, the SCI operates as described below.

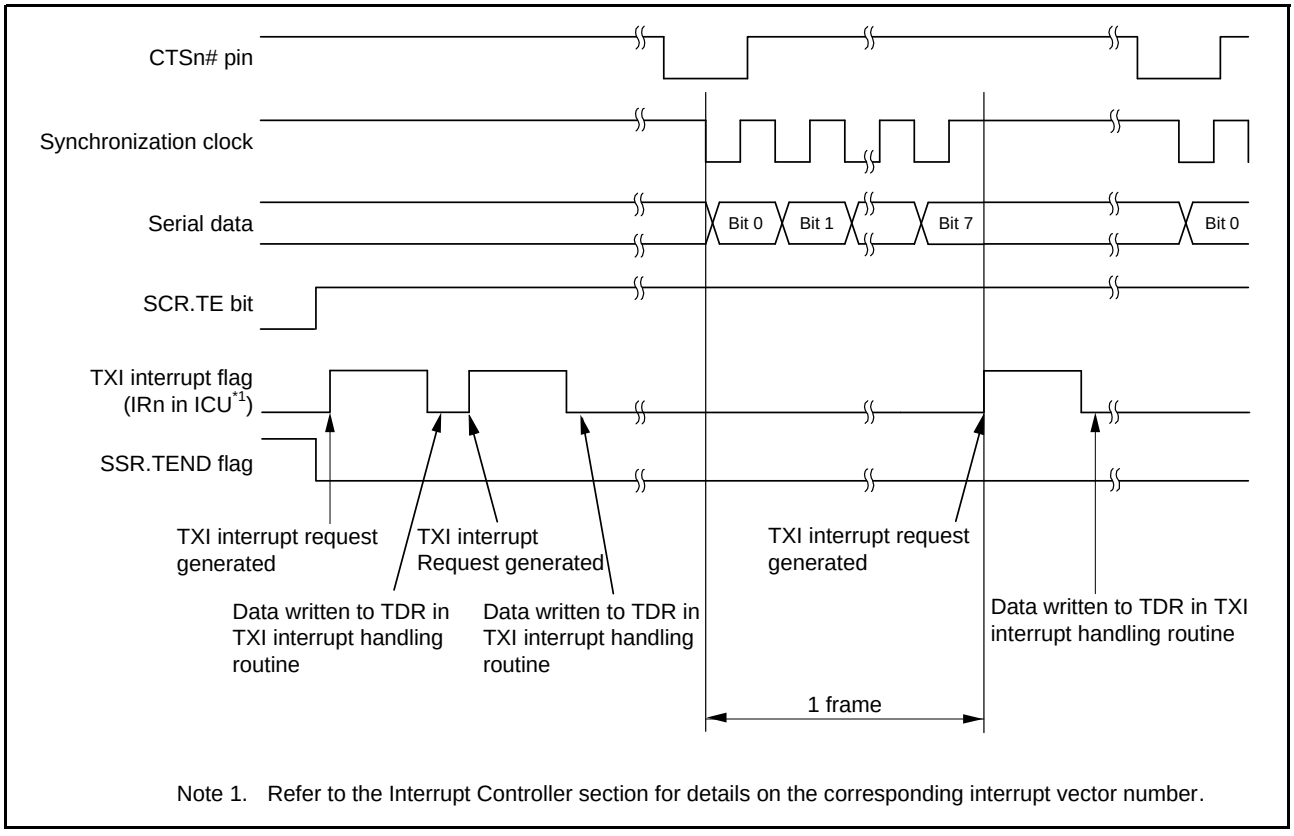
1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in the SCR register is set to 1 after the TIE bit in the SCR register is set to 1 or when these 2 bits are set to 1 simultaneously by a single instruction.
2. After transferring data from the TDR register to the TSR register, the SCI starts transmission. When the SCR.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to the TDR register in this TXI interrupt handling routine before transmission of the current transmit data has finished. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (a TEI interrupt request is enabled) after the last of the data to be transmitted are written to the TDR register from the handling routine for TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when clock output mode has been specified and in synchronization with the input clock when use of an external clock has been specified. Output of the clock signal is suspended until the input CTS signal is at the low level while the CTSE bit in the SPMR register is 1 (CTS function is enabled).
4. The SCI checks for updating of (writing to) the TDR register at the time of the last bit output.
5. When TDR is updated, the next transmit data is transferred from the TDR register to the TSR register, and serial transmission of the next frame is started.
6. If the TDR register is not updated, set the SSR.TEND flag to 1 and the TXDn pin retains the output state of the last bit. If the TEIE bit in the SCR register is 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Figure 40.27 shows a sample flowchart of serial data transmission.
Transmission will not start while a receive error flag (ORER, FER, or PER in the SSR register) is set to 1. Be sure to set the receive error flags to 0 before starting transmission. Note that setting the RE bit in the SCR register to 0 does not clear the receive error flags.



Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 40.24 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Not Used at the Beginning of Transmission



Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 40.25 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Used at the Beginning of Transmission

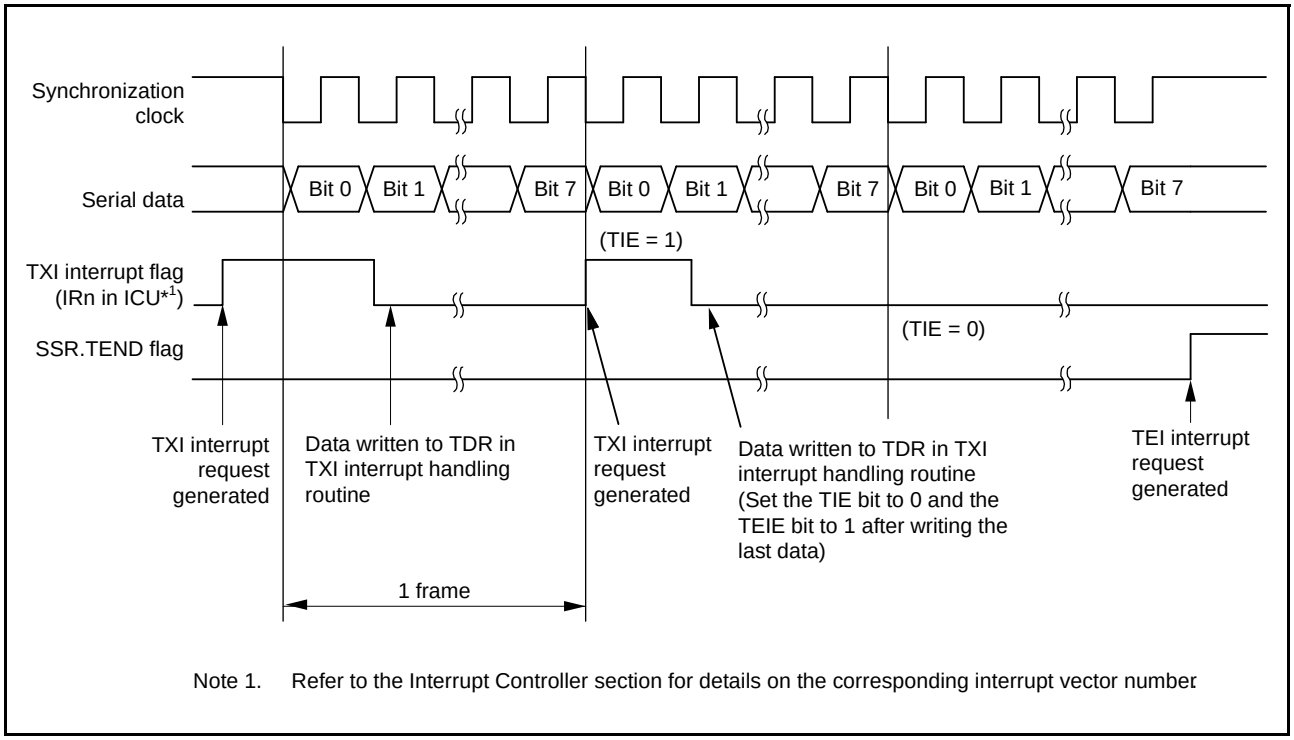
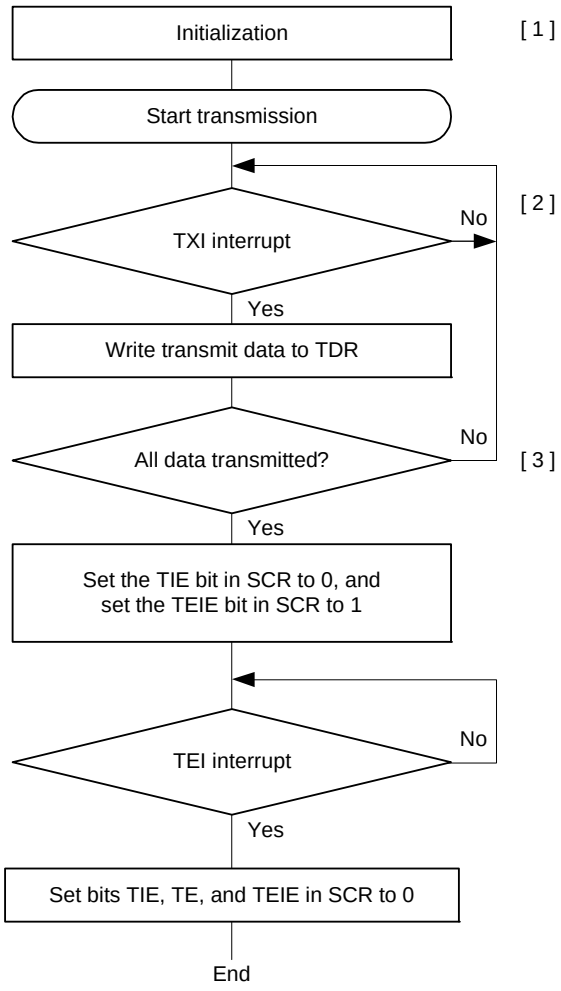


Figure 40.26 Example of Serial Data Transmission in Clock Synchronous Mode from the Middle of Transmission until Transmission Completion



[1] SCI initialization:
Set data transmission.

[2] Writing transmit data write to TDR by a TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (TXI) request is generated.
Transmit data is written to TDR once from the handling routine for TXI requests.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR upon accepting a transmit data empty interrupt (TXI) request. Transmit data can also be written to TDR by activating the DMAC or DTC by the TXI interrupt request.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR.

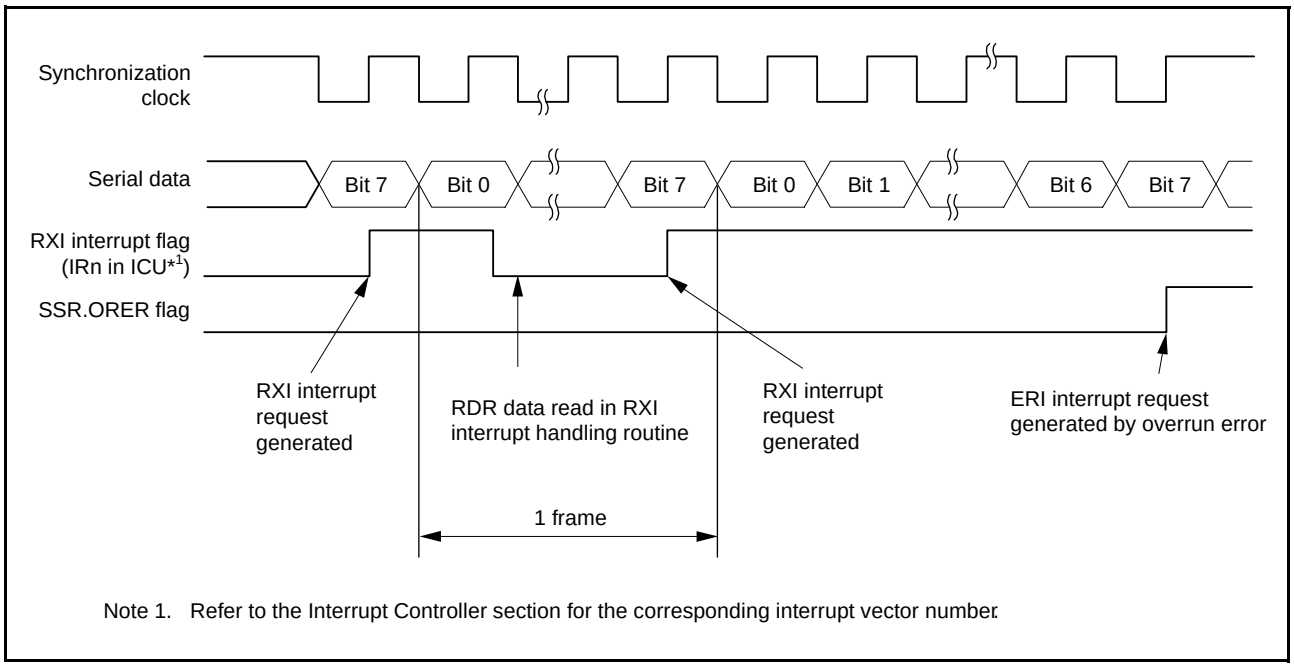
Note: When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR.TEND flag to 1. Setting the SCR.TE bit to 0 immediately after this may lead to insufficient received-data hold time on the receiver side.

Figure 40.27 Example Flowchart of Serial Transmission in Clock Synchronous Mode

40.5.5 Serial Data Reception (Clock Synchronous Mode)

Figure 40.28 and Figure 40.29 show an example of SCI operation for serial reception in clock synchronous mode. In serial data reception, the SCI operates as described below.

1. The value of the RE bit in the SCR register becoming 1 places the signal output on the RTSn# pin at the low level (when the RTS function is in use).
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in the RSR register.
3. If an overrun error occurs, the ORER flag in the SSR register is set to 1. If the RIE bit in the SCR register is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to the RDR register.
4. When reception finishes successfully, receive data is transferred to the RDR register. If the RIE bit in the SCR register is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register in this RXI interrupt handling routine before reception of the next receive data is completed. Reading out the received data that have been transferred to the RDR register causes the RTSn# pin to output the low level (when the RTS function is in use).



Note 1. Refer to the Interrupt Controller section for the corresponding interrupt vector number.

Figure 40.28 Example of Operation for Serial Reception in Clock Synchronous Mode (1) (When RTS Function is Not Used)

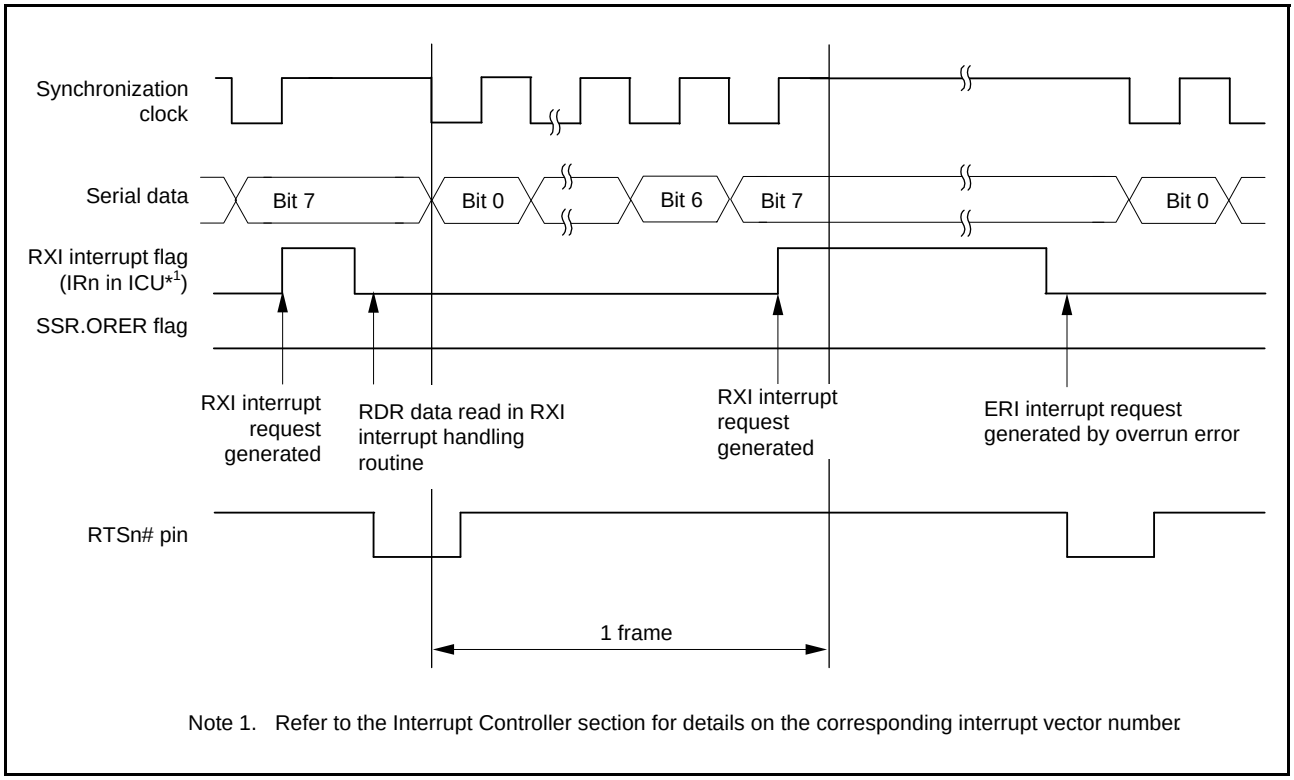
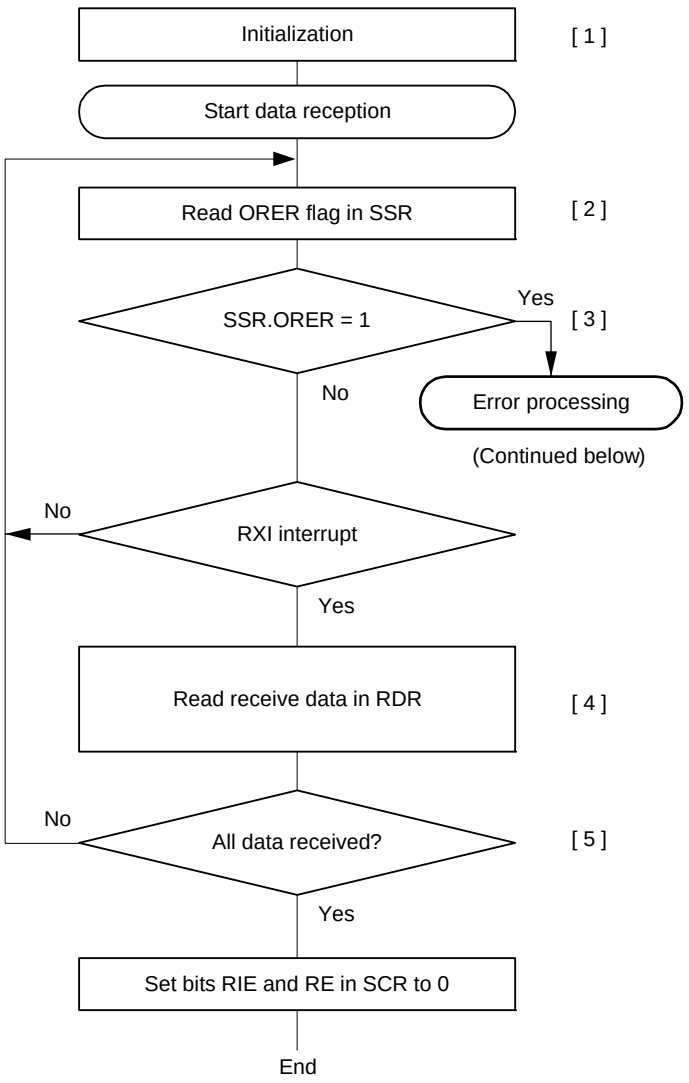
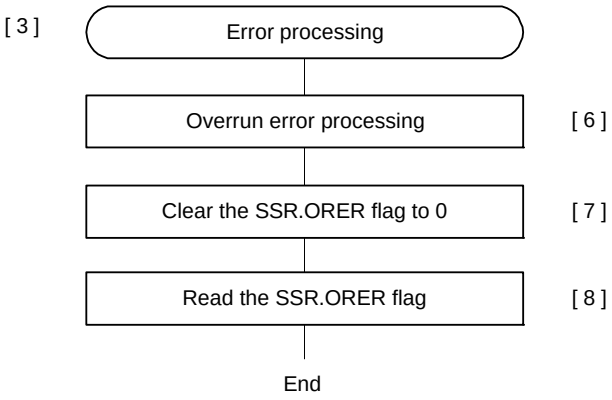


Figure 40.29 Example of Operation for Serial Reception in Clock Synchronous Mode (2) (When RTS Function is Used)

Data transfer cannot be resumed while a receive error flag is 1. Accordingly, clear the ORER, FER, and PER flags in the SSR register to 0 before resuming reception. Moreover, be sure to read the RDR register during overrun error processing. When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in the RDR register.

Figure 40.30 shows a sample flowchart for serial data reception.





[1] SCI initialization:
Make input port-pin settings for pins to be used as RXDn pins.

[2] [3] Receive error processing:
If a receive error occurs, read the ORER flag in the SSR register, perform the relevant error processing, and then set the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] Read the receive data in the RDR register once in the receive data full interrupt (RXI) request handling routine.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, finish reading the receive data in the RDR register. The RDR data can also be read by activating the DMAC or DTC by an RXI interrupt request.

[6] Processing in response to an overrun error:
Read the RDR register. In combination with step [7], this will make correct reception of the next frame possible.

[7] Clearing the error flag:
Write 0 to the error flag.

[8] Confirming that the error flag is cleared:
Read the error flag to confirm that its value is 0.

Figure 40.30 Example Flowchart of Serial Reception in Clock Synchronous Mode

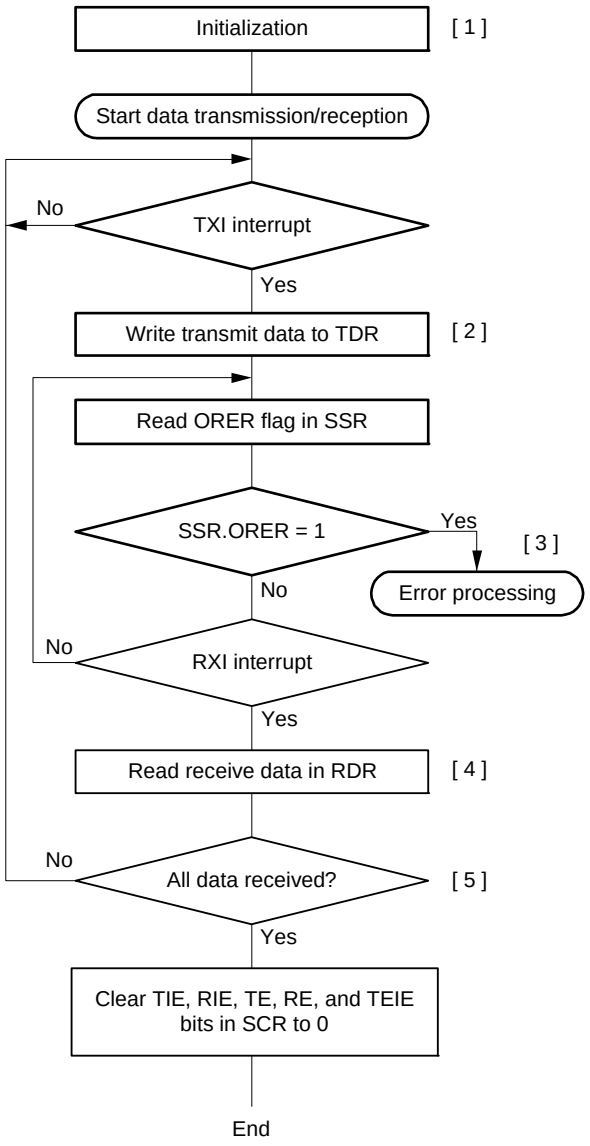
40.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

Figure 40.31 shows a sample flowchart for simultaneous serial transmit and receive operations in clock synchronous mode.

After initializing the SCI, the following procedure should be used for simultaneous serial data transmit and receive operations.

To switch from transmit mode to simultaneous transmit and receive mode, check that the SCI has finished transmission by reading that the TEND flag in the SSR register is 1, and then initialize the SCR register. Then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode, check that the SCI has finished reception, and then set the RIE and RE bits to 0. Then check that the receive error flags (ORER, FER, and PER in the SSR register) are 0, and then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.



[1] SCI initialization:
The TXDn pin can act as the output pin for transmitted data and the RXDn pin can act as the input pin for received data at the same time.

[2] Transmit data write:
Write transmit data to TDR once in the TXI interrupt handling routine.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in the SSR register, perform the relevant error processing, and then set the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] Reading receive data:
Read the receive data in the RDR register once in the RXI interrupt request handling routine.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission and reception, before the MSB (bit 7) of the current frame is received, finish reading the receive data in the RDR register by the RXI interrupt. Also, before the MSB (bit 7) of the current frame is transmitted, write data to the TDR register by the TXI interrupt. Transmit data can also be written to the TDR register by activating the DMAC or DTC by a transmit data empty interrupt (TXI) request. Similarly, the RDR data can also be read by activating the DMAC or DTC by a receive data full interrupt (RXI) request.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations first set the TIE, RIE, TE, RE, and TEIE bits in SCR to 0, and then set TIE, RIE, TE, and RE bits to 1 simultaneously.

Figure 40.31 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode

40.6 Operation in Smart Card Interface Mode

The SCI supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function of the SCI.
Smart card interface mode can be selected using the appropriate register.

40.6.1 Sample Connection

Figure 40.32 shows a sample connection between a smart card (IC card) and this MCU.
As in the figure, since this MCU communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor.
Setting the TE and RE bits in the SCR register to 1 with an IC card disconnected enables closed transmission/reception allowing self-diagnosis.
To supply an IC card with the clock pulses generated by the SCI, input the SCKn pin output to the CLK pin of an IC card.
The output port of the this MCU can be used to output a reset signal.

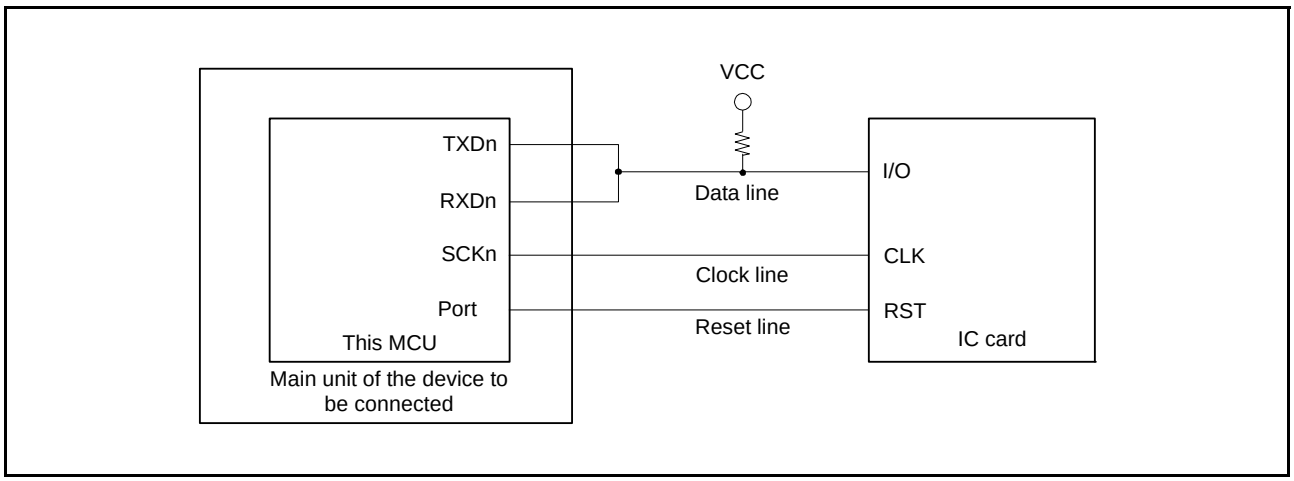


Figure 40.32 Sample Connection with a Smart Card (IC Card)

40.6.2 Data Format (Except in Block Transfer Mode)

Figure 40.33 shows the data transfer formats in smart card interface mode.

- One frame consists of 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etu (elementary time unit: time required for transferring 1 bit) is secured as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low-level error signal is output for 1 etu after 10.5 etu has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 etu.

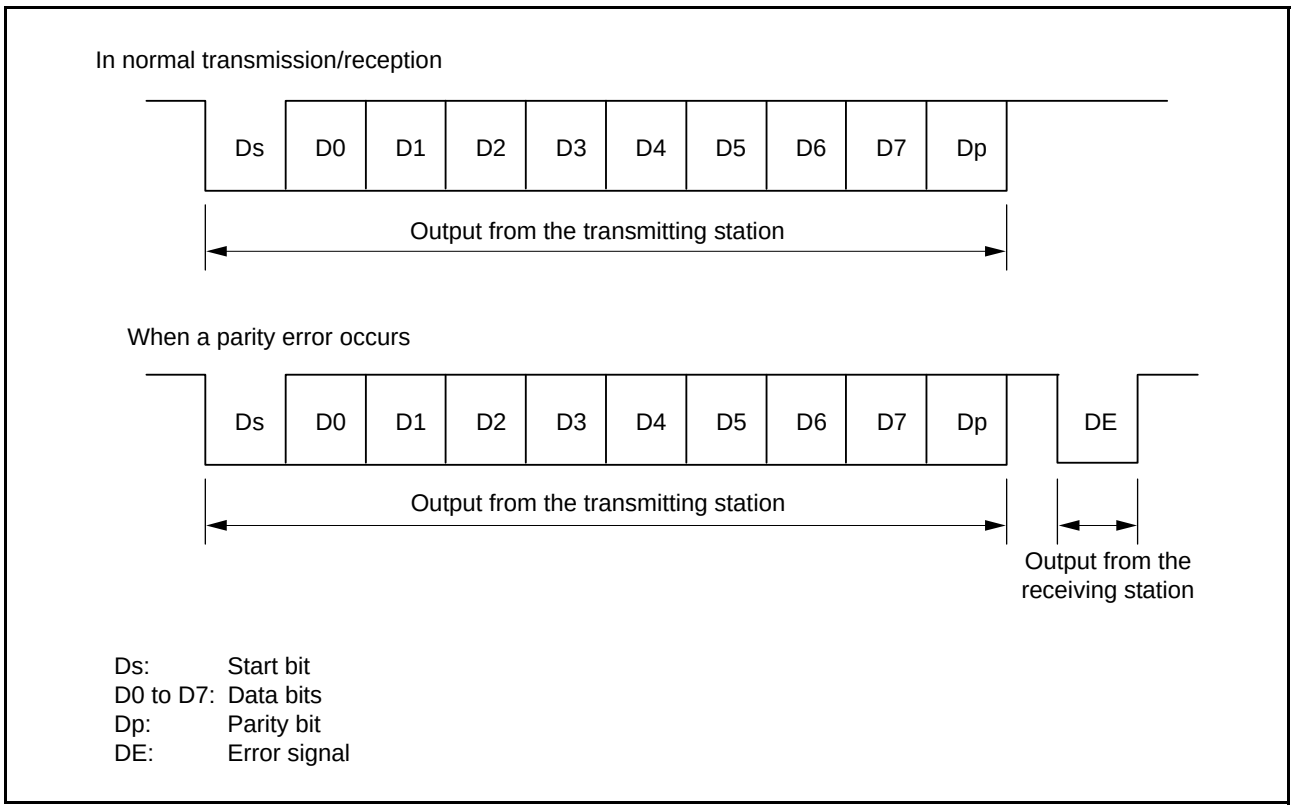


Figure 40.33 Data Formats in Smart Card Interface Mode

For communications with IC cards of the direct convention type and inverse convention type, follow the procedure below.

(1) Direct Convention Type

For the direct convention type, logic levels 1 and 0 correspond to states Z and A, respectively, and data is transferred with LSB first as the start character, as shown in Figure 40.34. Therefore, data in the start character in the figure is 3Bh. When using the direct convention type, write 0 to both the SDIR and SINV bits in the SCMR register. Write 0 to the PM bit in the SMR register in order to use even parity, which is prescribed by the smart card standard.

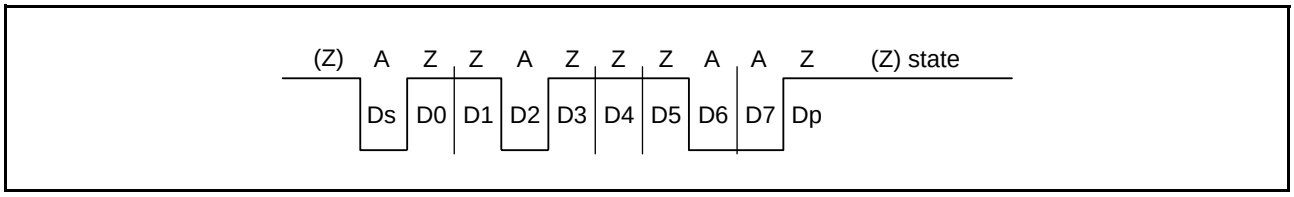


Figure 40.34 Direct Convention (SDIR in SCMR = 0, SINV in SCMR = 0, PM in SMR = 0)

(2) Inverse Convention Type

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respectively and data is transferred with MSB first as the start character, as shown in Figure 40.35. Therefore, data in the start character in the figure is 3Fh. When using the inverse convention type, write 1 to both the SDIR and SINV bits in the SCMR register. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to state Z. Since the SINV bit of the this MCU only inverts data bits D7 to D0, write 1 to the PM bit in the SMR register to invert the parity bit for both transmission and reception.

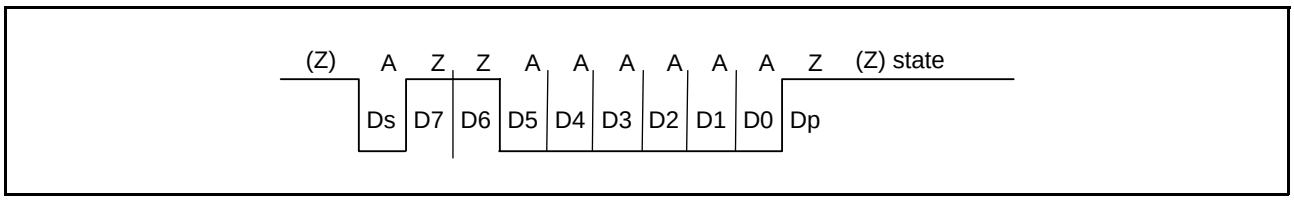


Figure 40.35 Inverse Convention (SDIR in SCMR = 1, SINV in SCMR = 1, PM in SMR = 1)

40.6.3 Block Transfer Mode

Block transfer mode is different from normal smart card interface mode in the following respects.

- Even if a parity error is detected during reception, no error signal is output. Since the PER bit in the SSR register is set by error detection, clear the PER bit before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time from the end of the parity bit until the start of the next frame.
- Since the same data is not retransmitted during transmission, the TEND flag in the SSR register is set 11.5 etu after transmission start.
- In block transfer mode, the ERS flag in the SSR register indicates the error signal status as in normal smart card interface mode, but the flag is read as 0 because no error signal is transferred.

40.6.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the on-chip baud rate generator can be used as a transfer clock in smart card interface mode.

In this mode, the SCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate according to the settings of the BCP2 bit in the SCMR register and the BCP[1:0] bits in the SMR register (the frequency is always 16 times the bit rate in normal asynchronous mode).

For data reception, the falling edge of the start bit is sampled with the base clock to perform internal synchronization. Receive data is sampled on the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, and 256th rising edges of the base clock so that it can be latched at the middle of each bit as shown in Figure 40.36. The reception margin here is determined by the following formula.

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\ F - \frac{D - 0.5}{N}\ (1 + F) \right| \times 100\ [\%]$$

M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, 256)
D: Duty cycle of clock (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the above formula, the reception margin is determined by the formula below.

$$M = \{0.5 - 1/(2 \times 372)\} \times 100\ [\%] = 49.866\%$$

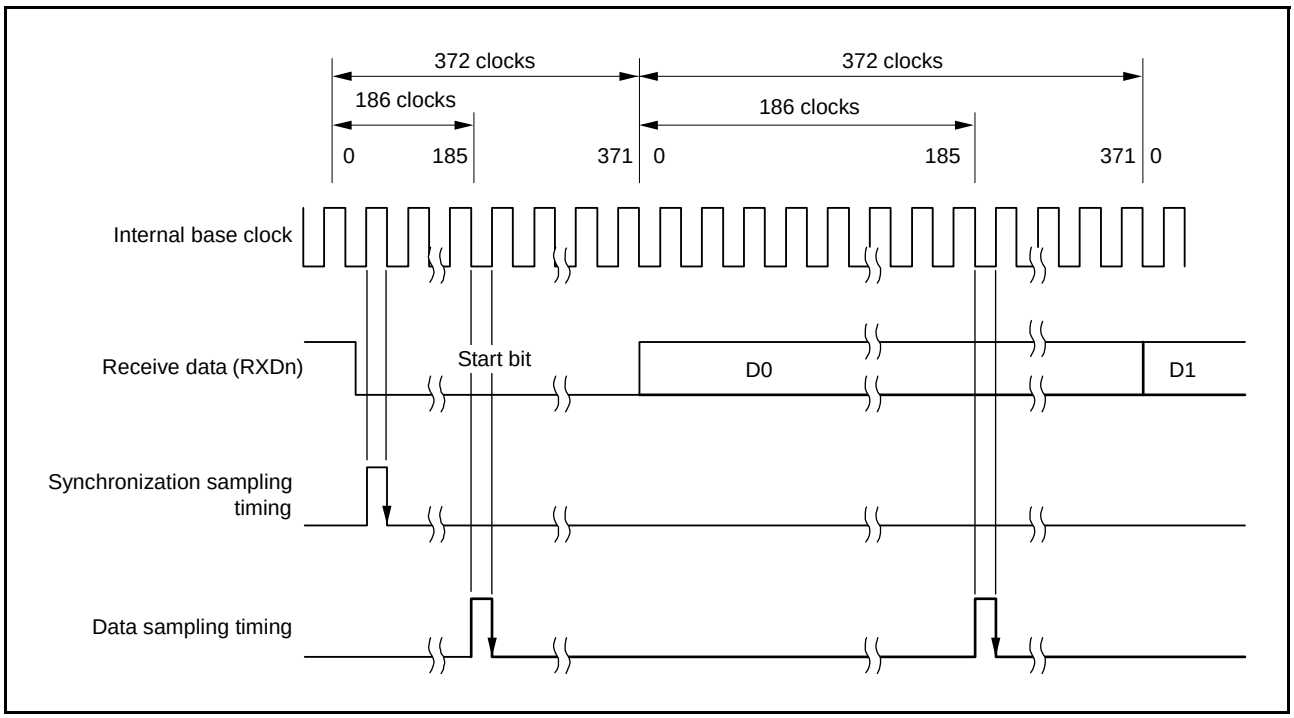


Figure 40.36 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

40.6.5 SCI Initialization (Smart Card Interface Mode)

Initialize the SCI following the example of flowchart shown in Figure 40.37.
Be sure to initialize the SCI before switching from transmission mode to reception mode and vice versa. Even if the RE bit is set to 0, the RDR register is not initialized.

To change reception mode to transmission mode, first check that reception has completed, and then initialize the SCI. At the end of initialization, set TE = 1 and RE = 0. Reception completion can be verified by reading the RXI request, ORER, or PER flag in the SSR register.
To change transmission mode to reception mode, first check that transmission has completed, and then initialize the SCI. At the end of initialization, set TE = 0 and RE = 1. Transmission completion can be verified by reading the TEND flag in the SSR register.

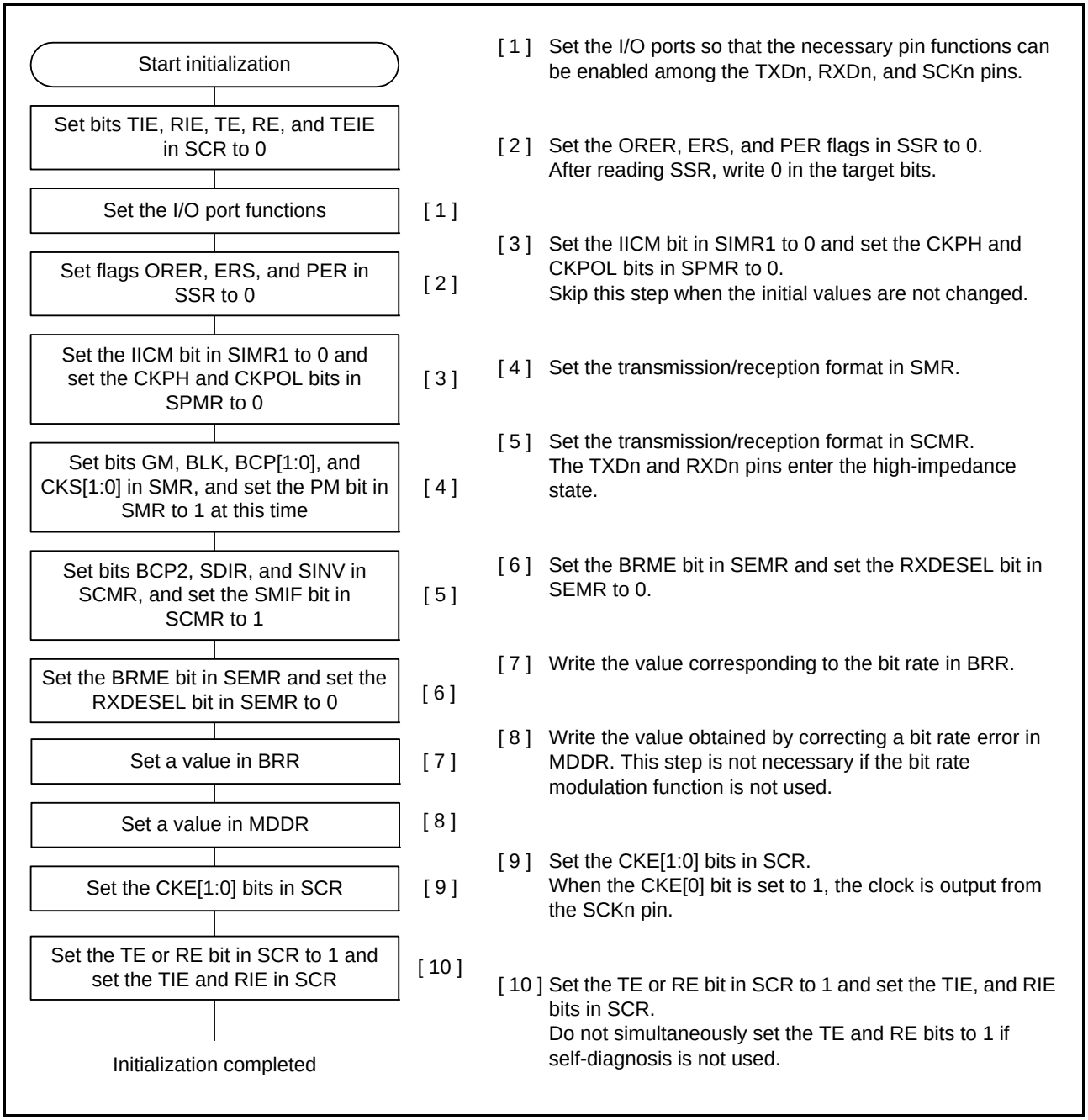


Figure 40.37 Example of SCI Initialization Flowchart (Smart Card Interface Mode)

40.6.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode), in that an error signal is sampled and data can be retransmitted, is different from that in non-smart card interface mode. Figure 40.38 shows the data retransfer operation during transmission.

1. When an error signal from the receiver end is sampled after one-frame data has been transmitted, the ERS flag in the SSR register is set to 1. If the RIE bit in the SCR register is 1 at this time, an ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.
2. For a frame in which an error signal is received, the TEND flag in the SSR register is not set. Data is retransferred from the TDR register to the TSR register allowing automatic data retransmission.
3. If no error signal is returned from the receiver, the ERS flag is not set to 1.
4. In this case, the SCI judges that transmission of one-frame data (including retransfer) has been completed, and the TEND flag is set. If the TIE bit in the SCR register is 1 at this time, a TXI interrupt request is generated. Writing transmit data to the TDR register starts transmission of the next data.

Figure 40.40 shows a sample flowchart of serial transmission. All the processing steps are automatically performed using a TXI interrupt request to activate the DTC or DMAC.
When the TEND flag in the SSR register is set to 1 in transmission, if the TIE bit in the SCR register is 1, a TXI interrupt request is generated. The DTC or DMAC is activated by a TXI interrupt request if the TXI interrupt request is specified as a source of DTC or DMAC activation beforehand, allowing transfer of transmit data. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.
If an error occurs, the SCI automatically retransmits the same data. During this retransmission, the TEND flag is kept to 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, since the ERS flag is not automatically cleared, set the RIE bit to 1 beforehand to enable an ERI interrupt request to be generated at error occurrence, and clear the ERS flag to 0.
When transmitting/receiving data using the DTC or DMAC, be sure to make settings to enable the DTC or DMAC before making SCI settings.
For DTC or DMAC settings, refer to section 18, DMA Controller (DMACAa), section 20, Data Transfer Controller (DTCa).

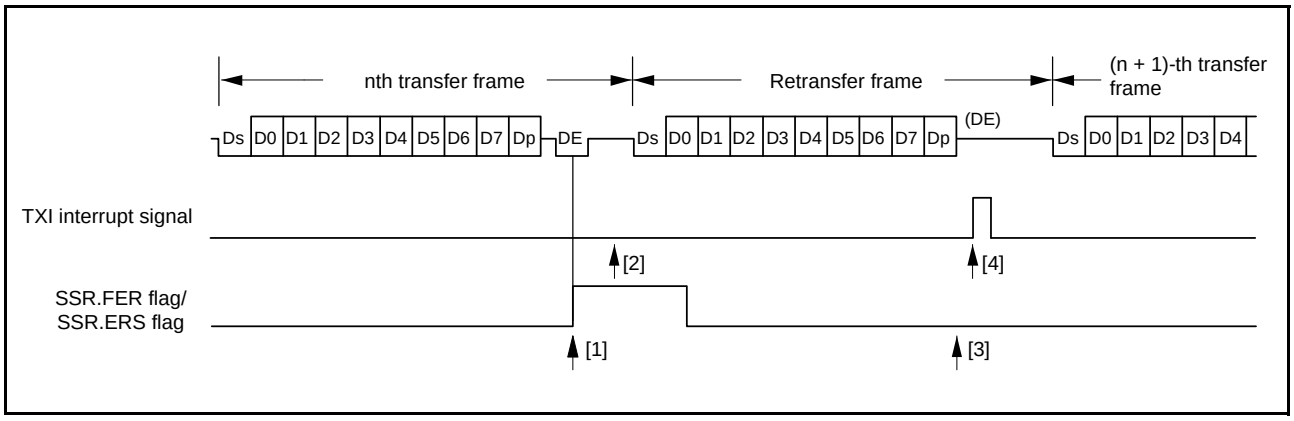


Figure 40.38 Data Retransfer Operation in SCI Transmission Mode

Note that the SSR.TEND flag is set in different timings depending on the GM bit setting in the SMR register. Figure 40.39 shows the TEND flag generation timing.

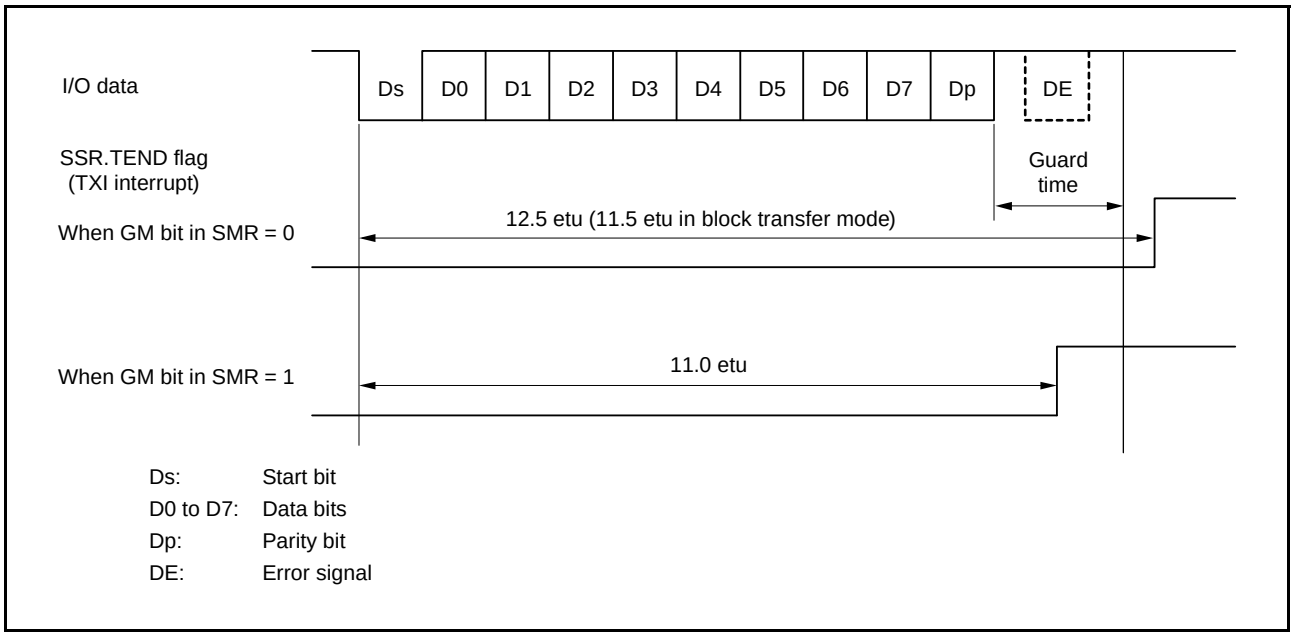


Figure 40.39 SSR.TEND Flag Generation Timing during Transmission

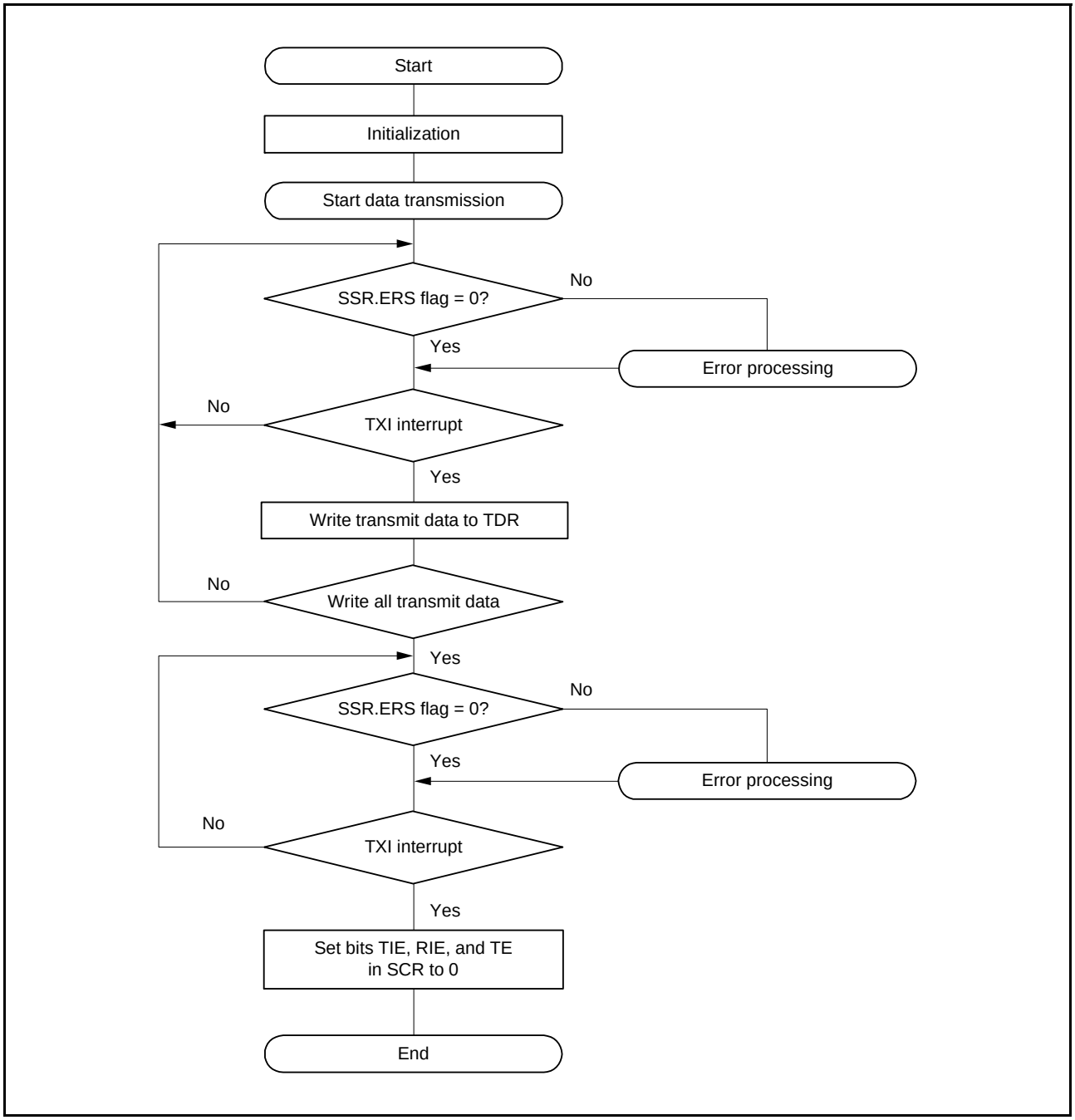


Figure 40.40 Sample Smart Card Interface Transmission Flowchart

40.6.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in non-smart card interface mode. Figure 40.41 shows the data retransfer operation in reception mode.

1. If a parity error is detected in receive data, the PER flag in the SSR register is set to 1. When the RIE bit in the SCR register is 1 at this time, an ERI interrupt request is generated. Clear the PER flag to 0 before the next parity bit is sampled.
2. For a frame in which a parity error is detected, no RXI interrupt is generated.
3. When no parity error is detected, the PER flag in the SSR register is not set to 1.
4. In this case, data is determined to have been received successfully. When the RIE bit in the SCR register is 1, an RXI interrupt request is generated.

Figure 40.42 shows a sample flowchart for serial data reception. All the processing steps are automatically performed using an RXI interrupt request to activate the DTC or DMAC.
In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated. The DTC or DMAC is activated by an RXI interrupt request if the RXI interrupt request is specified as a source of DTC or DMAC activation beforehand, allowing transfer of receive data.
If an error occurs during reception and either the ORER or PER flag in the SSR register is set to 1, a receive error interrupt (ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DTC or DMAC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC or DMAC is transferred.
Even if a parity error occurs and the PER flag is set to 1 during reception, receive data is transferred to RDR, thus allowing the data to be read.
When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR register because the received data which has not yet been read may be left in RDR.

Note 1. For operations in block transfer mode, refer to section 40.3, Operation in Asynchronous Mode.

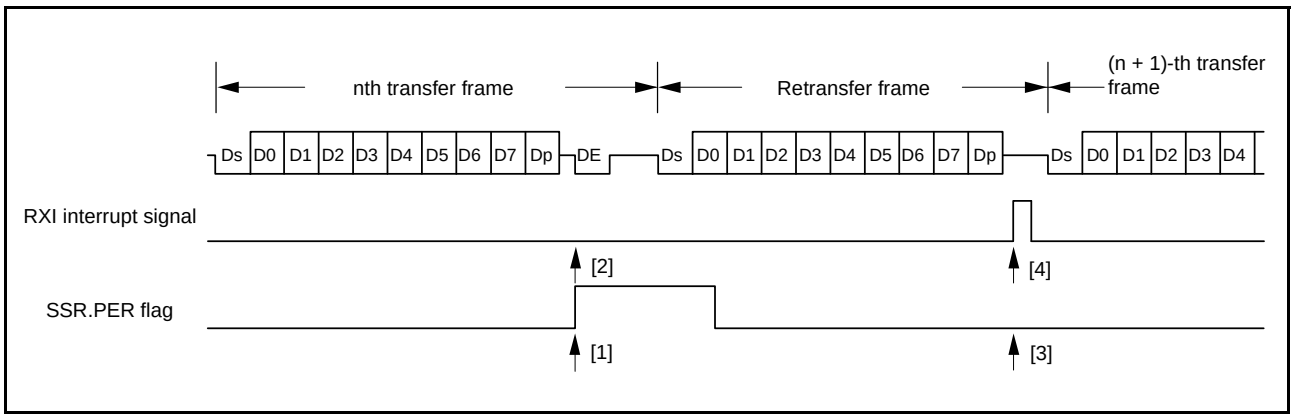


Figure 40.41 Data Retransfer Operation in SCI Reception Mode (Data Retransfer Operation during Reception)

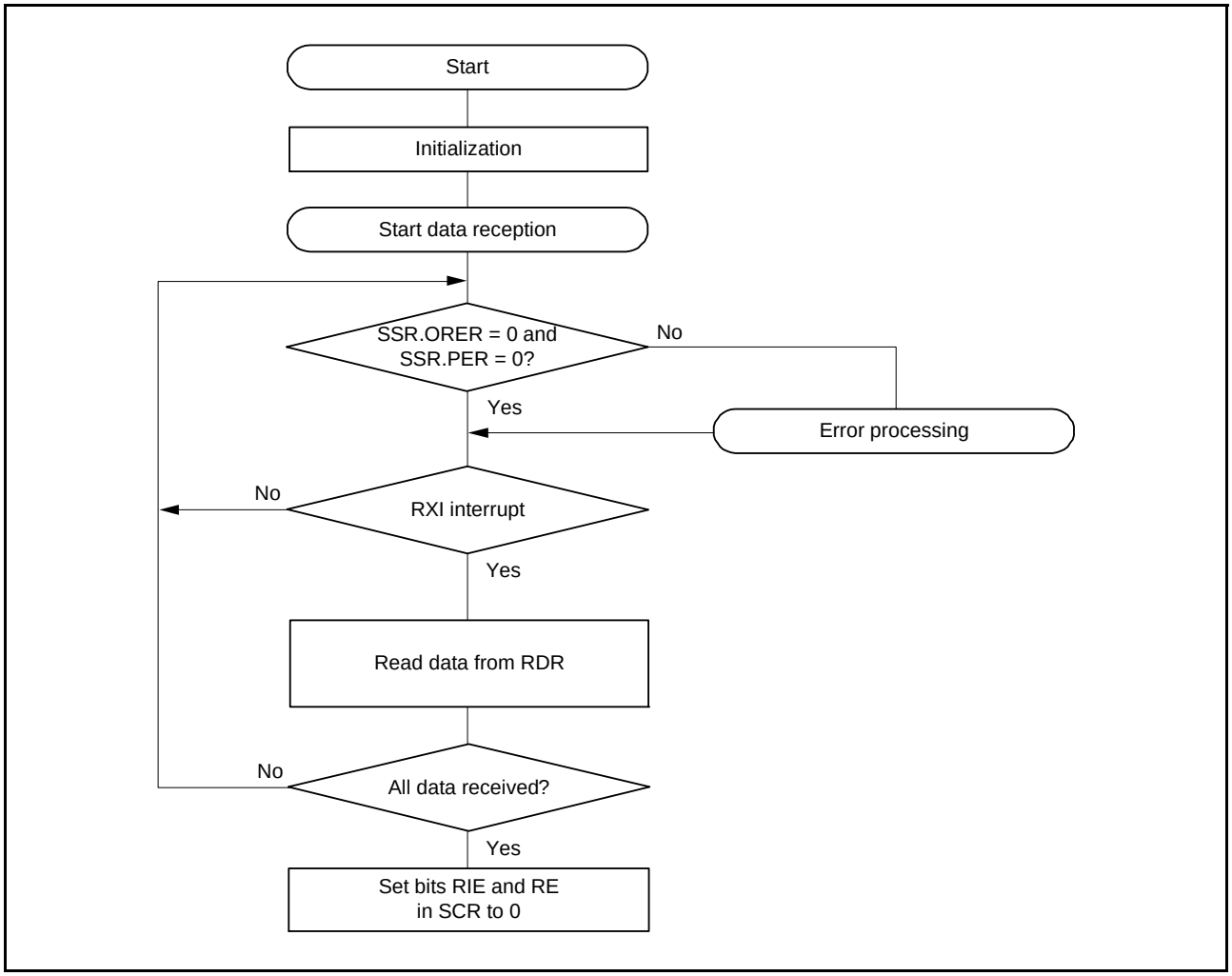


Figure 40.42 Sample Smart Card Interface Reception Flowchart

40.6.8 Clock Output Control

Clock output can be fixed using the CKE[1:0] bits in the SCR register when the GM bit in the SMR register is 1. Specifically, the minimum width of a clock pulse can be specified.

Figure 40.43 shows an example of clock output fixing timing when the CKE[0] bit is controlled with GM = 1 and CKE[1] = 0.

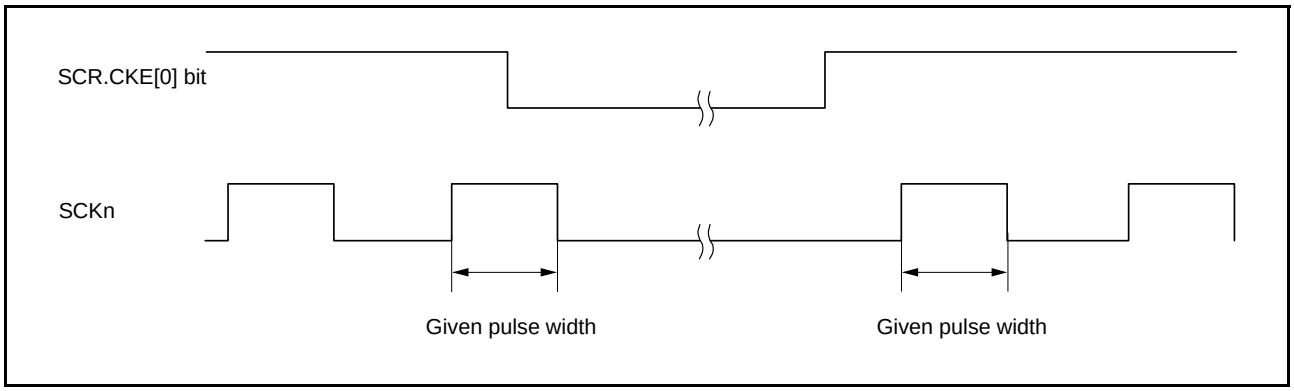


Figure 40.43 Clock Output Fixing Timing

To secure the appropriate clock duty cycle simultaneously with power-on, use the following procedure.

1. Initially, port input is enabled in the high-impedance state. To fix the potential level, use a pull-up or pull-down resistor.
2. Fix the SCKn pin to the specified output by setting the SCR.CKE[1] bit and I/O port functions.
3. Set SMR and SCMR to enable smart card interface mode.
4. Set the SCR.CKE[0] bit to 1 to start clock output.

40.7 Operation in Simple I²C Mode

Simple I²C-bus format is composed of 8 data bits and an acknowledge bit. By continuing into a slave-address frame after a start condition or restart condition, a master device is able to specify a slave device as the partner for communications. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied. The 8 data bits in all frames are transmitted in order from the MSB.

The I²C format and timing of the I²C-bus are shown in Figure 40.44 and Figure 40.45.

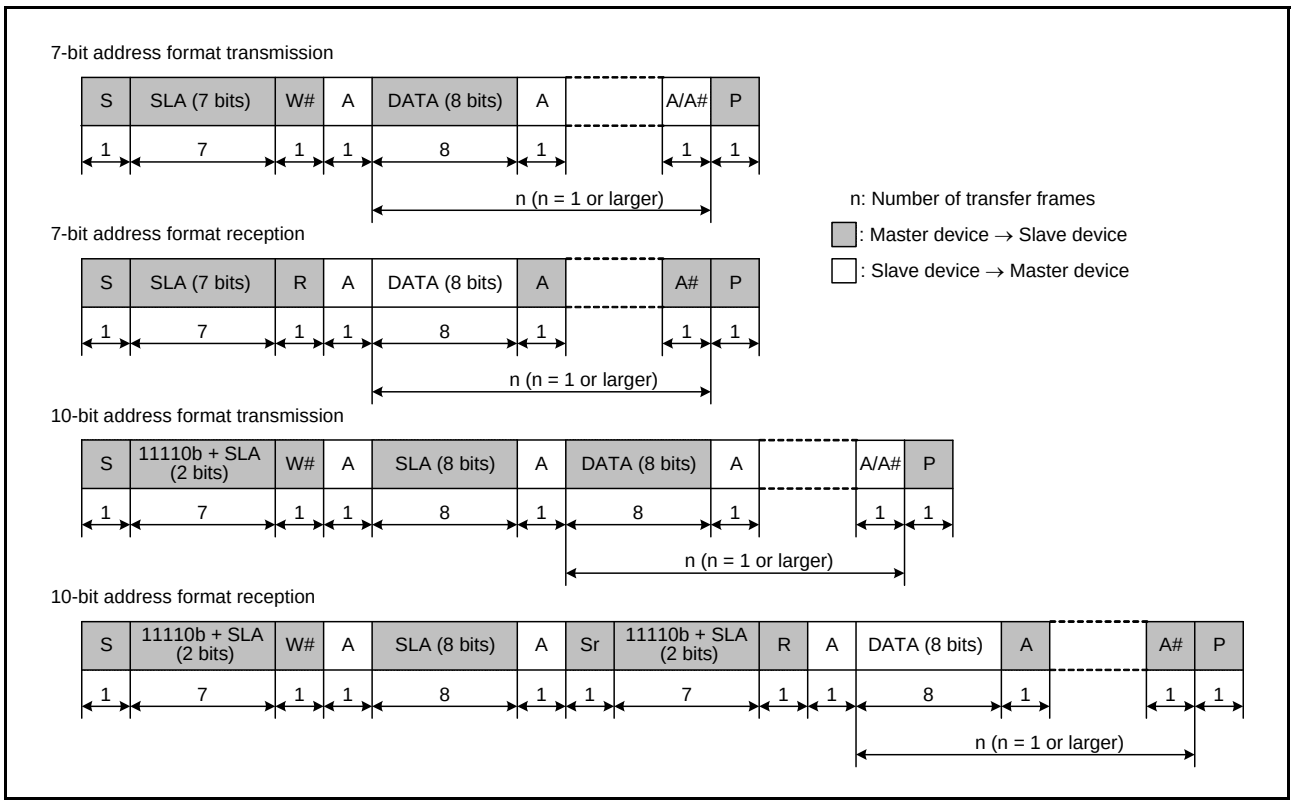


Figure 40.44 I²C-bus Format

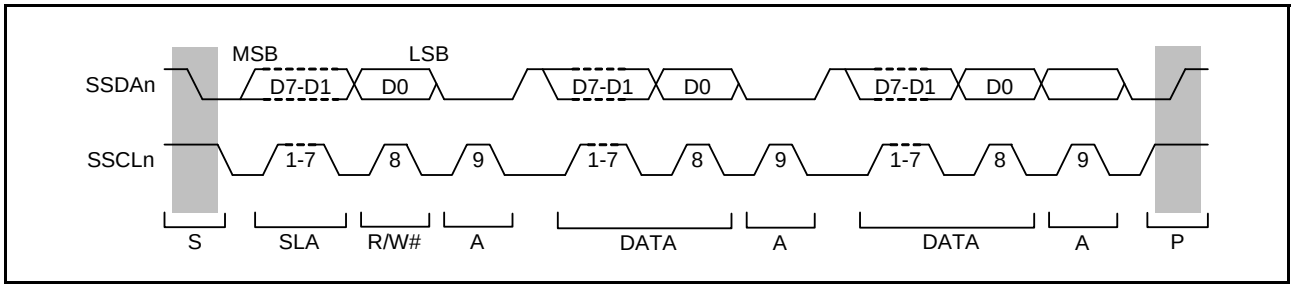


Figure 40.45 I²C-bus Timing (When SLA is 7 Bits)

S: Indicates a start condition, i.e. the master device changing the level on the SSDAn line from the high to the low level while the SSCLn line is at the high level.

SLA: Indicates a slave address, by which the master device selects a slave device.

R/W#: Indicates the direction of transfer (reception or transmission). The value 1 corresponds to transfer from the slave device to the master device and 0 corresponds to transfer from the master device to the slave device.

A/A#: Indicates an acknowledge bit. This is returned by the slave device for master transmission and by the master device for master reception. Return of the low level indicates ACK and return of the high level indicates NACK.

Sr: Indicates a restart condition, i.e. the master device changing the level on the SSDAn line from the high to the low level while the SSCLn line is at the high level and after the setup time has elapsed.

DATA: Indicates the data being received or transmitted.

P: Indicates a stop condition, i.e. the master device changing the level on the SSDAn line from the low to the high level while the SSCLn line is at the high level.

40.7.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to the IICSTAREQ bit in the SIMR3 register causes the generation of a start condition. The generation of a start condition proceeds through the following operations.

- The level on the SSDAn line falls (from the high level to the low level) and the SSCLn line is kept in the released state.
- The hold time for the start condition is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The level on the SSCLn line falls (from the high level to the low level), the IICSTAREQ bit in the SIMR3 register is set (to 0), and a start-condition generated interrupt is output.

Writing 1 to the IICRSTAREQ bit in the SIMR3 register causes the generation of a start condition. The generation of a start condition proceeds through the following operations.

- The SSDAn line is released and the SSCLn line is kept at the low level.
- The period at low level for the SSCLn line is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The SSCLn line is released (transition from the low to the high level).
- Once the high level on the SSCLn line is detected, the setup time for the restart condition is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The level on the SSDAn line falls (from the high level to the low level).
- The hold time for the restart condition is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The level on the SSCLn line falls (from the high level to the low level), the IICRSTAREQ bit in the SIMR3 register is set (to 0), and a restart-condition generated interrupt is output.

Writing 1 to the IICSTPREQ bit in the SIMR3 register causes the generation of a stop condition. The generation of a stop condition proceeds through the following operations.

- The level on the SSDAn line falls (from the high level to the low level) and the SSCLn line is kept at the low level.
- The period at low level for the SSCLn line is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The SSCLn line is released (transition from the low to the high level).
- Once the high level on the SSCLn line is detected, the setup time for the stop condition is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The SSDAn is released (transition from the low to the high level), the IICSTPREQ bit in the SIMR3 register is set (to 0), and a stop-condition generated interrupt is output.

Figure 40.46 shows the timing of operations in the generation of start, restart, and stop conditions.

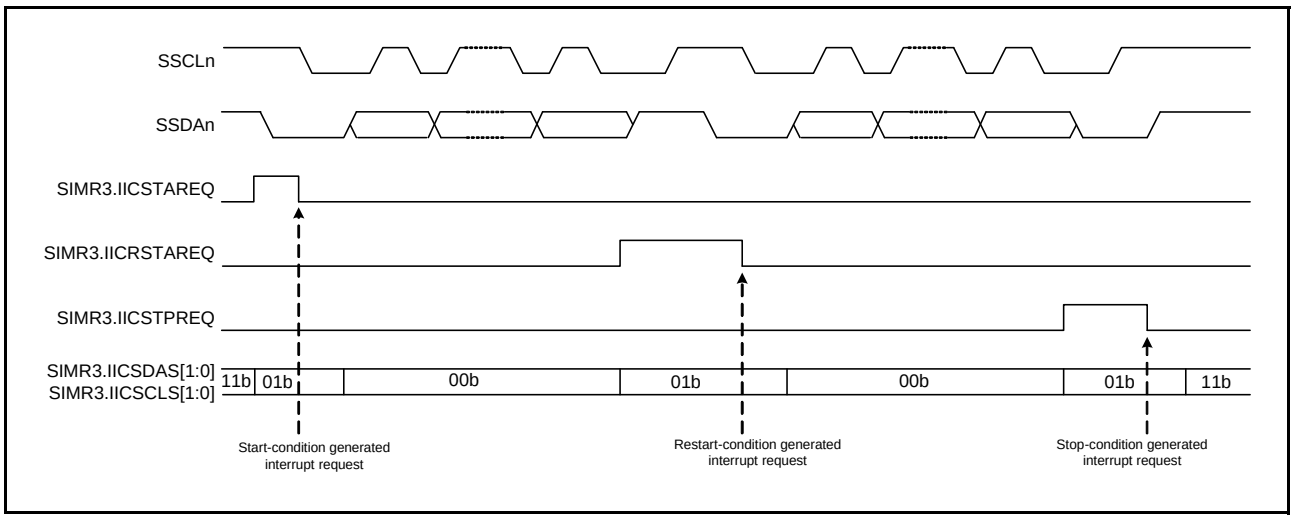


Figure 40.46 Timing of Operations in the Generation of Start, Restart, and Stop Conditions

40.7.2 Clock Synchronization

The SSCLn line may be placed at the low level in the case of a wait inserted by a slave device as the other side of transfer. Setting the IICCSC bit in the SIMR2 register to 1 applies control to obtain synchronization when the levels of the internal SSCLn clock signal and the level being input on the SSCLn pin differ.

When the IICCSC bit in the SIMR2 register is set to 1, the level of the internal SSCLn clock signal changes from low to high, counting to determine the period at high level is stopped while the low level is being input on the SSCLn pin, and counting to determine the period at high level starts after the transition of the input on the SSCLn pin to the high level. The interval from this time until counting to determine the period at high level starts on the transition of the SSCLn pin to the high level is the total of the delay of SSCLn output, delay for noise filtering of the input on the SSCLn pin (2 or 3 cycles of sampling clock for the noise filter), and delay for internal processing (1 or 2 cycles of PCLK). The period at high level of the internal SSCLn clock is extended even if other devices are not placing the low level on the SSCLn line.

If the IICCSC bit in the SIMR2 register is 1, synchronization is obtained for the transmission and reception of data by taking the logical AND of the input on the SSCLn pin and the internal SSCLn clock. If the IICCSC bit in the SIMR2 register is 0, synchronization with the internal SSCLn clock is obtained for the transmission and reception of data.

If a slave device inserts a period of waiting into the interval until the transition of the internal SSCLn clock signal from the low to the high level after a request for the generation of a start, restart, or stop condition is issued, the time until generation is prolonged by that period.

If a slave device inserts a period of waiting after the transition of the internal SSCLn clock signal from the low to the high level, although the generation-completed interrupt is issued without stopping the period of waiting, generation of the condition itself is not guaranteed. Figure 40.47 shows an example of operations to synchronize the clocks.

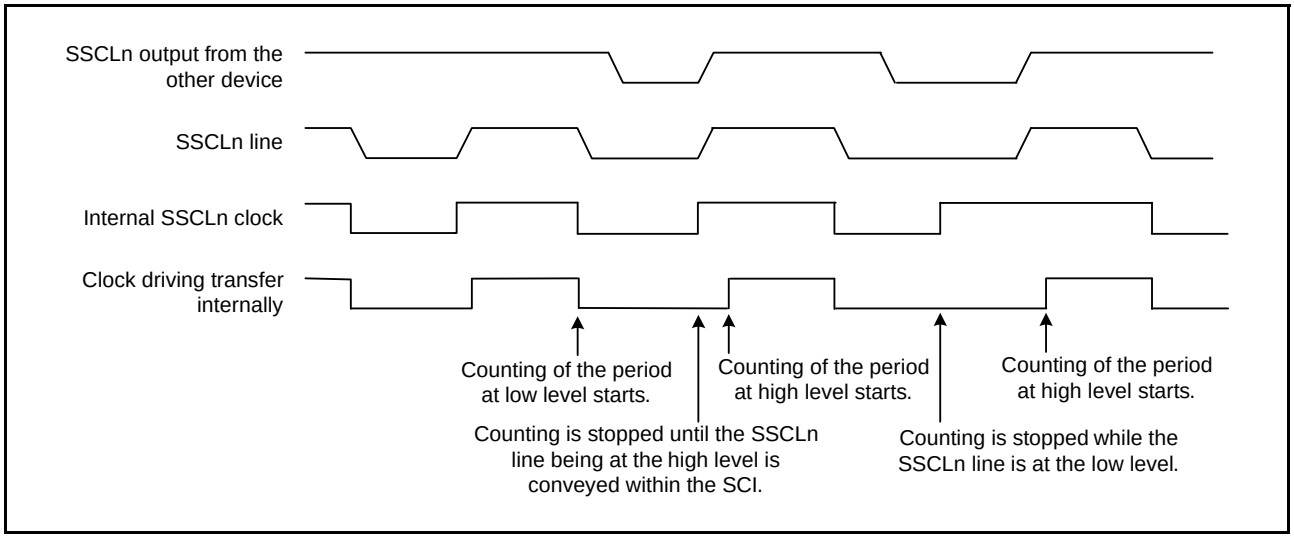


Figure 40.47 Example of Operations for Clock Synchronization

40.7.3 SSDA Output Delay

The IICDL[4:0] bits in the SIMR1 register can be used to set a delay for output on the SSDAn pin relative to falling edges of output on the SSCLn pin. Delay-time settings from 0 to 31 are selectable, representing periods of the corresponding numbers of cycles of the clock signal from the on-chip baud rate generator (derived by frequency-dividing the base clock, PCLK, by the divisor selected by the CKS[1:0] bits in the SMR register). A delay for output on the SSDAn pin is for the start condition/restart condition/stop condition signal, 8-bit transmit data, and an acknowledge bit. If the SSDA output delay is shorter than the time for the level on the SSCLn pin to fall, the change of the output on the SSDAn pin will start while the output level on the SSCLn pin is falling, creating a possibility of erroneous operation for slave devices. Ensure that settings for the delay of output on the SSDAn pin are for times greater than the time output on the SSCLn pin takes to fall (300 ns for I^2C in normal mode and fast mode).
Figure 40.48 shows the timing of delays in SSDA output.

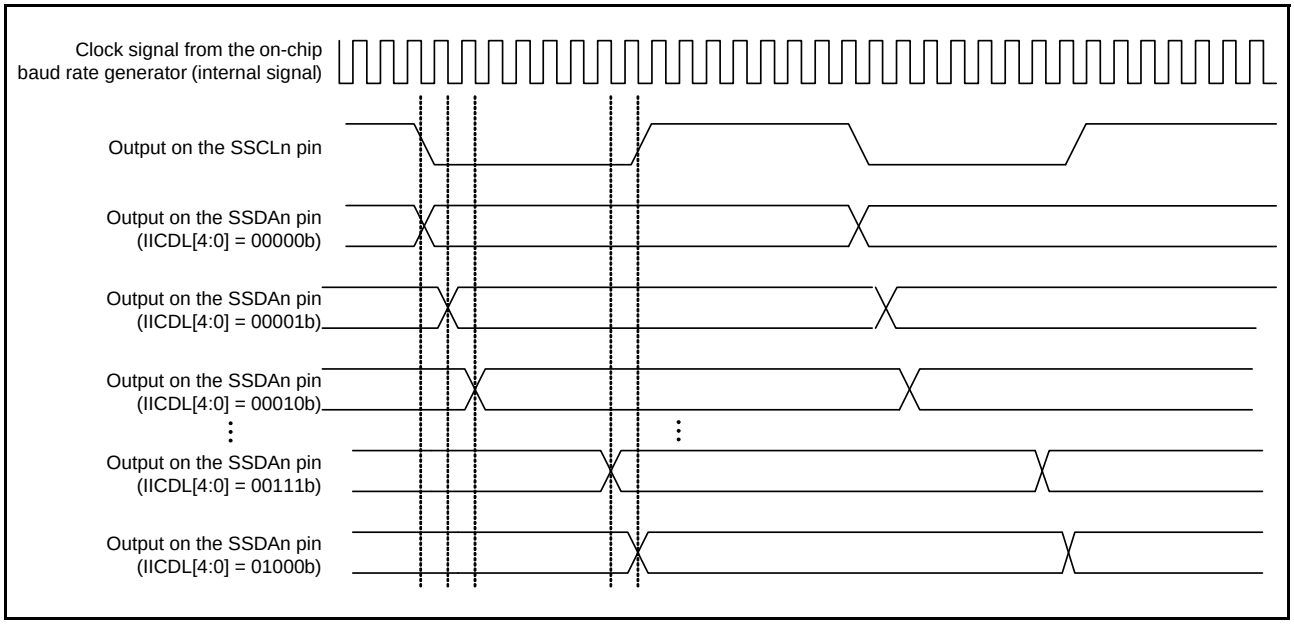


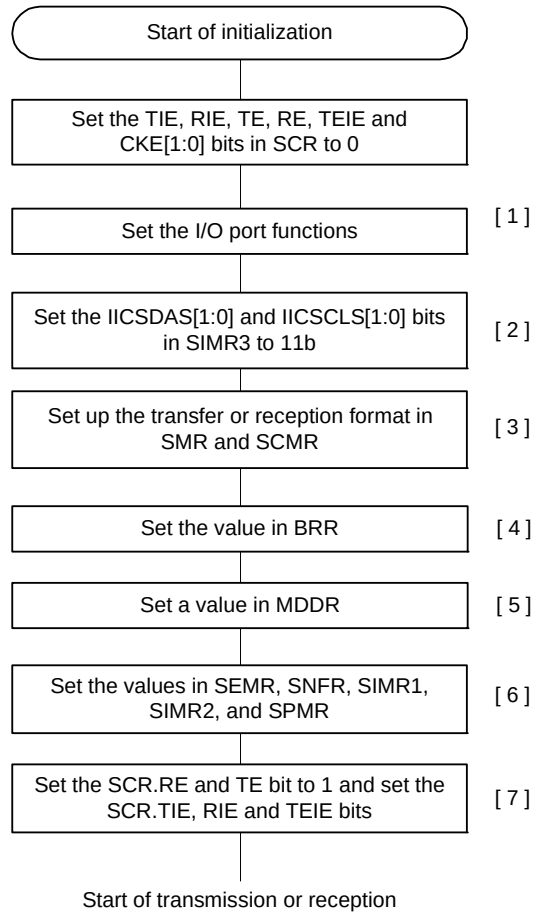
Figure 40.48 Timing of Delays in SSDA Output

40.7.4 SCI Initialization (Simple I2C Mode)

Before transferring data, write the initial value (00h) to SCR and initialize the interface following the example shown in Figure 40.49.

When changing the operating mode, transfer format, and so on, be sure to set SCR to its initial value before proceeding with the changes.

In simple I2C mode, the open-drain setting for the communication ports should be made on the port side.



[1] Make I/O port settings that allow use (on N-channel open-drain output pins) of the SSCLn and SSDAn pin functions.

[2] Place the SSCLn and SSDAn pins in the high-impedance state until a start condition is to be generated.

[3] Set the format for transmission and reception in SMR and SCMR.
In SMR, set the CKS[1:0] bits to the desired value and set the other bits to 0.
In SCMR, set the SDIR bit to 1 and the SINV and SMIF bits to 0.

[4] Write the value for the desired bit rate to BRR.

[5] Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the BRME bit in SEMR is cleared to 0.

[6] Set the values in SEMR, SNFR, SIMR1, SIMR2, and SPMR.
In SEMR, set the NFEN bit. Set the NFEN and BRME bits in SEMR.
In SNFR, set the NFCS[2:0] bits.
In SIMR1, set the IICM bit to 1 and the IICDL[4:0] bits as required.
In SIMR2, set the IICACKT and IICCSC bits to 1 and the IICINTM bits as required.
In SPMR, set all the bits to 0.

[7] Set the RE and TE bits in the SCR to 1. Then, set the SCR.TIE, RIE, and TEIE bits (for transmission and when the SIMR2.IICINTM bit is 1, set the RIE bit to 0). Setting the TE and RE bits to 1 makes the SSCLn and SSDAn pin functions available.

Note: Set the SCR.RE and TE bits to 0 or 1 at the same time.

Figure 40.49 Example of the Flowchart of SCI Initialization (for Simple I2C Mode)

40.7.5 Operation in Master Transmission (Simple I2C Mode)

Figure 40.50 and Figure 40.51 show examples of operations in master transmission and Figure 40.52 is a flowchart showing the procedure for data transmission. Refer to Table 40.33 for more information on the STI interrupt.
When 10-bit slave addresses are in use, steps [3] and [4] in Figure 40.52 are repeated twice.
In simple I2C mode, the transmit data empty interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock synchronous transmission.

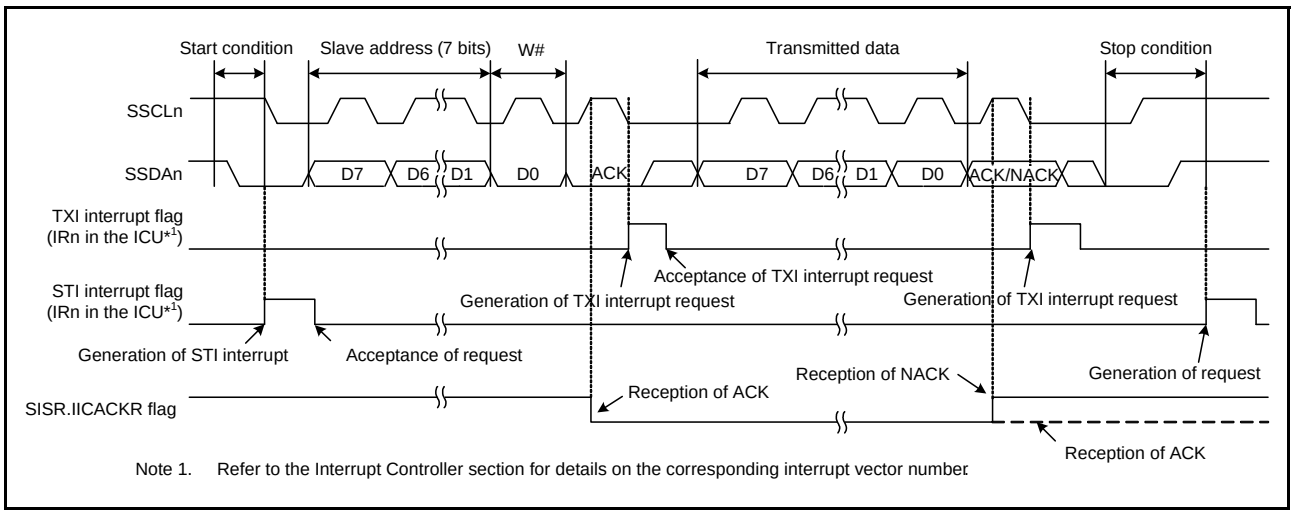


Figure 40.50 Example 1 of Operations for Master Transmission in Simple I2C-bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)

When the SIMR2.IICINTM bit is set to 0 (use ACK/NACK interrupts) during master transmission, the DTC or DMAC is activated by the ACK interrupt as the trigger and necessary number of data bytes are transmitted. When the NACK is received, error processing, such as transmission stop and retransmission, is performed by the NACK interrupt as the trigger.

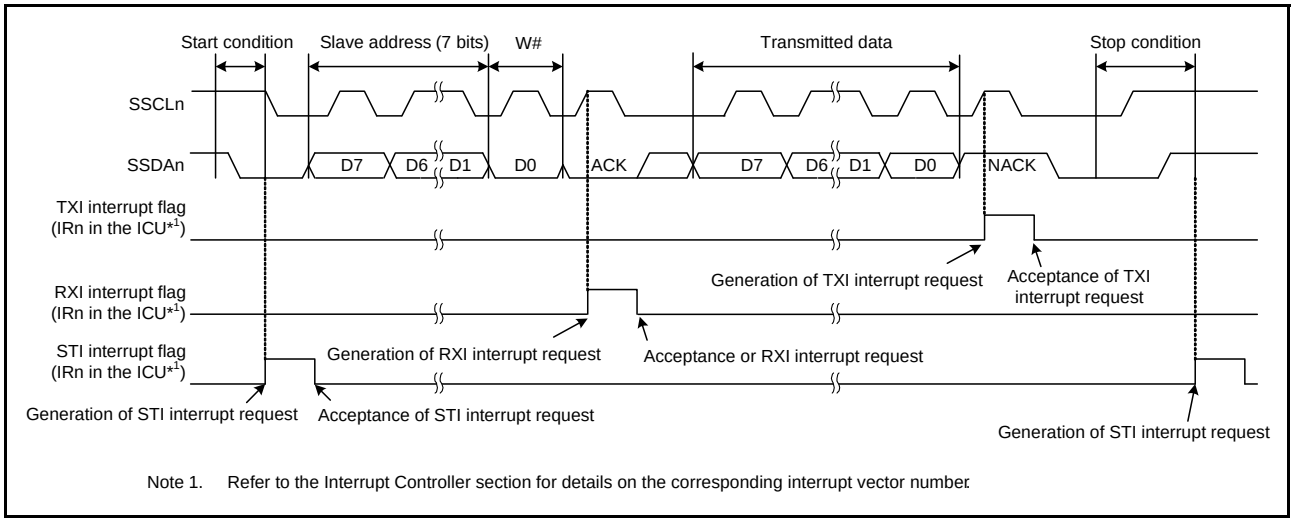
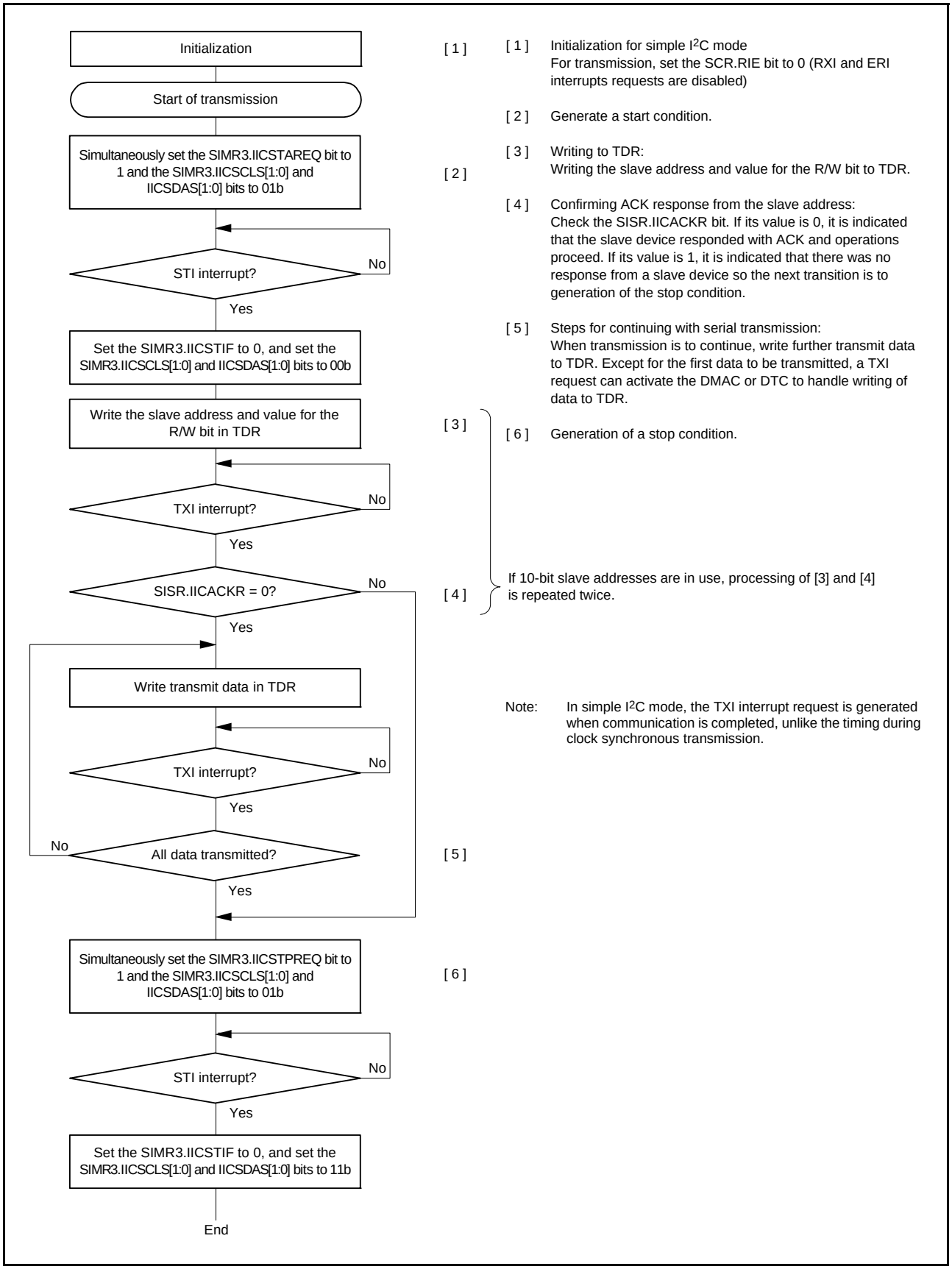


Figure 40.51 Example 2 of Operations for Master Transmission in Simple I2C-bus Mode (with 7-Bit Slave Addresses, ACK Interrupts, and NACK Interrupts in Use)



[1] [1] Initialization for simple I2C mode
For transmission, set the SCR.RIE bit to 0 (RXI and ERI interrupts requests are disabled)

[2] Generate a start condition.

[3] Writing to TDR:
[2] Writing the slave address and value for the R/W bit to TDR.

[4] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

[5] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to TDR. Except for the first data to be transmitted, a TXI request can activate the DMAC or DTC to handle writing of data to TDR.

[3] [6] Generation of a stop condition.

[4] If 10-bit slave addresses are in use, processing of [3] and [4] is repeated twice.

Note: In simple I2C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock synchronous transmission.

Figure 40.52 Example of the Procedure for Master Transmission Operations in Simple I2C Mode (with Transmission Interrupts and Reception Interrupts in Use)

40.7.6 Master Reception (Simple I2C Mode)

Figure 40.53 shows an example of operations in simple I2C mode master reception and Figure 40.54 is a flowchart showing the procedure for master reception.

The value of the SIMR2.IICINTM bit is assumed to be 1 (use reception and transmission interrupts).

In simple I2C mode, the transmit data empty interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock synchronous transmission.

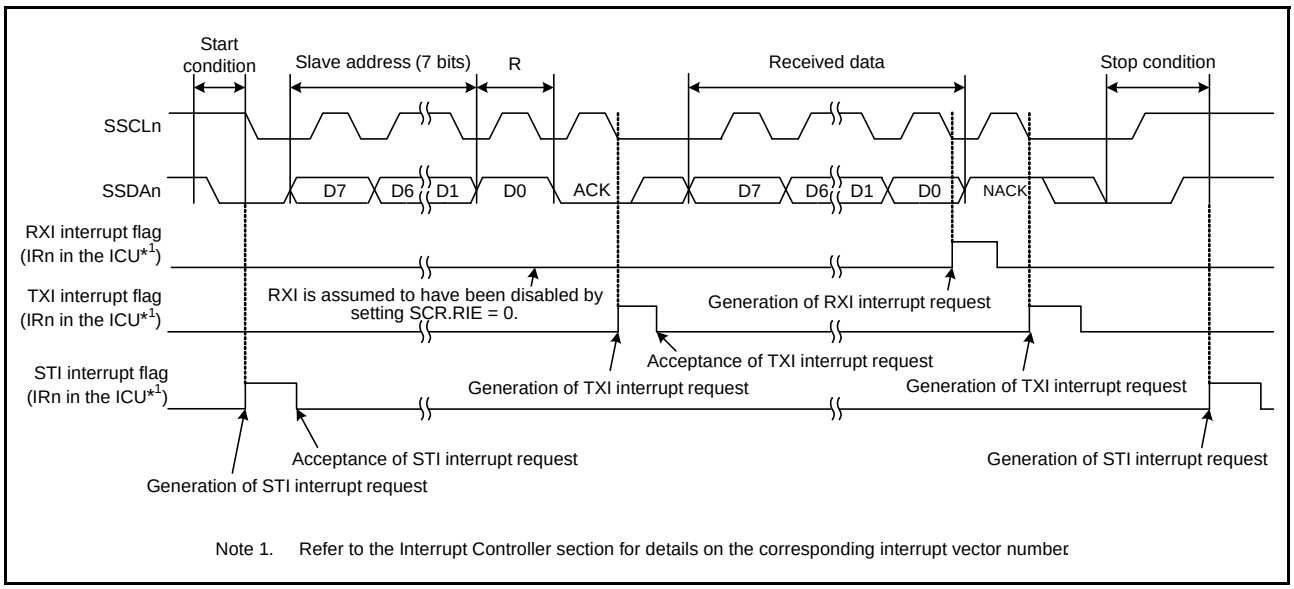
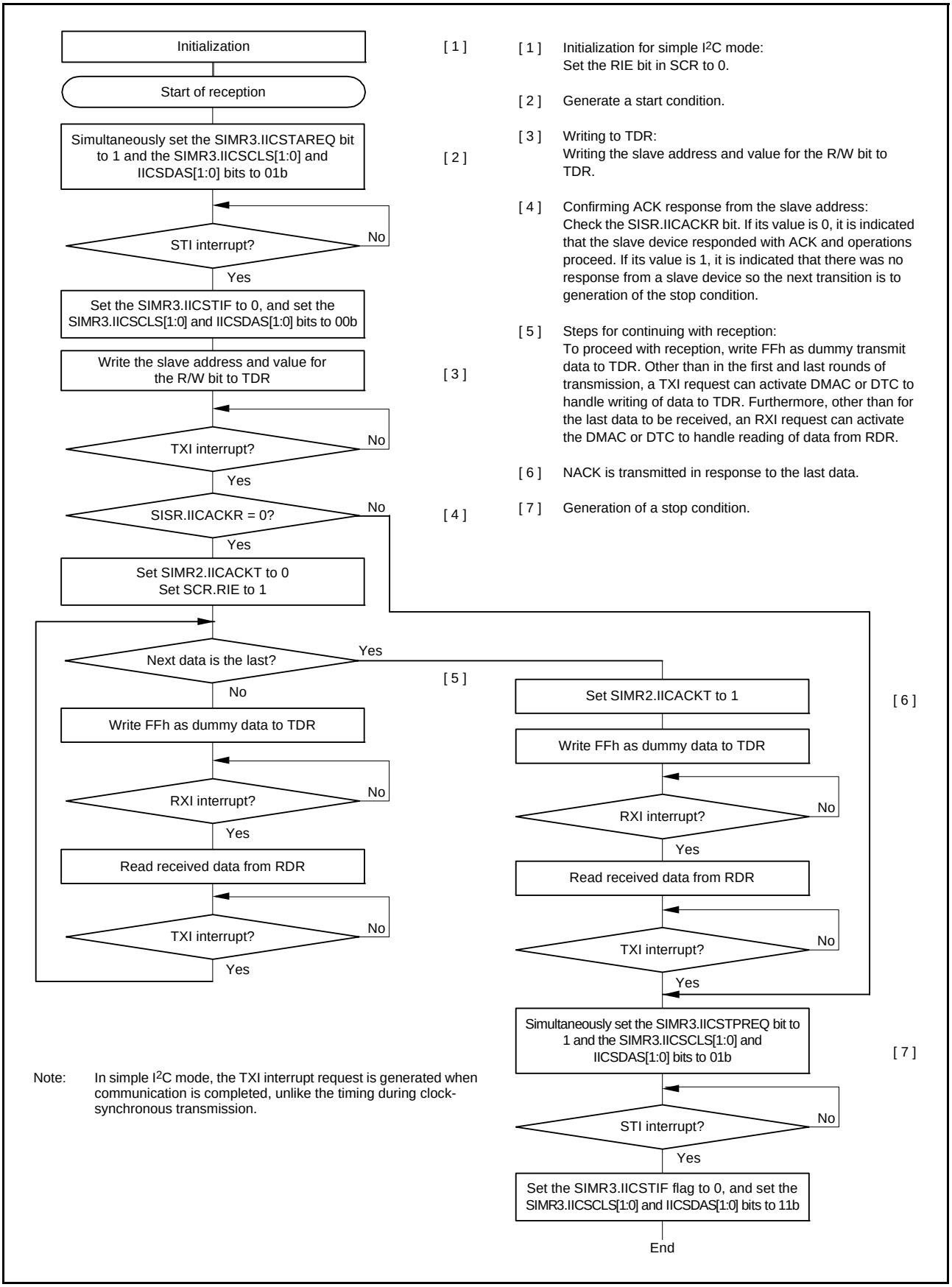


Figure 40.53 Example of Operations for Master Reception in Simple I2C-bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)



[1] Initialization for simple I²C mode:
Set the RIE bit in SCR to 0.

[2] Generate a start condition.

[3] Writing to TDR:
Writing the slave address and value for the R/W bit to TDR.

[4] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

[5] Steps for continuing with reception:
To proceed with reception, write FFh as dummy transmit data to TDR. Other than in the first and last rounds of transmission, a TXI request can activate DMAC or DTC to handle writing of data to TDR. Furthermore, other than for the last data to be received, an RXI request can activate the DMAC or DTC to handle reading of data from RDR.

[6] NACK is transmitted in response to the last data.

[7] Generation of a stop condition.

Note: In simple I²C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock-synchronous transmission.

Figure 40.54 Example of the Procedure for Master Reception Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

40.8 Operation in Simple SPI Mode

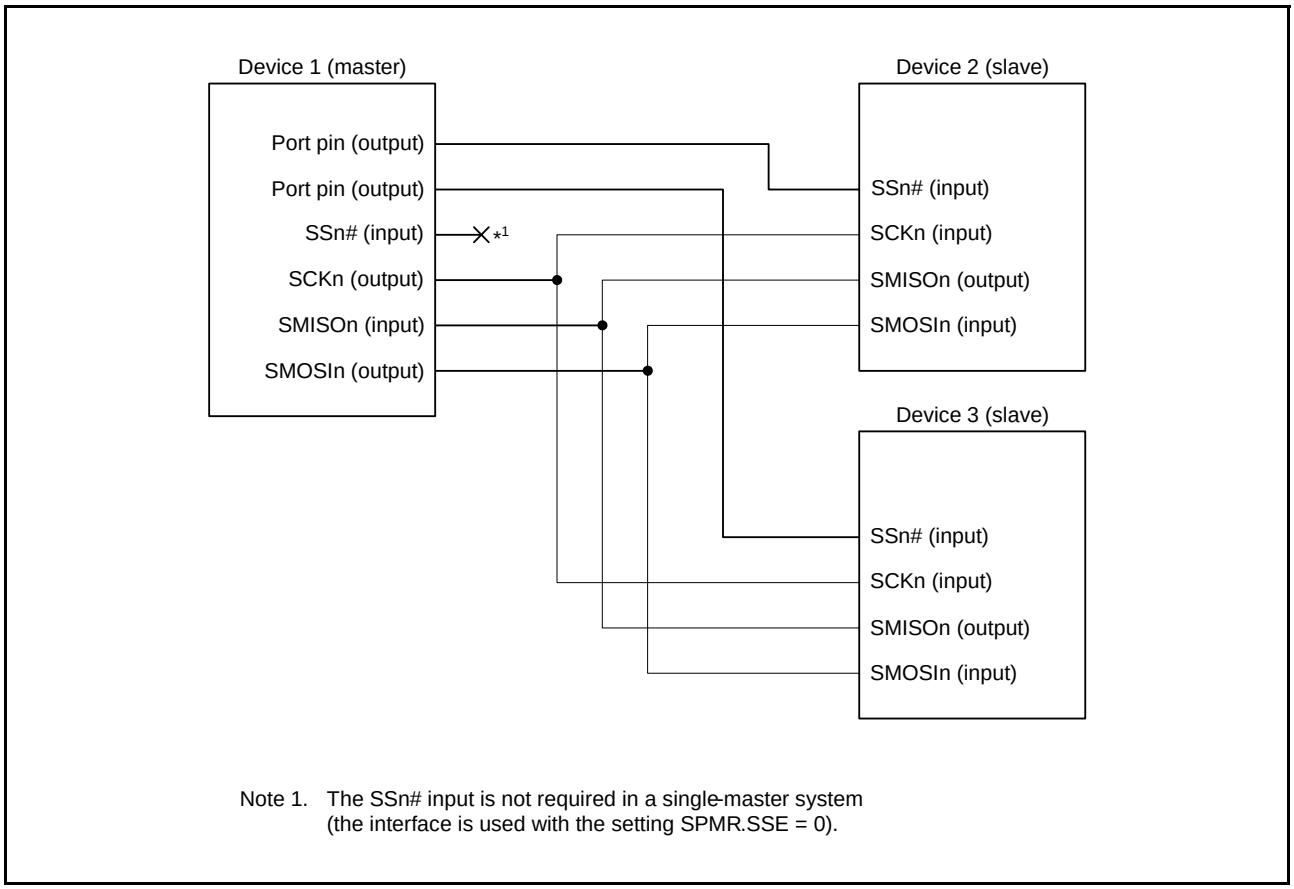
As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Making the settings for clock synchronous mode (SCMR.SMIF = 0, SIMR1.IICM = 0, SMR.CM = 1) plus setting the SSE bit in the SPMR to 1 places the SCI in simple SPI mode. However, the SS pin function on the master side is unnecessary for connection of the device used as the master in simple SPI mode when the configuration only has a single master, so set the SSE bit in the SPMR to 0 in such cases.

Figure 40.55 shows an example of connections for simple SPI mode. Control a general port pin to produce the SS output signal from the master.

In simple SPI mode, data are transferred in synchronization with clock pulses in the same way as in clock synchronous mode. One character of data for transfer consists of 8 bits of data, and parity bits cannot be appended to this. The data can be inverted by setting the SCMR.SINV bit to 1.

Since the receiver and transmitter are independent of each other within the SCI module, full-duplex communications are possible, with a common clock signal. Furthermore, since both the transmitter and receiver have a double-buffered structure, writing of further transmit data while transmission is in progress and reading of previously received data while reception is in progress are both possible. Continuous transfer is thus possible.



Note 1. The SSn# input is not required in a single-master system (the interface is used with the setting SPMR.SSE = 0).

Figure 40.55 Example of Connections via a Simple SPI Mode (In Single Master Mode, SPMR.SSE Bit = 0)

40.8.1 States of Pins in Master and Slave Modes

The direction (input or output) of pins for the simple SPI mode interface differs according to whether the device is a master (SCR.CKE[1:0] = 00b or 01b and SPMR.MSS = 0) or slave (SCR.CKE[1:0] = 10b or 11b and SPMR.MSS = 1). Table 40.29 lists the states of pins according to the mode and the level on the SSn# pin.

Table 40.29 States of Pins by Mode and Input Level on the SSn# Pin

Mode	Input on SSn# Pin	State of SMOSIn Pin	State of SMISOn Pin	State of SCKn Pin
Master mode*1	High level (transfer can proceed)	Output for data transmission*2	Input for received data	Clock output*3
	Low level (transfer cannot proceed)	High-impedance	Input for received data (but disabled)	High-impedance
Slave mode	High level (transfer can proceed)	Input for received data (but disabled)	High-impedance	Clock input (but disabled)
	Low level (transfer cannot proceed)	Input for received data	Output for data transmission	Clock input

Note 1. When there is only a single master (SPMR.SSE = 0), transfer is possible regardless of the input level on the SSn# pin (this is equivalent to input of a high level on the SSn# pin). Since the SSn# pin function is not required, the pin is available for other purposes.
Note 2. The SMOSIn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE bit = 0).
Note 3. The SCKn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE and RE bits = 00b) in a multi-master configuration (SPMR.SSE = 1).

40.8.2 SS Function in Master Mode

Setting the CKE[1:0] bits in the SCR to 00b and the MSS bit in the SPMR to 0 selects master operation. The SSn# pin is not used in single-master configurations (SPMR.SSE = 0), so transmission or reception can proceed regardless of the value of the SSn# pin.

When the level on the SSn# pin is high in a multi-master configuration (SPMR.SSE = 1), a master device outputs clock signals from the SCKn pin before starting transmission or reception to indicate that there are no other masters or another master is performing reception or transmission. When the level on the SSn# pin is low in a multi-master configuration (SPMR.SSE = 1), there are other masters, and this indicates that transmission or reception is in progress. At this time the SMOSIn output and SCKn pins will be placed in the high-impedance state and starting transmission or reception will not be possible. Furthermore, the value of the SPMR.MFF bit will be 1, indicating a mode fault error. In a multi-master configuration, start error processing by reading SPMR.MFF flag. Also, even if a mode fault error occurs while transmission or reception is in progress, transmission or reception will not be stopped, but the SMOSIn and SCKn pin output will be placed in the high-impedance state after the completion of the transfer.

Control a general port pin to produce the SS output signal from the master.

40.8.3 SS Function in Slave Mode

Setting the CKE[1:0] bits in the SCR to 10b and the MSS bit in the SPMR to 1 selects slave operation. When the level on the SSn# pin is high, the SMISOn output pin will be in the high-impedance state and clock input through the SCKn pin will be ignored. When the level on the SSn# pin is low, clock input through the SCKn pin will be effective and transmission or reception can proceed.

If the input on the SSn# pin changes from low to high level during transmission or reception, the SMISOn output pin will be placed in the high-impedance state. Meanwhile, the internal processing for transmission or reception will continue at the rate of the clock input through the SCKn pin until processing for the character currently being transmitted or received is completed, after which it stops. At that time, an interrupt (the appropriate one from among TXI, RXI, and TEI) will be generated.

40.8.4 Relationship between Clock and Transmit/Receive Data

The CKPOL and CKPH bits in the SPMR can be used to set up the clock for use in transmission and reception in four different ways. The relation between the clock signal and the transmission and reception of data is shown in Figure 40.56. The relation is the same for both master and slave operation.

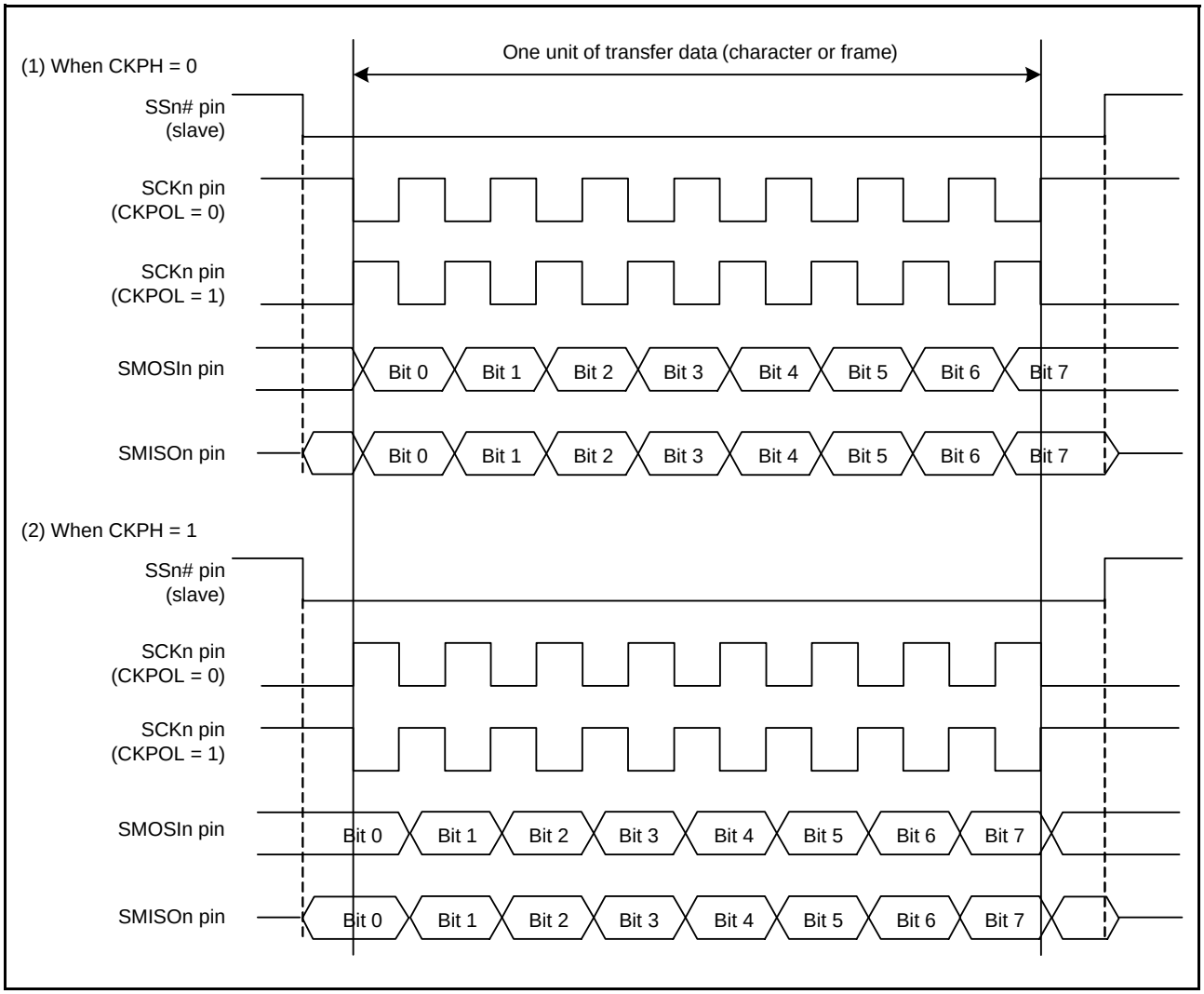


Figure 40.56 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode

40.8.5 SCI Initialization (Simple SPI Mode)

The procedure is the same as for initialization in clock synchronous mode Figure 40.23, Sample SCI Initialization Flowchart. The CKPOL and CKPH bits in the SPMR must be set to ensure that the kind of clock signal they select is suitable for both master and slave devices.
For initialization, changes to the operating mode, changes to the transfer format, and so on, initialize the SCR register before proceeding with changes.
As well as setting the RE bit to 0, note that the SSR.ORER, FER, and PER flags, as well as the RDR, are not initialized. Note that changing the value of the TE bit from 1 to 0 or from 0 to 1 will lead to the generation of a transmit data empty interrupt (TXI) if the value of the TIE bit in the SCR is 1 at the time.

40.8.6 Transmission and Reception of Serial Data (Simple SPI Mode)

In master operation, ensure that the SSn# pin of the slave device on the other side of the transfer is at the low level before starting the transfer and at the high level on completion of the transfer. Otherwise, the procedures are the same as in clock synchronous mode.

40.9 Bit Rate Modulation Function

The bit rate modulation function corrects the bit rate by thinning out the specified amount of clocks from those input to the baud rate generator.
When the SEMR.BRME bit is 1, the baud rate generator validates and counts the average interval of the number of clocks set in the MDDR register out of the total 256 clocks input.
Figure 40.57 assumes the SCI is in asynchronous mode, bits SMR.CKS[1:0] are 00b, the BRR register is 00h, and the MDDR register is 160. In this example, the cycle of the base clock is evenly corrected to 256/160, and the bit rate is corrected to 160/256. Note that there is an imbalance in thinning out the internal clock, and expansion and contraction occur in the pulse width of the internal base clock.

Note: Do not use this function in clock synchronous mode and in the highest speed settings in simple SPI mode (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).

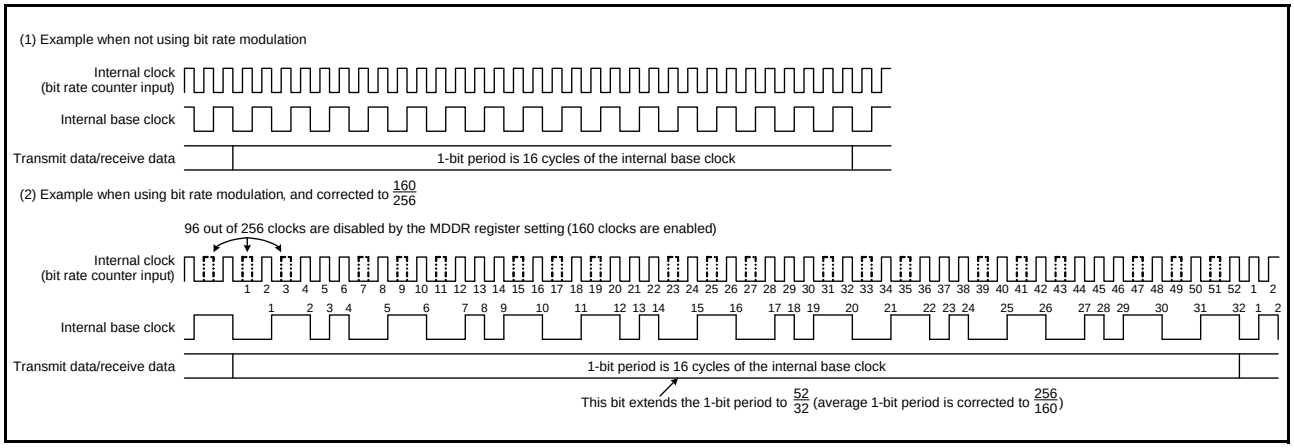


Figure 40.57 Example of the Base Clock When the Bit Rate Modulation Function is Used

40.10 Extended Serial Mode Control Section: Description of Operation

40.10.1 Serial Transfer Protocol

In conjunction with the SCIg module, the extended serial mode control section of the SCIh module can realize the serial transfer protocol composed of Start Frames and Information Frames that is shown in Figure 40.58.
A Start Frame is composed of a Break Field, Control Field 0, and Control Field 1. An Information Frame is composed of a number of Data Fields, a CRC16 Upper Field, and a CRC16 Lower Field.

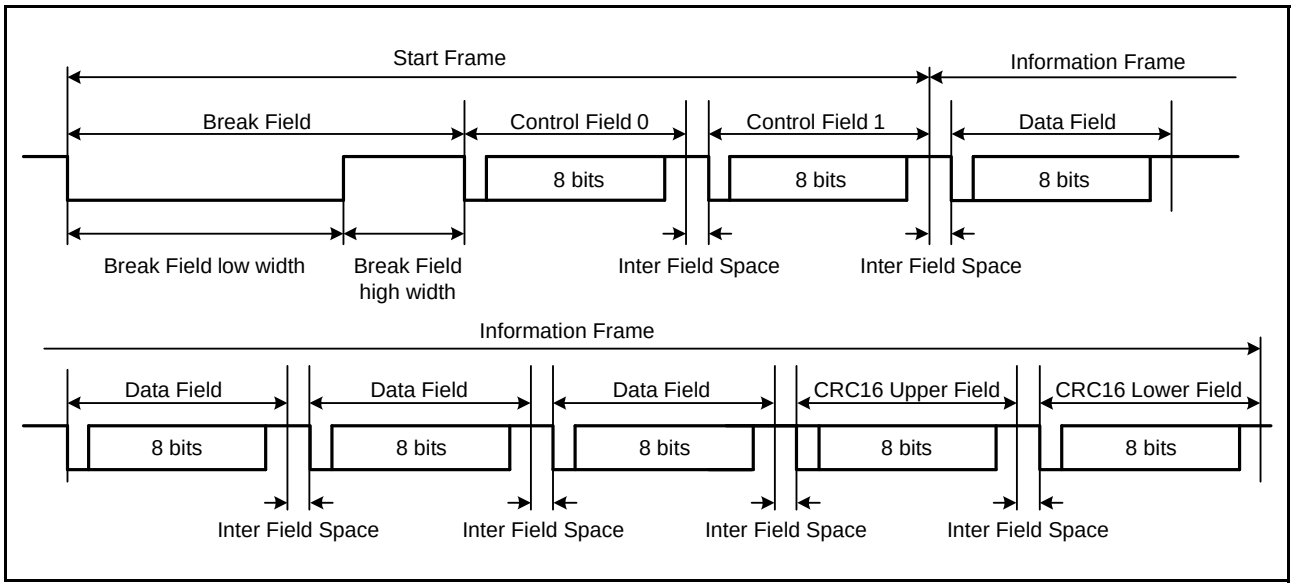


Figure 40.58 Protocol for Serial Transfer by the Extended Serial Mode Control Section

40.10.2 Transmitting a Start Frame

Figure 40.59 shows an example of operations to transmit a Start Frame, which is composed of the Break Field low width, Control Field 0, and Control Field 1. Figure 40.60 and Figure 40.61 are flowcharts for the transmission of a Start Frame.
Operations when the extended serial mode control section is to be used to transmit a Start Frame are as listed below. Be sure to use the SCI12 in asynchronous mode.

(1) With Break Field low width output mode as the operating mode for the timer, writing 1 to the TCR.TCST bit starts counting by the timer, and the low level will be output from the TXDX12 pin over the period corresponding to registers TCNT and TPRE settings.
(2) The output on the TXDX12 pin is inverted when the timer counter underflows, and the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1.
(3) Write 0 to the TCR.TCST bit to stop counting by the timer, and send the data for Control Field 0 by using SCI12. After the Break Field low width output, stop counting before the next underflow occurs.
(4) When the data for Control Field 0 have been transmitted, send the data for Control Field 1.
(5) When the data for Control Field 1 have been transmitted, send an Information Frame.

Omit the Break Field and Control Field 0 to suit the structure of the Start Frame.

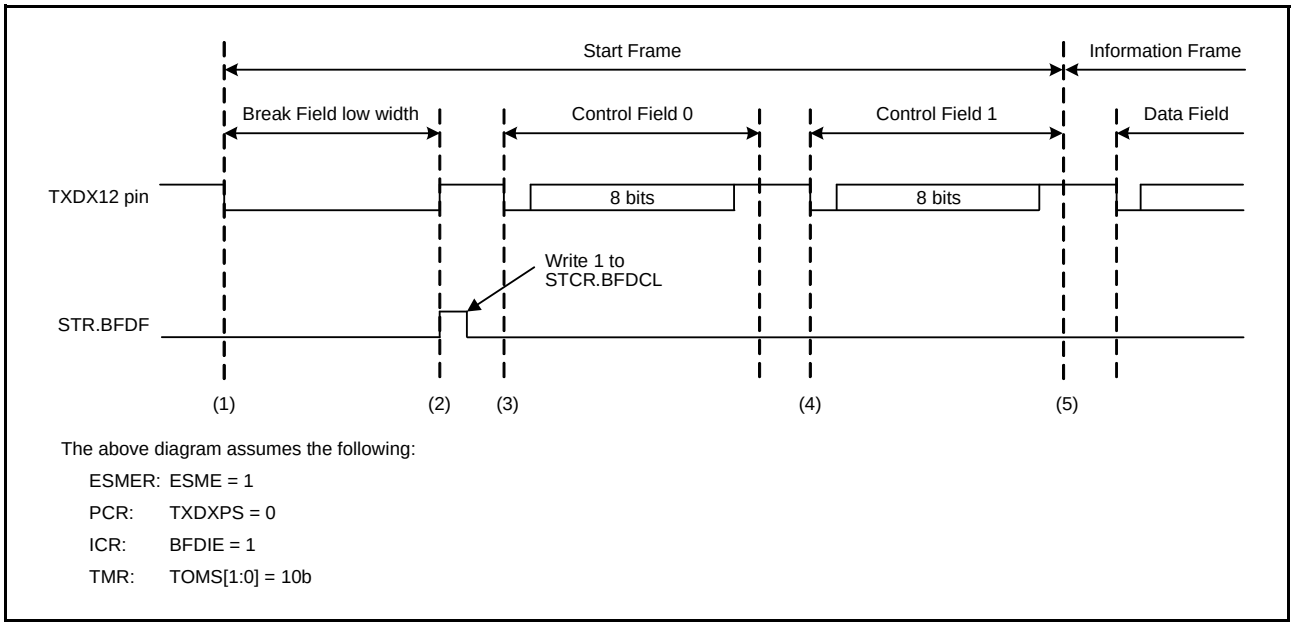


Figure 40.59 Example of Operations When Transmitting a Start Frame

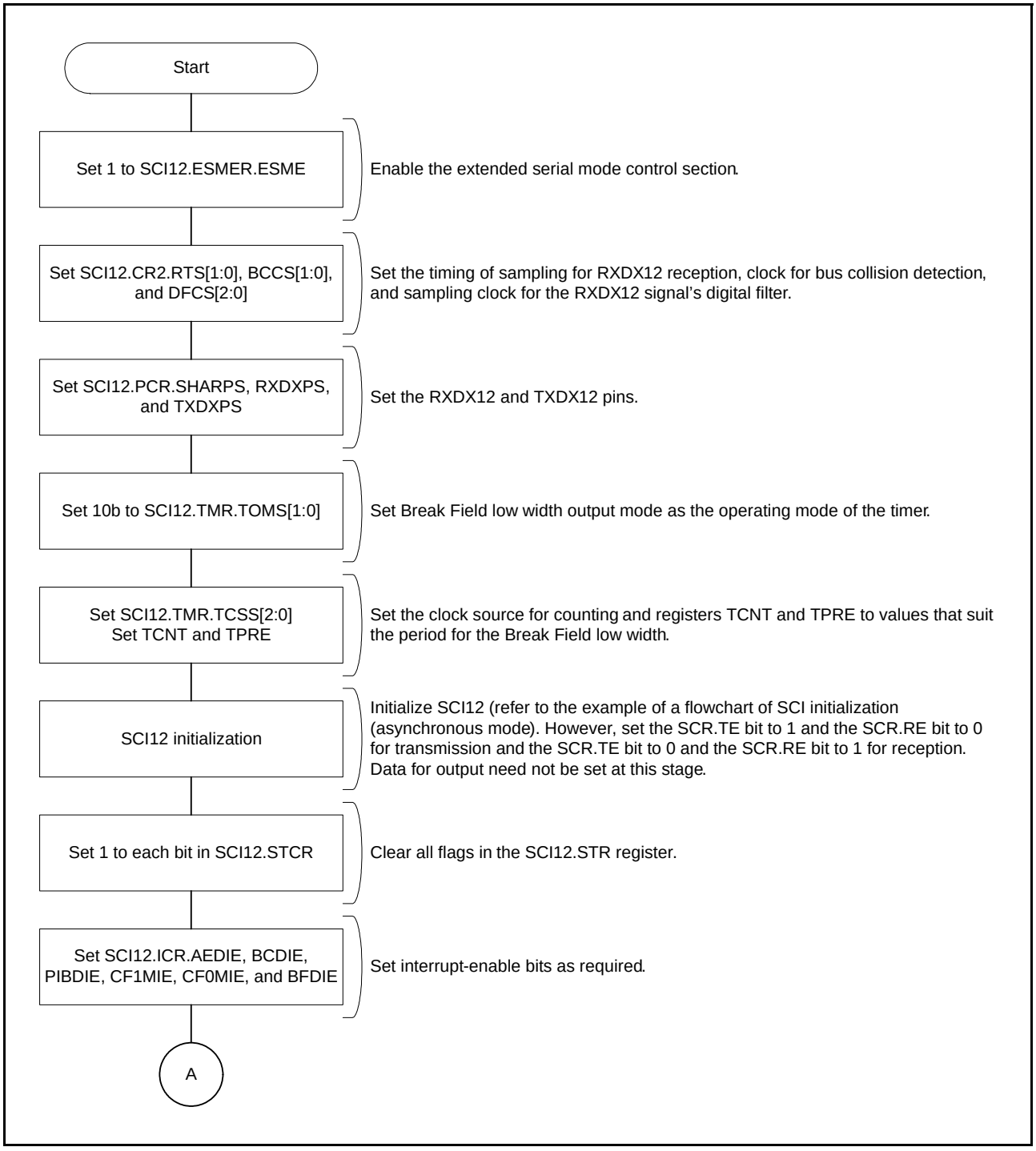


Figure 40.60 Example of Start Frame Transmission (1/2)

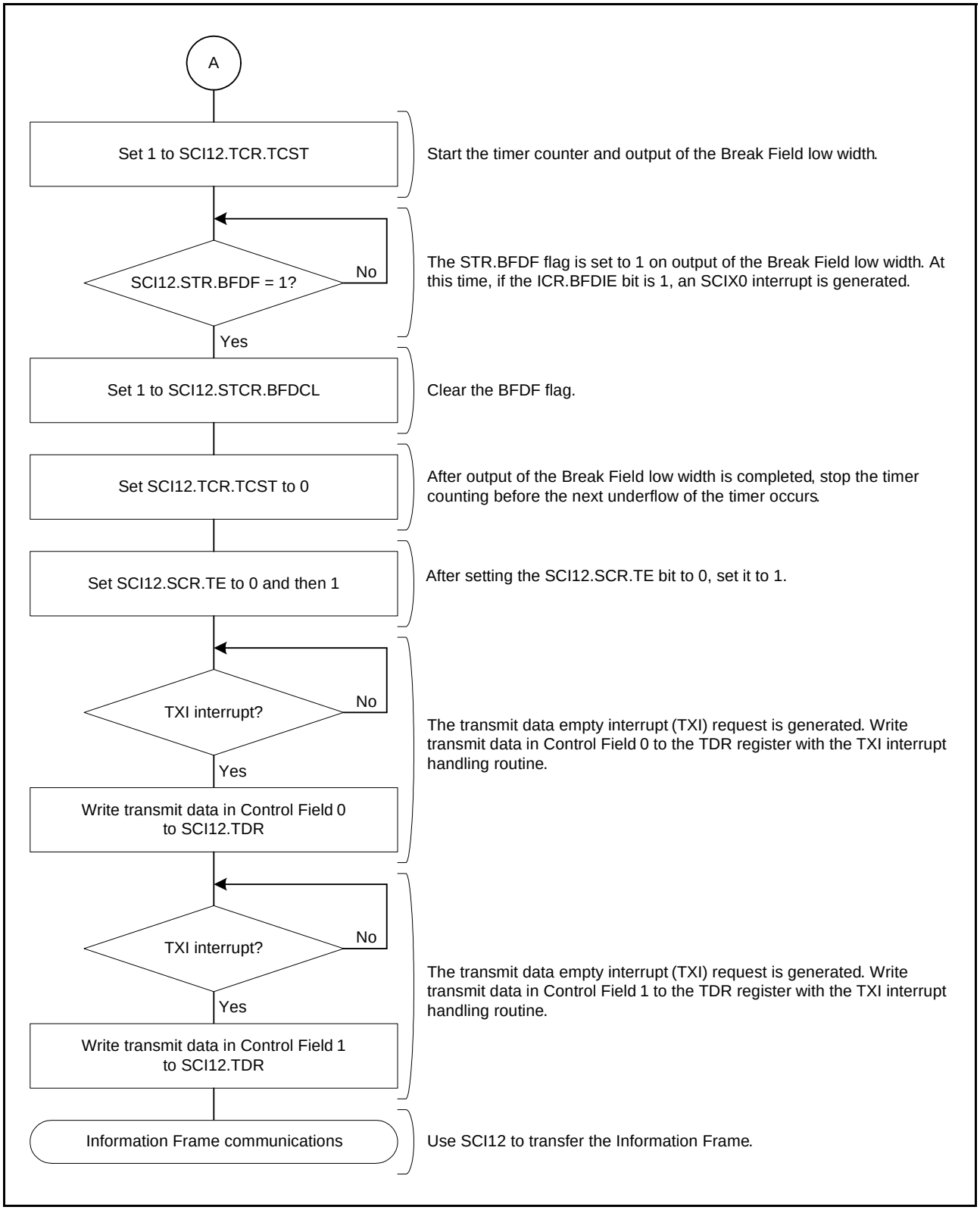


Figure 40.61 Example of Start Frame Transmission (2/2)

40.10.3 Receiving a Start Frame

The extended serial mode control section is capable of receiving Start Frames with the structures listed in Table 40.30.

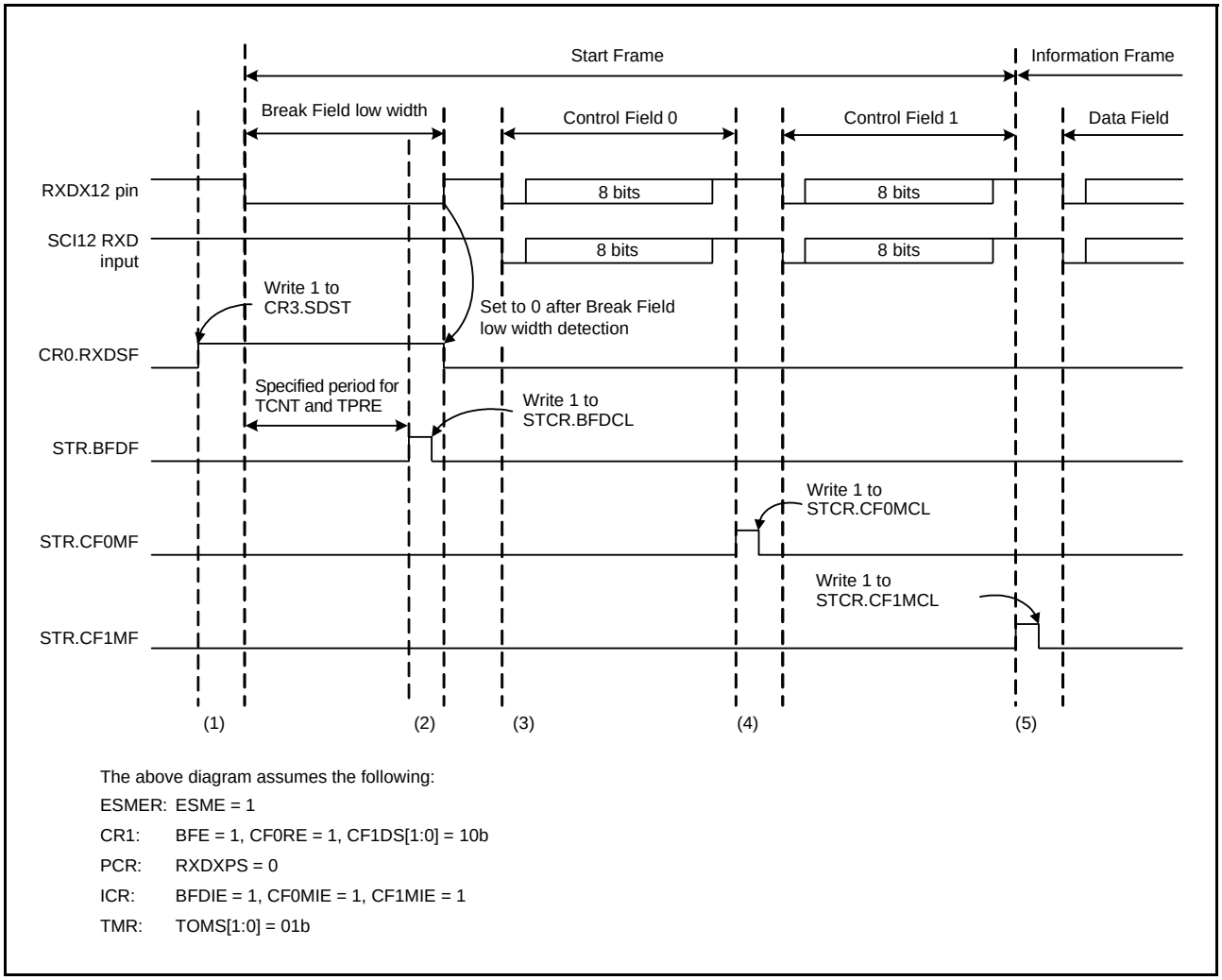
Table 40.30 Structures of Start Frames

Bit Setting		Structures of Start Frames
BFE	CF0RE	
0	0	Control Field 1 (8 bits) → Information Frame
0	1	Control Field 0 (8 bits) → Control Field 1 (8 bits) → Information Frame
1	0	Break Field (low width) → Control Field 1 (8 bits) → Information Frame
1	1	Break Field (low width) → Control Field 0 (8 bits) → Control Field 1 (8 bits) → Information Frame

Figure 40.62 shows an example of operations to receive a Start Frame, which is composed of the Break Field low width, Control Field 0, and Control Field 1. Figure 40.63 and Figure 40.64 are flowcharts for the reception of a Start Frame, and Figure 40.65 is a state transition diagram for the extended serial mode control section.
Operations when the extended serial mode control section is to be used to receive a Start Frame are as listed below. Be sure to use the SCI12 in asynchronous mode.

(1) With Break Field low width detection mode as the operating mode for the timer, writing 1 to the CR3.SDST bit enables detection of the Break Field low width. RXDX12 input to the SCI12 is disabled at this time.
(2) Low-level input on the RXDX12 pin continuing over a period longer than that corresponding to the settings of registers TCNT and TPRE is detected as the Break Field low width. At this time, the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1.
(3) When the input from the RXDX12 pin goes high after the Break Field low width, the CR0.RXDSF flag becomes 0 and reception of Control Field 0 by the SCI12 starts.
(4) If the data received in Control Field 0 match the data set in the CF0DR register, the STR.CF0MF flag is set to 1. An SCIX1 interrupt is also generated if the value of the ICR.CF0MIE bit is 1. Reception of Control Field 1 by the SCI12 starts after that. If the data received in Control Field 0 do not match the data set in the CF0DR register, a transition to the state prior to Break Field low width detection proceeds.
(5) If the data received in Control Field 1 match the data set in registers PCF1DR and SCF1DR, the STR.CF1MF flag is set to 1. An SCIX1 interrupt is also generated if the value of the ICR.CF1MIE bit is 1. Transfer of the Information Frame by the SCI12 starts after that. If the data received in Control Field 1 do not match the data set in either or both of registers PCF1DR and SCF1DR, a transition to the state prior to Break Field low width detection proceeds.

Omit the Break Field and Control Field 0 to suit the structure of the Start Frame.



The above diagram assumes the following:

ESMER: ESME = 1

CR1: BFE = 1, CF0RE = 1, CF1DS[1:0] = 10b

PCR: RXDXPS = 0

ICR: BFDIE = 1, CF0MIE = 1, CF1MIE = 1

TMR: TOMS[1:0] = 01b

Figure 40.62 Example of Operations at the Time of Start Frame Reception

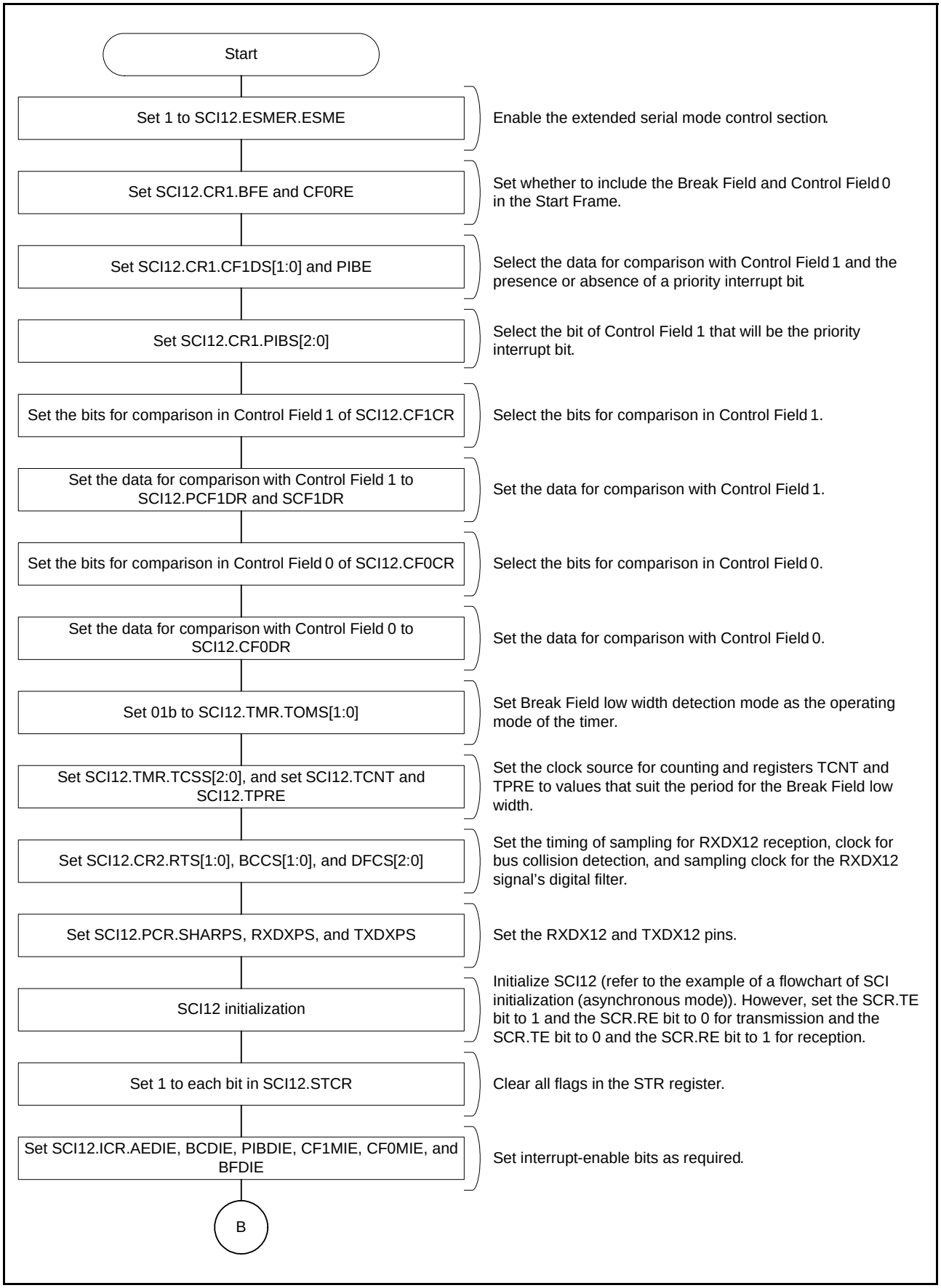


Figure 40.63 Sample Flowchart for Reception of a Start Frame (1)

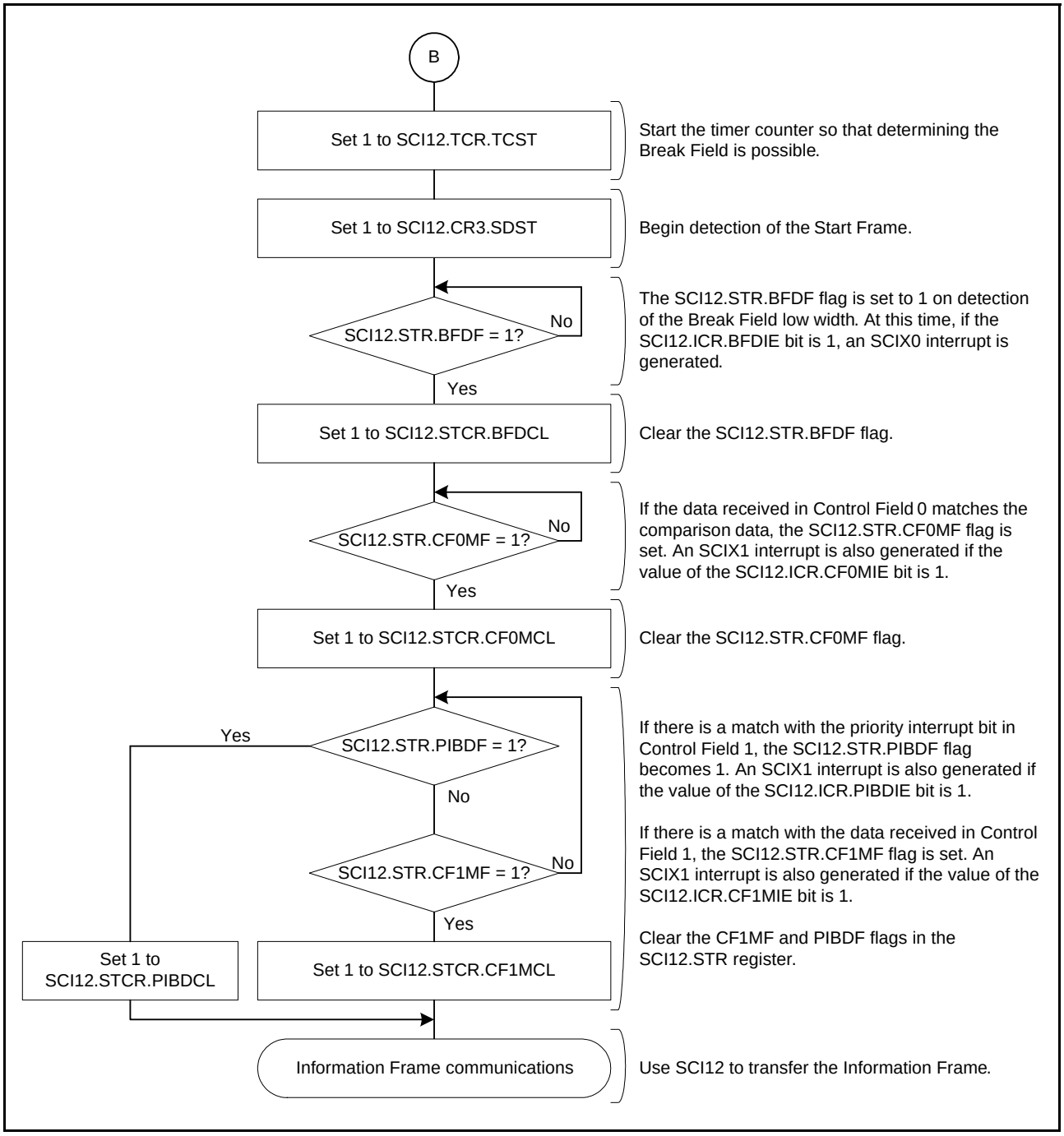


Figure 40.64 Sample Flowchart for Reception of a Start Frame (2)

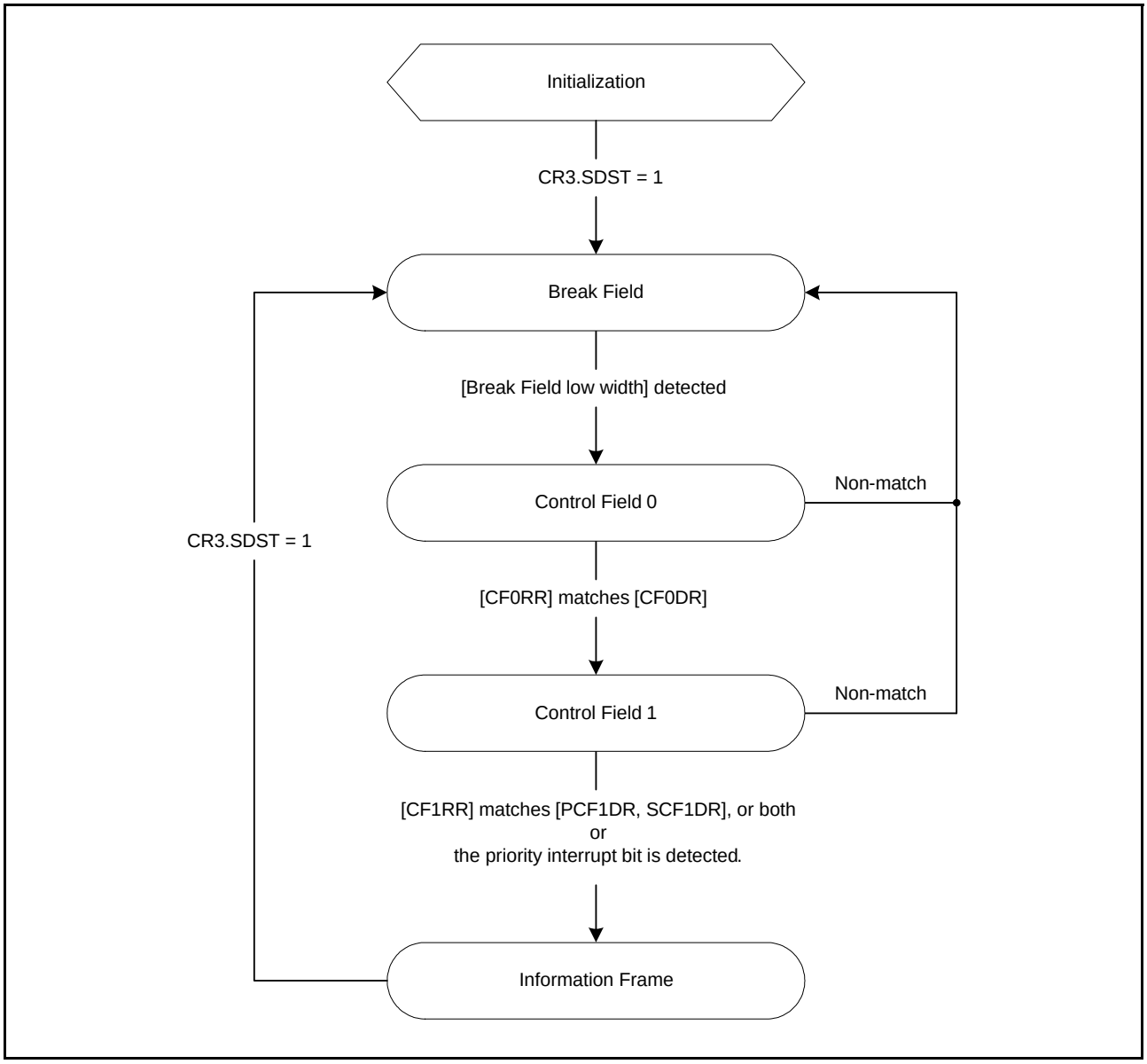


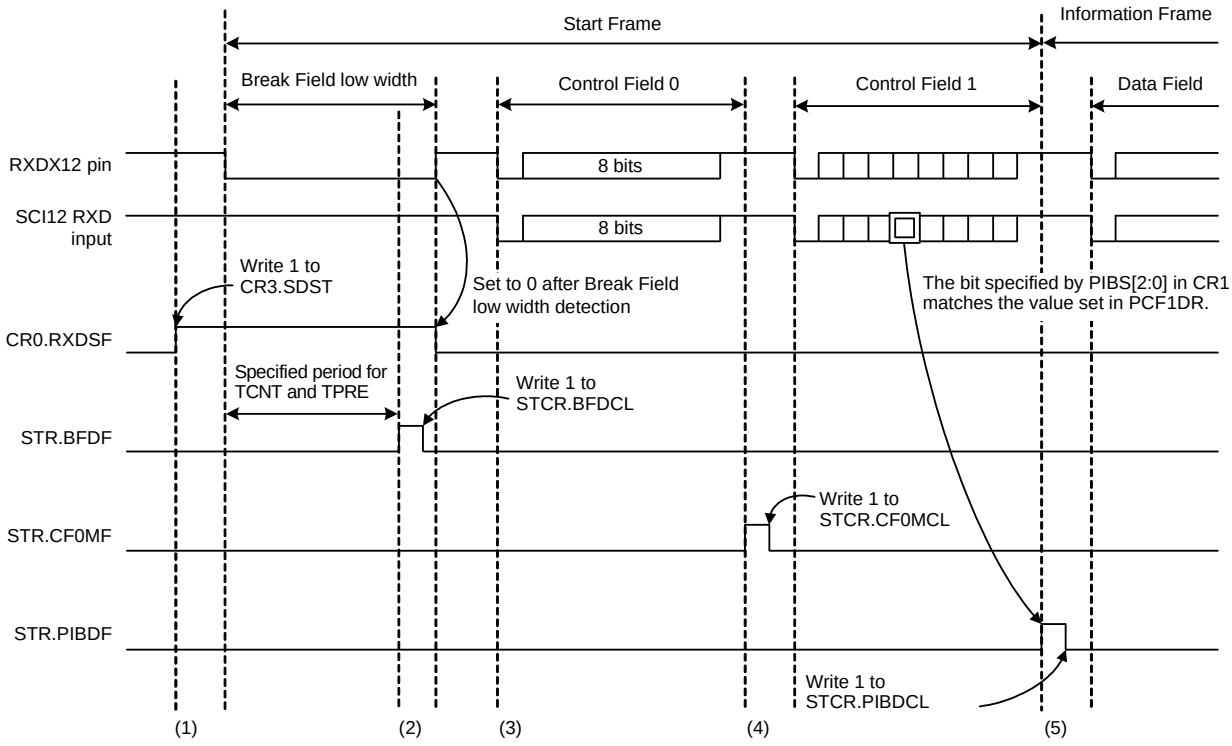
Figure 40.65 State Transitions When Receiving a Start Frame

40.10.3.1 Priority Interrupt Bit

Figure 40.66 shows an example of operation in Start Frame reception where a priority interrupt bit is in use. Setting the CR1.PIBE bit to 1 enables the use of a priority interrupt bit.
Operations of the extended serial mode control section in start Frame reception where a priority interrupt bit is in use are as described below.

Steps (1) to (4) are the same as in Figure 40.62, for Start Frame reception.

(5) If the value of the bit selected by the CR1.PIBS[2:0] bits matches the corresponding bit in the PCF1DR register, the STR.PIBDF flag is set to 1. An SCIX1 interrupt is also generated if the value of the ICR.PIBDIE bit is 1. Transfer of the Information Frame by the SCI12 starts after that. If the data received in Control Field 1 do not match the data set in either or both of registers PCF1DR and SCF1DR and the priority interrupt bit is not detected, a transition to the state prior to Break Field low width detection proceeds.



The above diagram assumes the following:

ESMER: ESME = 1
CR1: BFE = 1, PIBE = 1, CF0RE = 0, CF1DS[1:0] = 10b, PIBS[2:0] = 011b
PCR: RXDXPS = 0
ICR: BFDIE = 1, CF0MIE = 1, PIBDIE = 1
TMR: TOMS[1:0] = 01b

Figure 40.66 Example of Operations When Receiving a Start Frame While the CR1.PIBE Bit is 1

40.10.4 Detection of Bus Collisions

Detection of bus collisions operate for cases where output of the Break Field low width and transmission of data by the SCI12 are in progress when the ESMER.ESME bit and the SCI12.SCI.TE bit are set to 1.

Figure 40.67 shows an example of operations with bus collision detection. Signals output through TXDX12 and input through RXDX12 are sampled with the bus collision detection clock set with the CR2.BCCS[1:0] bits as the sampling clock, and the STR.BCDF flag is set to 1 if the signals fail to match three times in a row. An SCIX2 interrupt is also generated if the value of the ICR.BCDIE bit is 1.

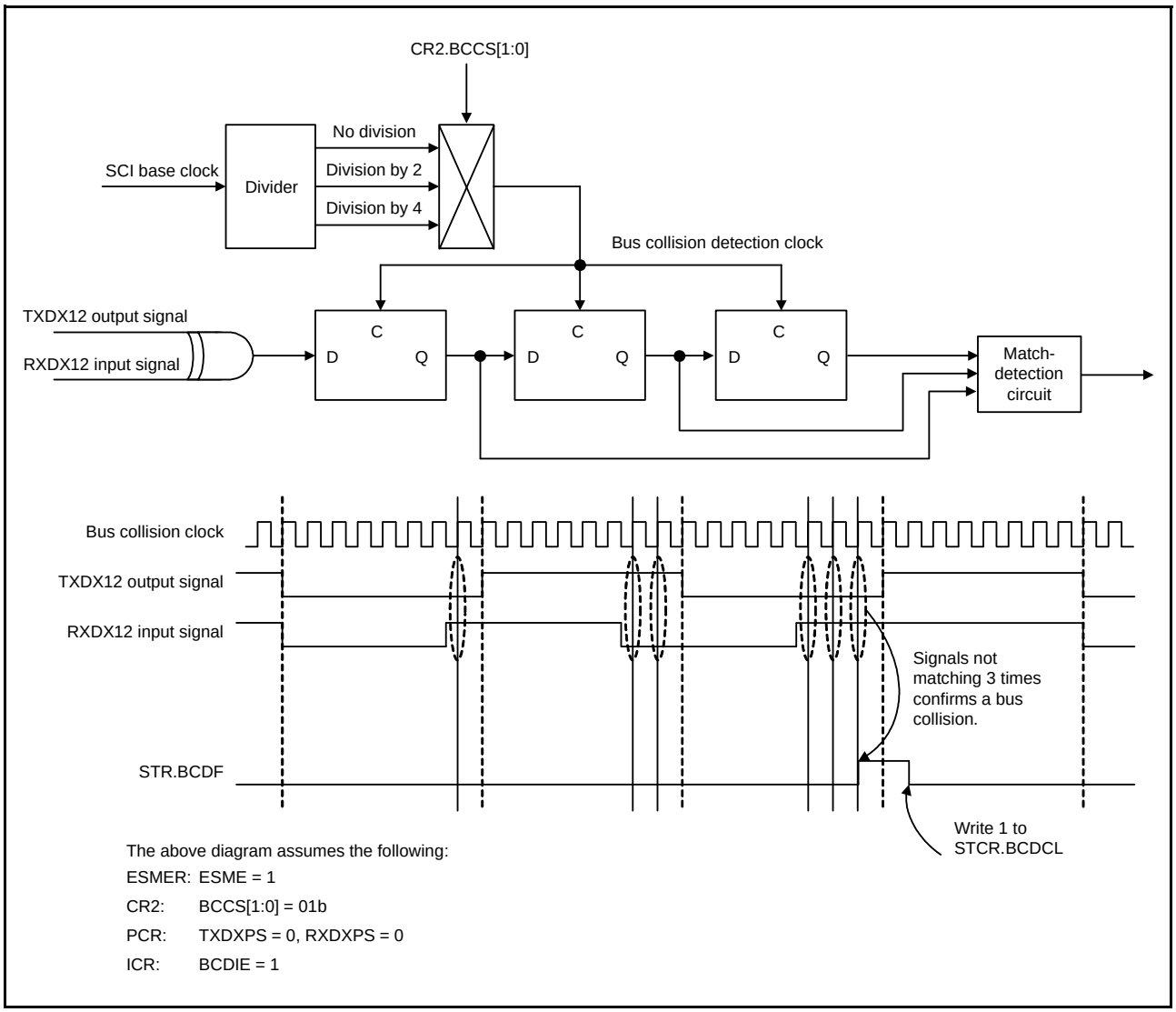


Figure 40.67 Example of Operations with Bus Collision Detection

40.10.5 Digital Filter for Input on the RXDX12 Pin

Signals input through the RXDX12 pin can be passed through a digital filter before they are conveyed to the internal circuits. The digital filter consists of three flip-flop circuit stages connected in series and a match-detecting circuit. The CR2.DFCS[2:0] bits select the sampling clock for the RXDX12 pin input signals. If the outputs of all three latches match, the given level is conveyed to subsequent circuits. If the levels do not match, the previous value is retained. In other words, levels are confirmed as being the signal if they are retained for at least three cycles of the sampling clock but judged to be noise rather than changes in the signal level if they change within three cycles of the sampling clock. Figure 40.68 shows an example of operations with the digital filter.

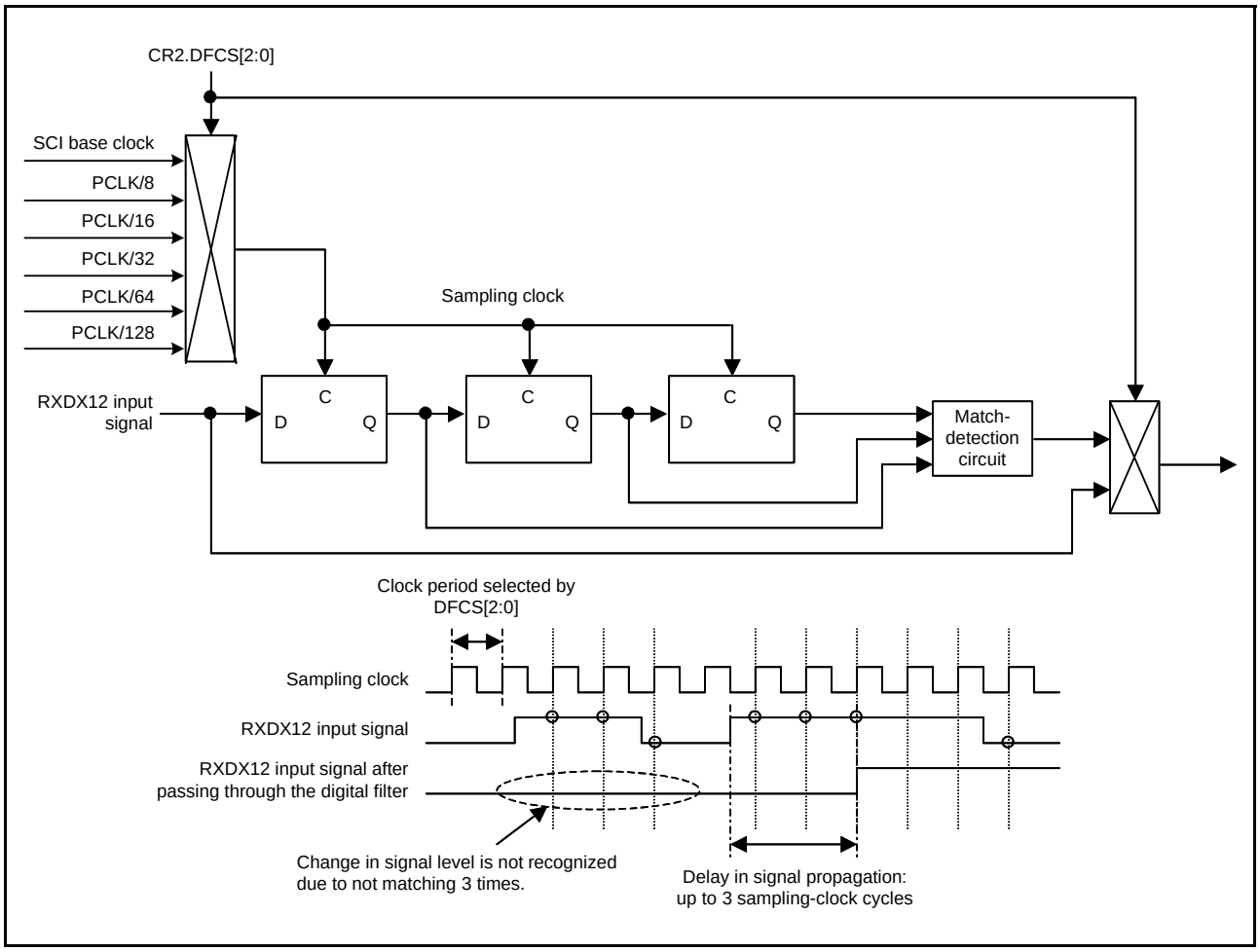
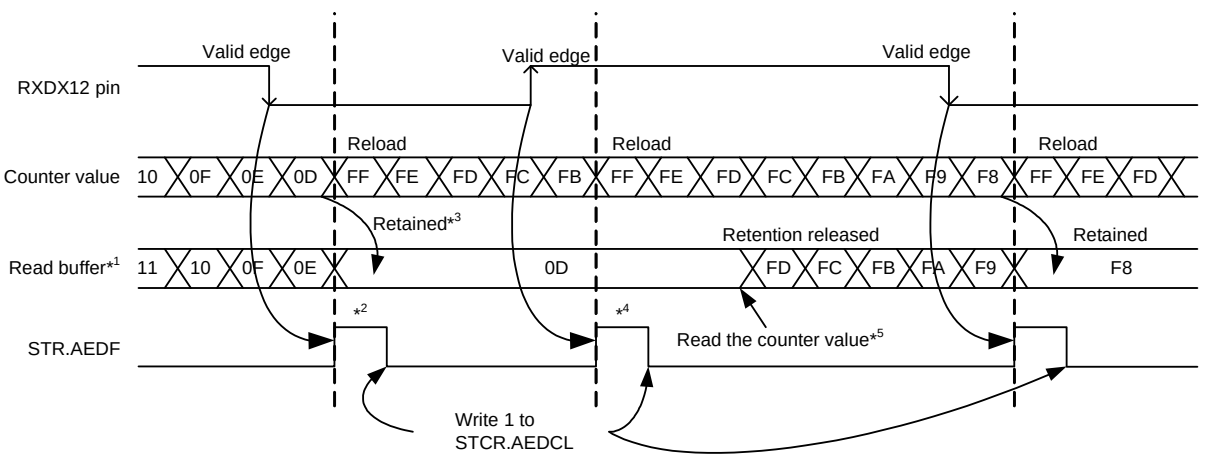


Figure 40.68 Example of Operations with the Digital Filter

40.10.6 Bit Rate Measurement

The bit rate measurement function measures the intervals between rising and falling edges and between falling and rising edges of the signal input from the RXDX12 pin. Figure 40.69 shows an example of operations for bit rate measurement.

(1) Writing 1 to the CR0.BRME bit enables bit rate measurement. Only set the BRME bit to 1 when you wish to proceed with bit rate measurement. Furthermore, bit rate measurement will not proceed during a Break Field, even if the BRME bit is set to 1.
(2) After detection of the Break Field low width, bit rate measurement starts when the level input on the RXDX12 pin becomes high.
(3) Once bit rate measurement has started, counter values from the timer are retained in the read buffers on the input of valid edges from the RXDX12 pin (rising and falling edges) and the counter is reloaded. An SCIX3 interrupt is also generated if the value of the ICR.AEDIE bit is 1. Retention by registers TCNT and TPRE is released by reading these registers.
(4) The bit rate as calculated from the values counted during intervals between valid edges can be used for adjusting the rate by changing the settings of the SCI12. To disable the bit rate measurement after a match with Control Field 1, write 0 to the CR0.BRME bit.



Note 1. The TCNT or TPRE register indicates the contents of the read buffers.
Note 2. The STR.AEDF flag becomes 1 on detection of a valid edge.
Note 3. When an effective edge is detected, the current value of the counter is stored in the read buffer.
Note 4. The value retained in the read buffers due to one valid edge is not updated to the new value even if a further valid edge is detected.
Note 5. Retention of a value in a read buffer due to a valid edge is released by reading the TCNT or TPRE register.

Figure 40.69 Example of Operations for Bit Rate Measurement

40.10.7 Selectable Timing for Sampling Data Received through RXDX12

The extended serial mode control section provides a way of adjusting the timing for the sampling of data received through the RXDX12 pin of an SCI12 by setting the CR2.RTS[1:0] bits to select the rising edges of 8th, 10th, 12th, or 14th cycle of the SCI base clock. If the value of the SEMR.ABCS bit is 1, the bits select the rising edges of 4th, 5th, 6th, or 7th cycle of the PCLK clock of the SCI12. Figure 40.70 shows timing for the sampling of data received through RXDX12.

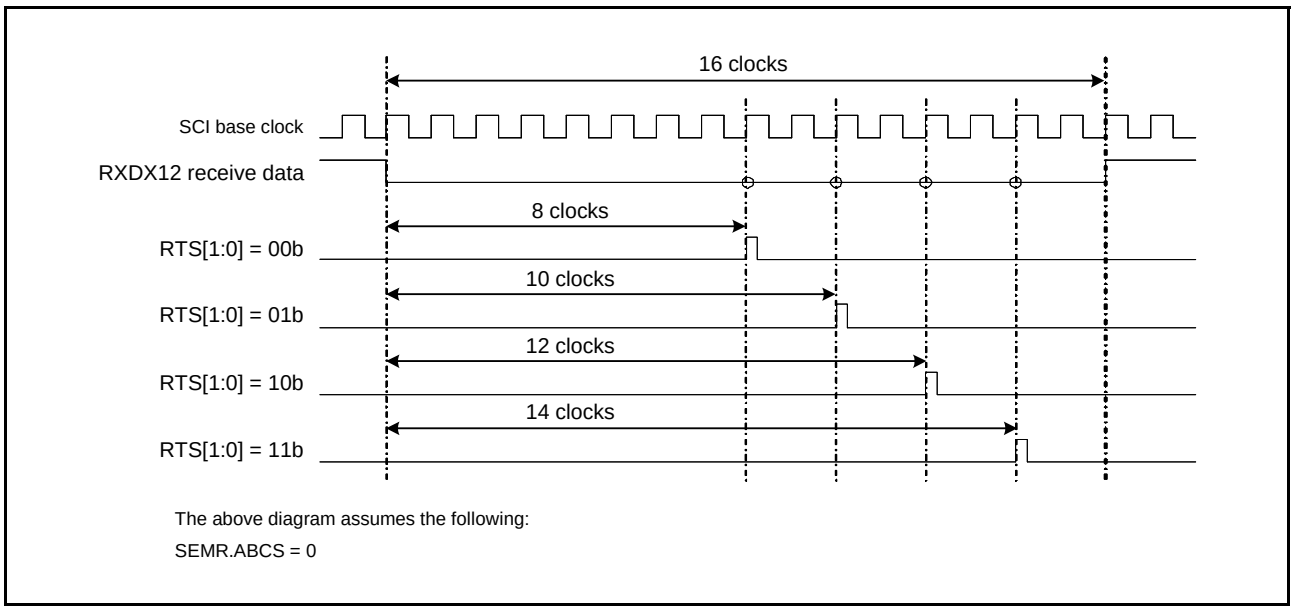


Figure 40.70 Timing for Sampling of Data Received through RXDX12

40.10.8 Timer

The timer has the following operating modes.

(1) Break Field Low Width Output Mode

This mode is for output through the TXDX12 pin of the low level over the Break Field low width at the transmission of a Start Frame. Setting the TMR.TOMS[1:0] bits to 10b switches operation to Break Field low width output mode. The TMR.TCSS[2:0] bits select the clock source for the counter. When the TCR.TCST bit is set to 1, the output on the TXDX12 pin goes to the low level and counting starts. When the timer underflows, the output on the TXDX12 pin goes to the high level and the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1. When 0 is written to the TCR.TCST bit, counting stops after reloading of registers TPRE and TCNT. After output of the Break Field low width is completed, stop the timer before it underflows again. Figure 40.71 shows an example of operations in Break Field low width output mode.

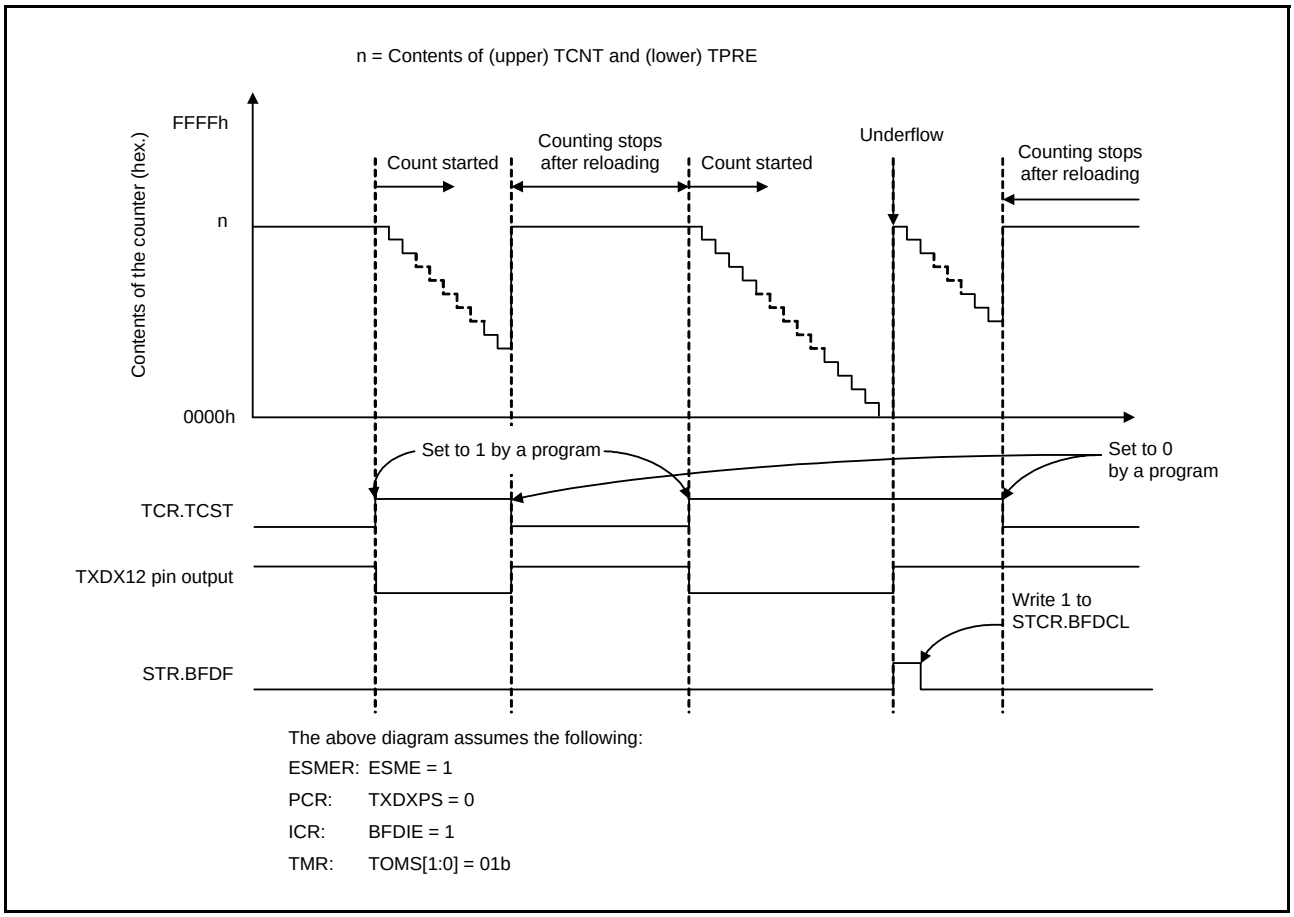


Figure 40.71 Example of Operations in Break Field Low Width Output Mode

(2) Break Field Low Width Determination Mode

This mode is for determining the Break Field low width in the input signal on the RXDX12 pin at the reception of a Start Frame. Setting the TMR.TOMS[1:0] bits to 01b switches operation to Break Field low width determination mode. The TMR.TCSS[2:0] bits select the clock source for the counter. When the TCR.TCST bit is set to 1, the interface enters the Break Field low width determinable state. Determination starts when a low level is input from the RXDX12 pin. When a high level is then input on the RXDX12 pin, registers TPRE and TCNT are reloaded and the interface enters the Break Field low width determinable state. When the timer underflows during Break Field low width determination, the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1. If an underflow of the timer during data transfer cause a problem in the form of interrupt generation, stop the timer after Break Field low width determination. Figure 40.72 shows an example of operations in Break Field low width output mode.

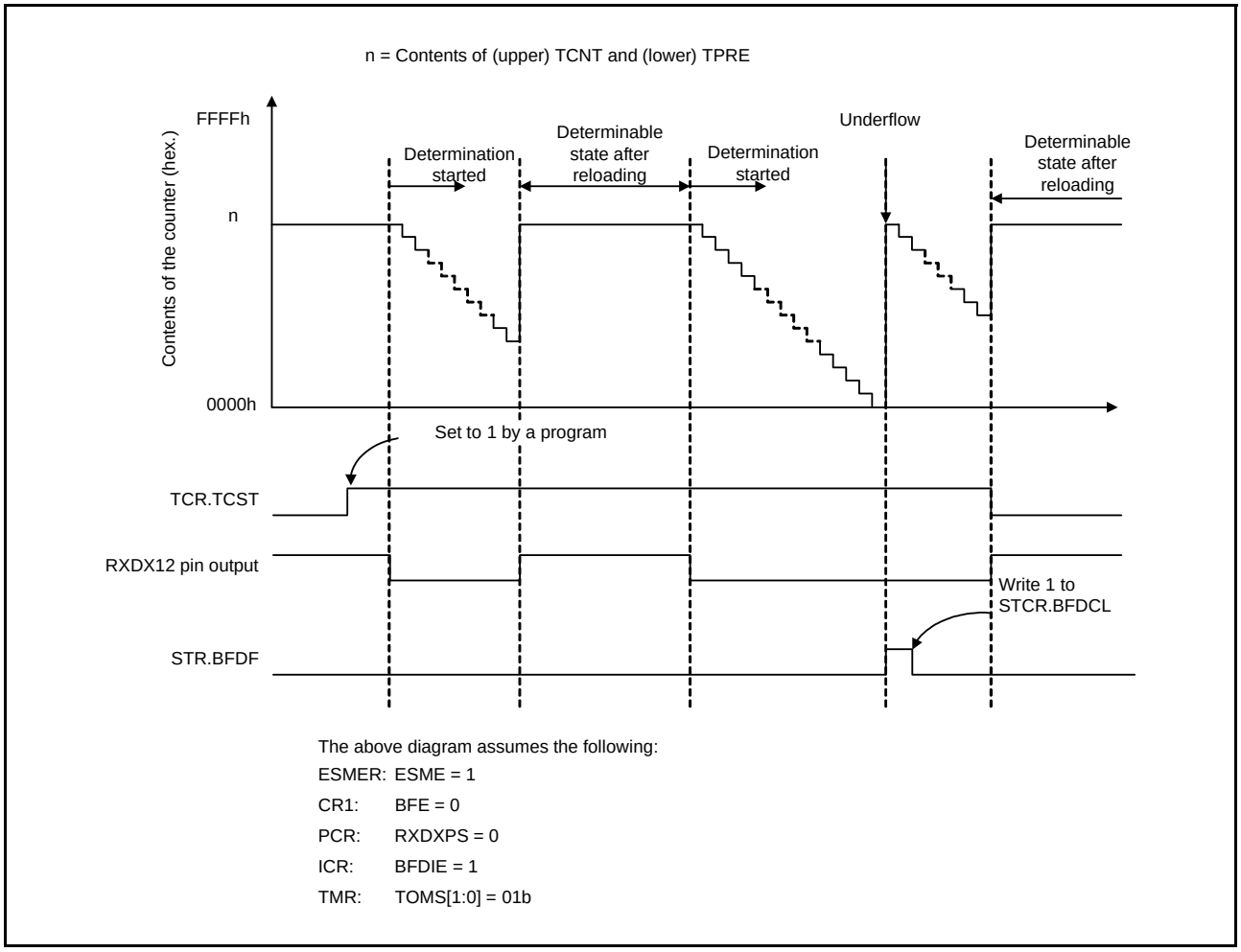


Figure 40.72 Example of Operations in Break Field Low Width Determination Mode

(3) Timer Mode

This mode is for counting cycles of the internal clock as the clock source. Setting the TMR.TOMS[1:0] bits to 00b switches operation to timer mode. The TMR.TCSS[2:0] bits select the clock source for the counter. Counting starts when 1 is written to the TCR.TCST bit and stops when 0 is written to the TCST bit. Registers TPRE and TCNT both count down. The TPRE register counts cycles of the clock source for counting, and underflows of the TPRE register provide the clock source for counting by the TCNT register. When the timer underflows, the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1.

40.11 Noise Cancellation Function

Figure 40.73 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of two stages of flip-flop circuits and a match-detection circuit. When the level on the pin matches in three consecutive samples taken at the set sampling interval, the matching level continues to be conveyed internally until the level on the pin again matches in three consecutive samples.

In asynchronous mode, the noise cancellation function can be applied on the RXDn input signal. The period of the base clock (1/16th of a bit-period when SEMR.ABCS = 0 and 1/8th of a bit-period when SEMR.ABCS = 1) is the sampling interval.

In simple I2C mode, the noise cancellation function can be applied on the SSDAn and SSCLn input signals. The sampling clock is the clock signal produced by frequency-dividing the signal from the clock source for the internal baud-rate generator by one, two, four, or eight as selected by the setting of the SNFR.NFCS[2:0] bits.

If the base clock is stopped with the noise filter enabled and then the clock input is started again, the noise filter operation resumes from where the clock was stopped. If SCR.TE and SCR.RE are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, it is determined that a level match is detected and is conveyed to the internal signal. When the level being input corresponds to 0, the initial output of the noise filter is retained until the level matches in three consecutive samples.

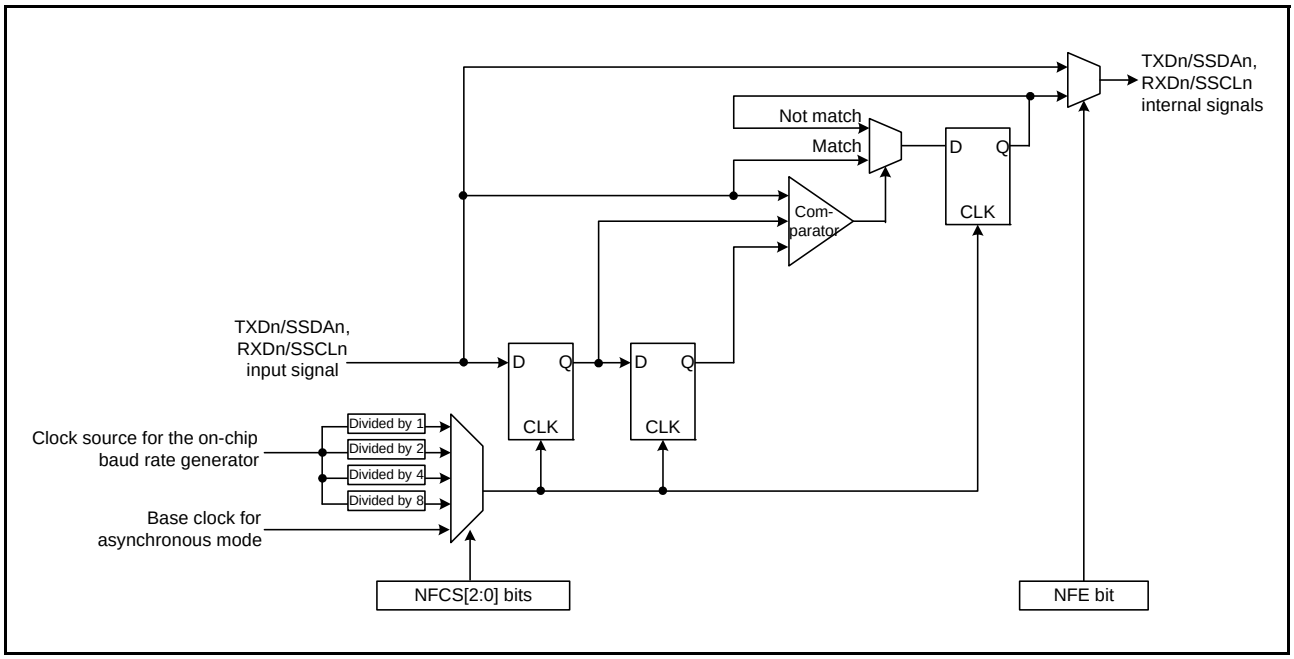


Figure 40.73 Block Diagram of Digital Noise Filter Circuit

40.12 Interrupt Sources

40.12.1 Buffer Operations for TXI and RXI Interrupts

If the conditions for a TXI and RXI interrupt are satisfied while the interrupt status flag in the interrupt controller is 1, the SCI does not output the interrupt request but retains it internally (with a capacity for retention of one request per source). When the value of the interrupt status flag in the interrupt controller becomes 0, the interrupt request retained within the SCI is output. The internally retained interrupt request is automatically discarded once the actual interrupt is output. Clearing of the corresponding interrupt enable bit (the TIE or RIE bit in the SCR) can also be used to discard an internally retained interrupt request.

40.12.2 Interrupts in Asynchronous Mode, Clock Synchronous Mode, and Simple SPI Mode

Table 40.31 lists interrupt sources in asynchronous mode, clock synchronous mode, and simple SPI mode. A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled with the enable bits in the SCR register.

If the SCR.TIE bit is 1, a TXI interrupt request is generated when transmit data is transferred from the TDR or TDRL register*1 to the TSR. A TXI interrupt request can also be generated by setting the SCR.TE bit to 1 after setting the SCR.TIE bit to 1 or by using a single instruction to set the SCR.TE and SCR.TIE bit to 1 at the same time. A TXI interrupt request can activate the DTC or DMAC to handle data transfer.

A TXI interrupt request is not generated by setting the SCR.TE bit to 1 while the setting of the SCR.TIE bit is 0 or by setting the SCR.TIE bit to 1 while the setting of the SCR.TE bit is 1.*2

When new data is not written by the time of transmission of the last bit of the current transmit data and the setting of the SCR.TEIE bit is 1, the SSR.TEND flag becomes 1 and a TEI interrupt request is generated. Furthermore, when the setting of the SCR.TE bit is 1, the SSR.TEND flag retains the value 1 until further transmit data are written to the TDR or TDRL register*1, and setting the SCR.TEIE bit to 1 leads to the generation of a TEI interrupt request.

Writing data to the TDR or TDRL register*1 leads to clearing of the SSR.TEND flag and, after a certain time, discarding of the TEI interrupt request.

If the SCR.RIE bit is 1, an RXI interrupt request is generated when received data is stored in the RDR. An RXI interrupt request can activate the DTC or DMAC to handle data transfer.

Setting of any from among the ORER, FER, and PER flags in the SSR to 1 while the SCR.RIE bit is 1 leads to the generation of an ERI interrupt request. An RXI interrupt request is not generated at this time. Clearing all three flags (ORER, FER, and PER) leads to discarding of the ERI interrupt request.

Note 1. In the case where asynchronous mode and 9-bit data length are selected

Note 2. To temporarily disable TXI interrupts at the time of transmission of the last of the data and so on when you wish a new round of transmission to start after handling of the transmission-completed interrupt, control disabling and enabling of the interrupt by using the interrupt request enable bit in the interrupt controller rather than using the SCR.TIE bit. This can prevent the suppression of TXI interrupt requests in the transfer of new data.

Table 40.31 Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation	Priority
ERI	Receive error	ORER, FER, or PER	Not possible	Not possible	High
RXI	Receive data full	RDRF	Possible	Possible	↑
TXI	Transmit data empty	TDRE	Possible	Possible	
TEI	Transmit end	TEND	Not possible	Not possible	Low

40.12.3 Interrupts in Smart Card Interface Mode

Table 40.32 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 40.32 SCI Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation	Priority
ERI	Receive error or error signal detection	ORER, PER, or ERS	Not possible	Not possible	High
RXI	Receive data full	—	Possible	Possible	↑
TXI	Transmit data empty	TEND	Possible	Possible	Low

Data transmission/reception using the DTC or DMAC is also possible in smart card interface mode, similar to in the normal SCI mode. In transmission, when the TEND flag in the SSR register is set to 1, a TXI interrupt request is generated. This TXI interrupt request activates the DTC or DMAC allowing transfer of transmit data if the TXI request is specified beforehand as a source of DTC or DMAC activation. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During the retransmission, the TEND flag is kept to 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, the ERS flag in the SSR register is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the RIE bit in the SCR register to 1 to enable an ERI interrupt request to be generated at error occurrence.

When transmitting/receiving data using the DTC or DMAC, be sure to make settings to enable the DTC or DMAC before making SCI settings. For DTC or DMAC settings, refer to section 18, DMA Controller (DMACAa) and section 20, Data Transfer Controller (DTCa).

In reception, an RXI interrupt request is generated when receive data is set to RDR. This RXI interrupt request activates the DTC or DMAC allowing transfer of receive data if the RXI request is specified beforehand as a source of DTC or DMAC activation. If an error occurs, the error flag is set. Therefore, the DTC or DMAC is not activated and an ERI interrupt request is issued to the CPU instead; the error flag must be cleared.

40.12.4 Interrupts in Simple I2C Mode

The interrupt sources in simple I2C mode are listed in Table 40.33. The STI interrupt is allocated to the transmit end interrupt (TEI) request. The receive error interrupt (ERI) request cannot be used.

The DTC or DMAC can also be used to handle transfer in simple I2C mode.

When the value of the IICINTM bit in the SIMR2 register is 1, an RXI request will be generated on the falling edge of the SSCLn signal for the eighth bit. If the RXI has been set up as an activating request for the DTC or DMAC beforehand, the RXI request will activate the DTC or DMAC to handle transfer of the received data. Furthermore, a TXI request is generated on the falling edge of the SSCLn signal for the ninth bit (acknowledge bit). If the TXI has been set up as an activating request for the DTC or DMAC beforehand, the TXI request will activate the DTC or DMAC to handle transfer of the transmit data.

When the value of the IICINTM bit in the SIMR2 register is 0, an RXI request (ACK detection) if the input on the SSDAn pin is at the low level or a TXI request (NACK detection) if the input on the SSDAn pin is at the high level will be generated on the rising edge of the SSCLn signal for the ninth bit (acknowledge bit). If the RXI has been set up as an activating request for the DTC or DMAC beforehand, the RXI request will activate the DTC or DMAC to handle transfer of the received data.

Also, if the DTC or DMAC is used for data transfer in reception or transmission, be sure to set up and enable the DTC or DMAC before setting up the SCI.

When the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits in the SIMR3 register are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 40.33 SCI Interrupt Sources

Name	Interrupt Source		Interrupt Flag	DTC Activation	DMAC Activation	Priority
	IICINTM bit = 0	IICINTM bit = 1				
RXI	ACK detection	Reception	—	Possible	Possible	High
TXI	NACK detection	Transmission	—	Possible*1	Possible*1	
STI	Completion of generation of a start, restart, or stop condition		IICSTIF	Not possible	Not possible	Low

Note 1. Activation of the DTC or DMAC is only possible when the SIMR2.IICINTM bit is 1 (use reception and transmission interrupts).

40.12.5 Interrupt Requests from the Extended Serial Mode Control Section

The extended serial mode control section has a total of six types of interrupt request for generating the SCIX0 interrupt (Break Field low width detected), SCIX1 interrupt (Control Field 0 match, Control Field 1 match, priority interrupt bit detected), SCIX2 interrupt (bus collision detected), and SCIX3 interrupt (valid edge detected). When any of the interrupt factors is generated, the corresponding status flag is set to 1. Details of all of the interrupt requests are listed in Table 40.34.

Table 40.34 Interrupt Sources of the Extended Serial Mode Control Section

Interrupt Request	Status Flag	Interrupt Factors
SCIX0 interrupt (Break Field low width detected)	BFDF	• Detection of a Break Field low width longer than the interval corresponding to the timer setting • Completion of the output of a Break Field low width over the interval corresponding to the timer setting • Underflow of the timer
SCIX1 interrupt (Control Field 0 match)	CF0MF	The data received in Control Field 0 matching the value set in CF0DR
SCIX1 interrupt (Control Field 1 match)	CF1MF	The data received in Control Field 1 matching the value set in PCF1DR or SCF1DR
SCIX1 interrupt (priority interrupt bit detected)	PIBDF	The value of the bit specified as the priority interrupt bit matching the value set in PCF1DR
SCIX2 interrupt (bus collision detected)	BCDF	The output level on the TXDX12 pin and the input level on the RXDX12 pin not matching on three consecutive cycles of the bus collision detection clock
SCIX3 interrupt (valid edge detected)	AEDF	Detection of a valid edge during bit rate measurement

40.13 Event Linking

By employing interrupt request signals as event signals, SCI5 is able to provide linked operation through the event link controller (ELC) for modules selected in advance.
Event signals can be output regardless of the values of the corresponding interrupt request enable bits.

(1) Error (receive error, error signal detected) event output

- Indicates abnormal termination due to a parity error during reception in asynchronous mode.
- Indicates abnormal termination due to a framing error during reception in asynchronous mode.
- Indicates abnormal termination due to an overrun error during reception.
- Indicates detection of the error signal during transmission in smart card interface mode.

(2) Receive data full event output

- Indicates that received data have been set in the receive data register (RDR or RDRL).
- Indicates that ACK has been detected if the SIMR2.IICINTM bit is 0 in simple I^2C mode.
- Indicates that the 8th-bit SSCL5 falling edge has been detected if the SIMR2.IICINTM bit is 1 in simple I^2C mode.
- When the SIMR2.IICINTM bit is 1 during master transmission in simple I^2C mode, set the event link controller (ELC) so that receive data full events are not used.

(3) Transmit data empty event output

- Indicates that the SCR.TE bit has been changed from 0 to 1.
- Indicates that transmit data have been transferred from the transmit data register (TDR or TDRL) to the transmit shift register (TSR).
- Indicates that transmission has been completed in smart card interface mode.
- Indicates that NACK has been detected if the SIMR2.IICINTM bit is 0 in simple I^2C mode.
- Indicates that the ninth-bit SSCL5 falling edge has been detected if the SIMR2.IICINTM bit is 1 in simple I^2C mode.

(4) Transmit end event output

- Indicates the completion of transmission.
- Indicates that the starting condition, resumption condition, or termination condition has been generated in simple I^2C mode.

40.14 Usage Notes

40.14.1 Setting the Module Stop Function

Module stop control register B (MSTPCRB) is used to stop and start SCI operations. With the value after a reset, SCI operations are stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

40.14.2 Break Detection and Processing

When a framing error is detected, a break can be detected by reading the RXDn pin value directly. In a break, the input from the RXDn pin becomes all 0s, and so the FER flag in the SSR register is set to 1 (framing error has occurred), and the PER flag in the SSR register may also be set to 1 (parity error has occurred). When the SEMR.RXDESEL bit is 0, the SCI continues the receive operation even after a break is received. Therefore, note that even if the FER flag is set to 0 (no framing error occurred), it will be set to 1 again. When the SEMR.RXDESEL bit is 1, the SCI sets the SSR.FER flag to 1 and stops receiving operation until a start bit of the next data frame is detected. If the SSR.FER flag is set to 0 at this time, the SSR.FER flag retains 0 during the break. When the RXDn pin becomes high and the break ends, detecting the beginning of the start bit at the first falling edge of the RXDn pin allows the SCI to start the receiving operation.

40.14.3 Mark State and Generating Breaks

When the SCR.TE bit is 0 (serial transmission is disabled), setting the I/O port function makes selection of the level and direction (input or output) of the TXDn pin possible. If this is done, the TXDn pin can be placed in the mark state to send a break at the time of data transmission. Until the SCR.TE bit is set to 1 (serial transmission is enabled), the I/O port function is used to set the TXDn pin to output high and set the pin mode to a general I/O port pin, and thus place the communication line in the mark state (state of having the value 1). On the other hand, to output a break at the time of data transmission, set the TXDn pin to output low and make the pin mode settings for a general I/O port pin. When the SCR.TE bit is set to 0, the transmitter is initialized regardless of the current state of transmission.

40.14.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)

Transmission cannot be started when a receive error flag (ORER) in the SSR register is set to 1, even if data is written to the TDR register. Be sure to set the receive error flags to 0 before starting transmission. Note also that the receive error flags cannot be set to 0 even if the RE bit in the SCR register is set to 0 (serial reception is disabled).

40.14.5 Writing Data to the TDR Register

Data can be written to registers TDR, TDRH, and TDRL. However, if new data is written to registers TDR, TDRH, and TDRL when transmit data is remaining in registers TDR, TDRH, and TDRL, the previous data in registers TDR, TDRH, and TDRL is lost because it has not been transferred to the TSR register yet. Be sure to write transmit data to registers TDR, TDRH, and TDRL in the TXI interrupt request handling routine.

40.14.6 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode)

When the external clock source is used as a synchronization clock, the following restrictions apply.

(1) Start of transmission

Update TDR by the CPU, DMAC, or DTC and wait for at least five PCLK cycles before allowing the transmit clock to be input (refer to Figure 40.74).

(2) Continuous transmission

(a) Write the next transmit data to TDR or TDRL before the falling edge of the transmit clock (bit 7) (refer to Figure 40.74).

(b) When updating TDR after bit 7 has started to transmit, update TDR while the synchronization clock is in the low-level period, and set the high-level width of the transmit clock (bit 7) to four PCLK cycles or longer (refer to Figure 40.74).

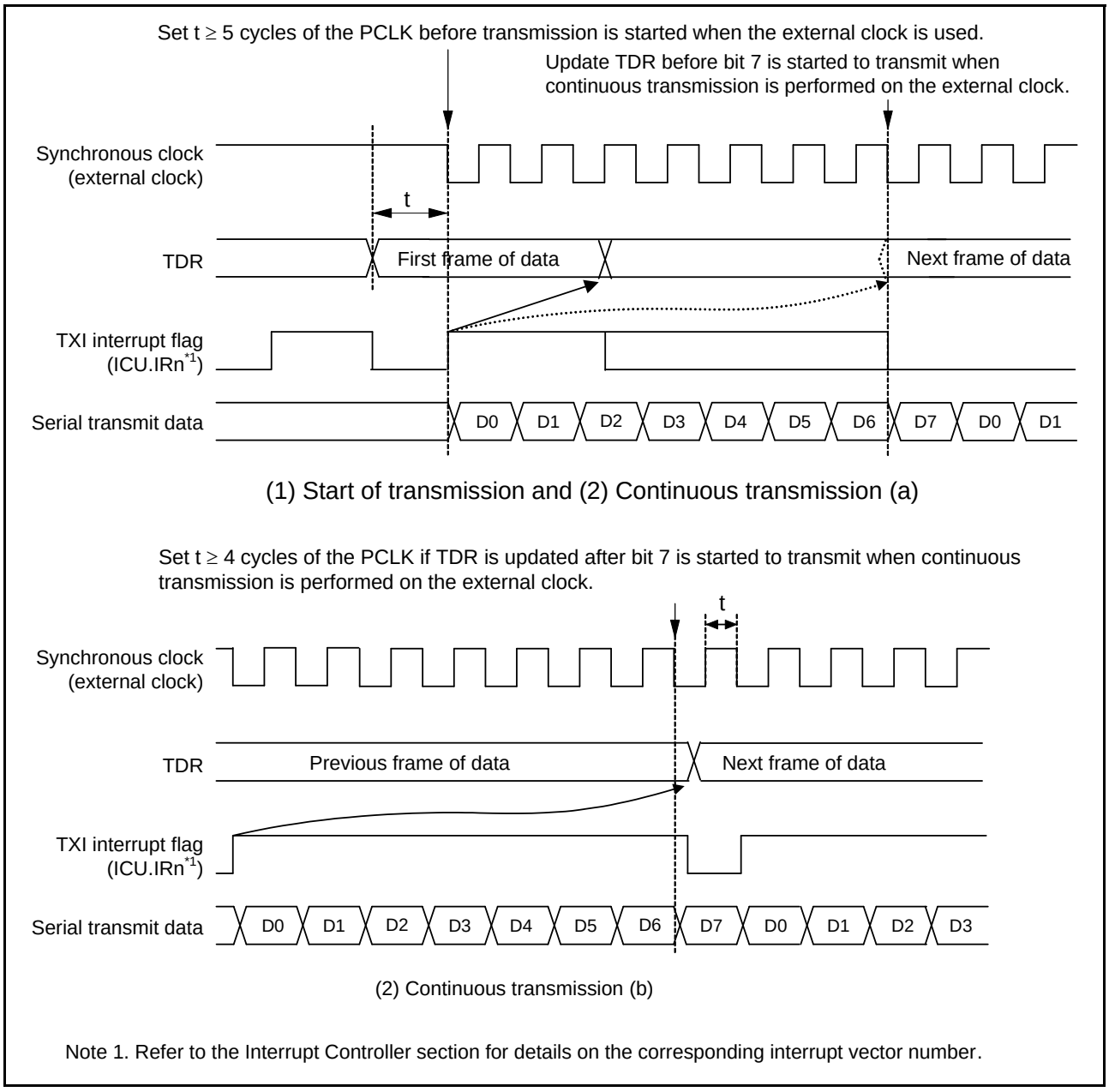


Figure 40.74 Restrictions on Use of External Clock in Clock Synchronous Transmission

40.14.7 Restrictions on Using DMAC or DTC

When using the DMAC or DTC to read RDR, RDRH, and RDRL, be sure to set the receive data full interrupt (RXI) as the activation source of the relevant SCI.

40.14.8 Notes on Starting Transfer

At the point where transfer starts when the interrupt status flag (IRn.IR bit) in the interrupt controller is 1, follow the procedure below to clear interrupt requests before permitting operations (by setting the SCR.TE or SCR.RE bit to 1). For details on the interrupt status flag, refer to section 15, Interrupt Controller (ICUA).

- Confirm that transfer has stopped (the setting of the SCR.TE or SCR.RE bits is 0).
- Set the corresponding interrupt enable bit (SCR.TIE or SCR.RIE) to 0.
- Read the corresponding interrupt enable bit (SCR.TIE or SCR.RIE bit) to check that it has become 0.
- Set the interrupt status flag (IRn.IR bit) in the interrupt controller to 0.

40.14.9 SCI Operations during Low Power Consumption State

(1) Transmission

When making settings for the module stopped state or in transitions to software standby, stop operations (by setting the TIE, TE, and TEIE bits in the SCR to 0) after switching the TXDn pin to the general I/O port pin function. Setting the TE bit to 0 resets the TSR register and the TEND bit in the SSR. Depending on the port settings, output pins may output the level before a transition to the low power consumption state is made after release from the module stopped state or software standby mode. When transitions to these states are made during transmission, the data being transmitted become indeterminate.

To transmit data in the same transmission mode after cancellation of the low power consumption state, set the TE bit to 1, read SSR, and write data to TDR sequentially to start data transmission. To transmit data with a different transmission mode, initialize the SCI first.

Figure 40.75 shows a sample flowchart for transition to software standby mode during transmission. Figure 40.76 and Figure 40.77 show the port pin states during transition to software standby mode.

Before specifying the module stop state or making a transition to software standby mode from the transmission mode using DTC/DMA transfer, stop the transmit operations (TE = 0). To start transmission after cancellation using the DTC/DMAC, set the TE and TIE bits to 1. The TXI interrupt flag is set to 1 and transmission starts using the DTC/DMAC.

(2) Reception

Before specifying the module stop state or making a transition to software standby mode, stop the receive operations (RE = 0 in the SCR register). If transition is made during data reception, the data being received will be invalid.

To receive data in the same reception mode after cancellation of the low power consumption state, set the RE bit to 1, and then start reception. To receive data in a different reception mode, initialize the SCI first.

Figure 40.78 shows a sample flowchart for transition to software standby mode during reception.

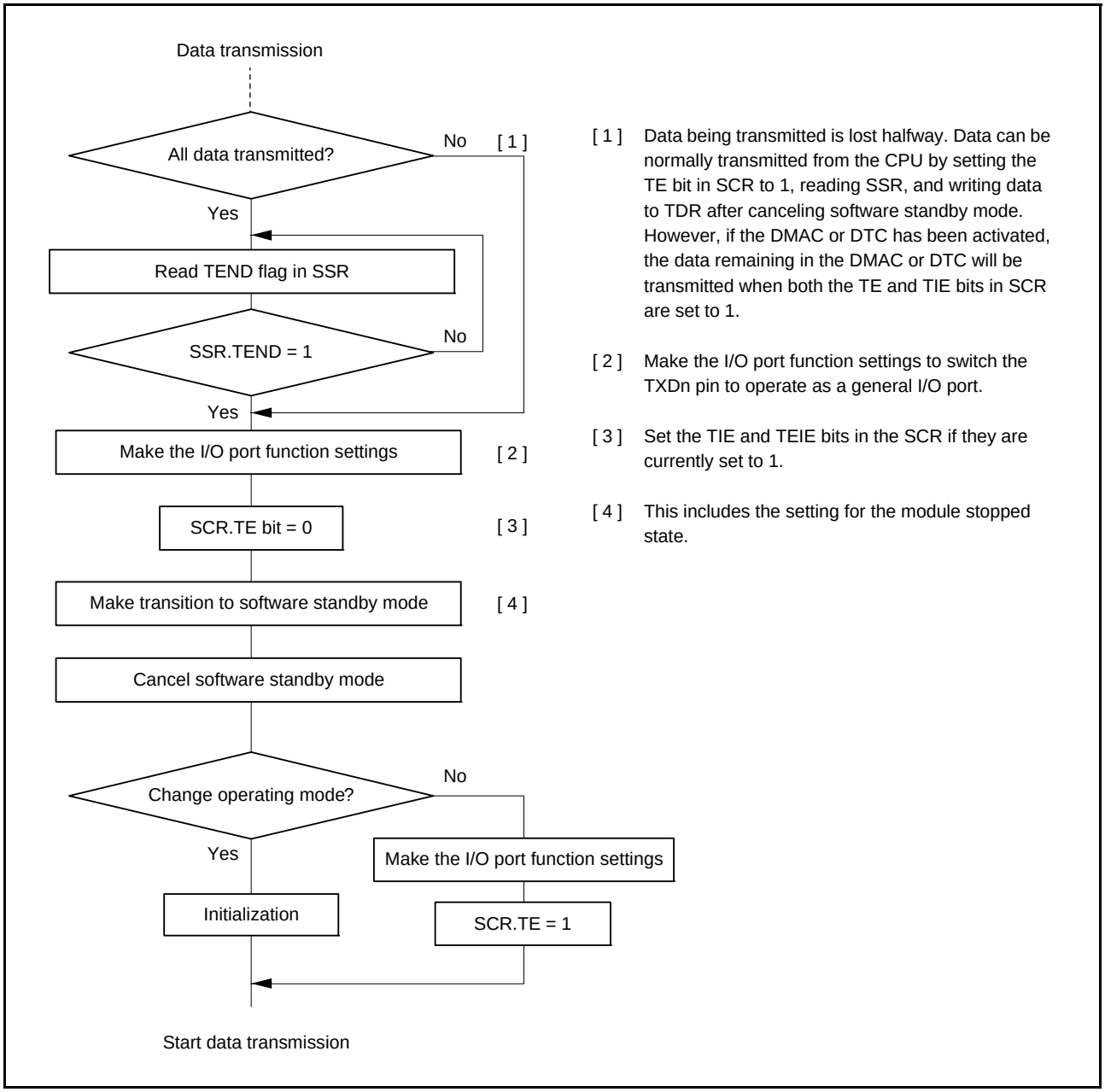


Figure 40.75 Example of Flowchart for Transition to Software Standby Mode during Transmission

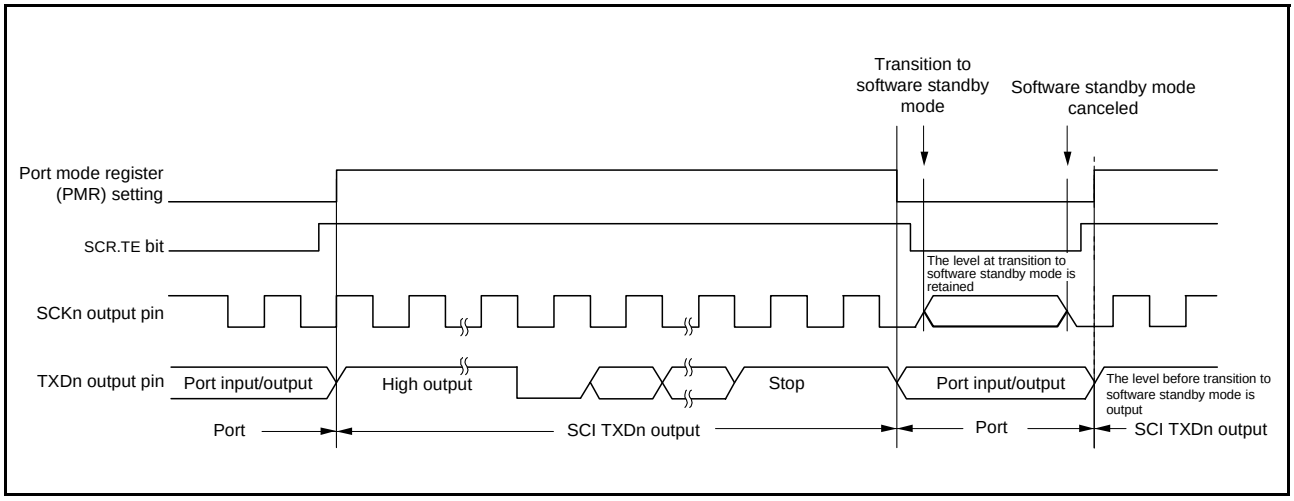


Figure 40.76 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

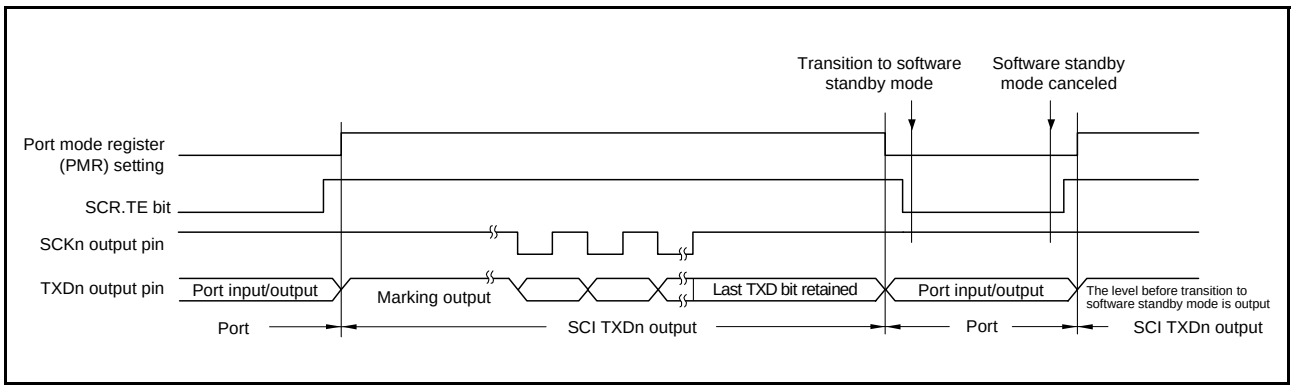


Figure 40.77 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

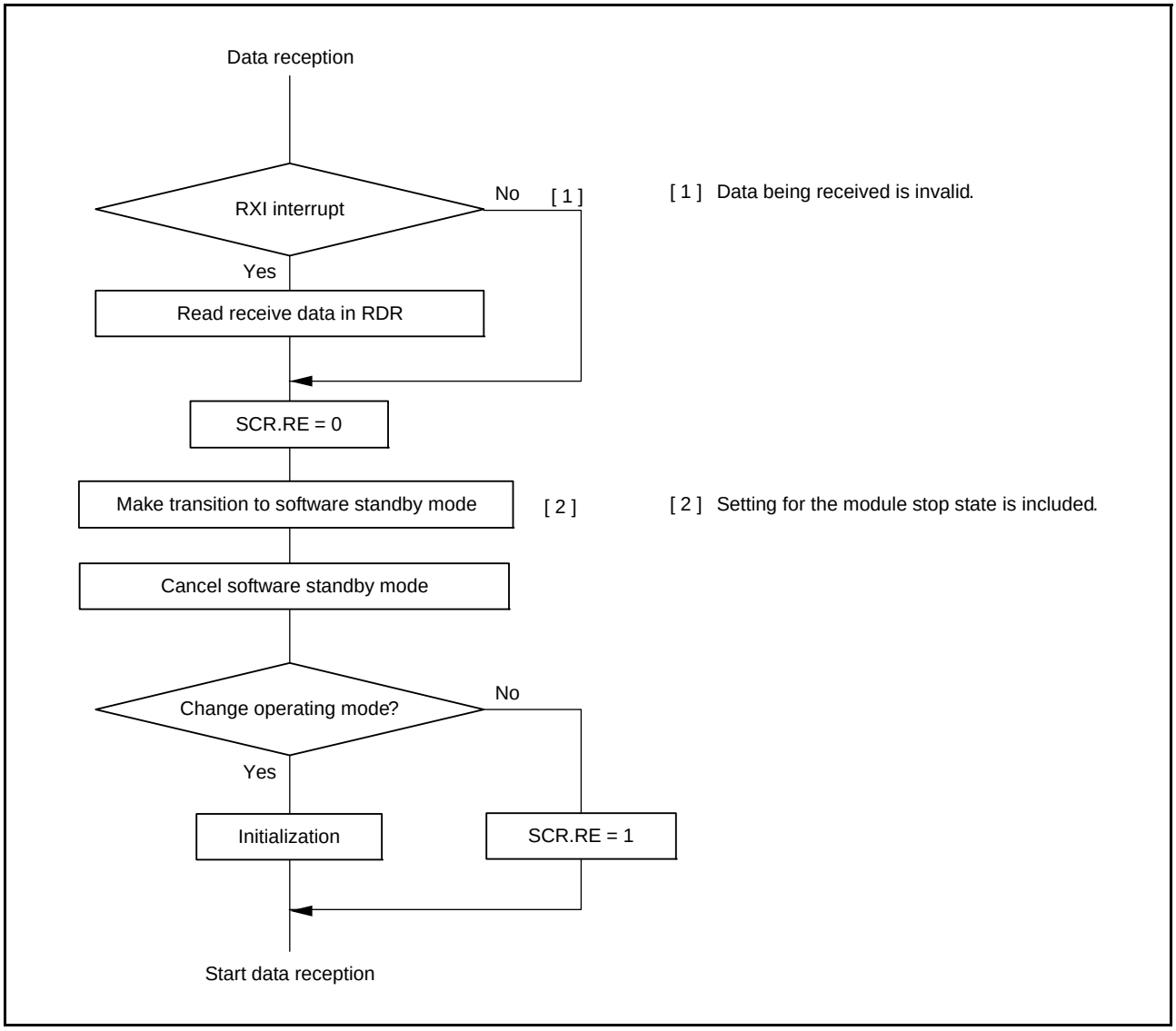


Figure 40.78 Example of Flowchart for Transition to Software Standby Mode during Reception

40.14.10 External Clock Input in Clock Synchronous Mode and Simple SPI Mode

In clock synchronous mode and simple SPI mode, the external clock SCKn must be input as follows:
High-pulse period, low-pulse period = 2 PCLK cycles or more, period = 6 PCLK cycles or more

40.14.11 Limitations on Simple SPI Mode

(1) Master Mode

- Use a resistor to pull up or pull down the clock line matching the initial settings for the transfer clock set by the SPMR.CKPH and CKPOL bits when the SPMR.SSE bit is 1.
  This prevents the clock line from being placed in the high-impedance state when the SCR.TE bit is set to 0 or unexpected edges from being generated on the clock line when the SCR.TE bit is changed from 0 to 1. When the SPMR.SSE bit is 0 in single master mode, pulling up or pulling down the clock line is not necessary because the clock line is not placed in the high-impedance state even when the SCR.TE bit is set to 0.
- In the case of the setting for clock delay (SPMR.CKPH bit is 1), the receive data full interrupt (RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 40.79. If the TE and RE bits in the SCR become 0 at this time before the final edge of the clock signal on the SCKn pin, the SCKn pin is placed in the high-impedance state, so the width of the last clock pulse of the transfer clock is shortened. Furthermore, an RXI interrupt may lead to the input signal on the SSn# pin of a connected slave going to the high level before the final edge of the clock signal on the SCKn pin, leading to incorrect operation of the slave.
- In a multi-master configuration, take care because the SCKn pin output becomes high-impedance while the input on the SSn# pin is at the low level if a mode fault error occurs as the current character is being transferred, stopping supply of the clock signal to the connected slave. Remake the settings for the connected slave to avoid misaligned bits when transfer is restarted.

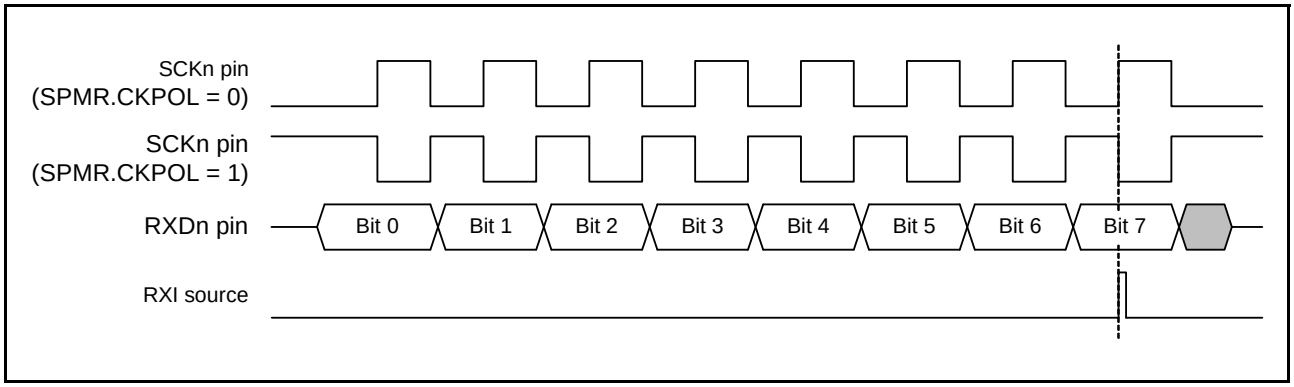


Figure 40.79 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)

(2) Slave Mode

- Secure at least five cycles of the PCLK from writing transmit data in the TDR register to start of the external clock input. Also secure at least five cycles of the PCLK from input of low level on the SSn# pin to start of the external clock input.
- Provide an external clock signal from the master the same as the transmit/receive data length.
- Control the input on the SSn# pin before the start and after the end of data transfer.
- When the level being input on the SSn# pin is to be changed from low to high while the current character is being transferred, set the TE and RE bits in the SCR to 0 and, after remaking the settings, restart transfer of the first byte.

40.14.12 Limitation 1 on Usage of the Extended Serial Mode Control Section

When the PCR.SHARPS bit is set to 1, output on the TXDX12/RXDX12 pin is only possible when the following conditions apply.

- The timer of the SCIh module is in Break Field low width output mode and the value of the TCR.TCST bit is 1 (when the TCST bit is set to 1, the high level continues to be output for up to one cycle of the clock source for counting by the timer counter before output of the low level)
- The value of the SCI12.SCR.TE bit is 1.

40.14.13 Limitation 2 on Usage of the Extended Serial Mode Control Section

An SCIg interrupt request is generated even if the extended serial mode is enabled. However, the SCIg interrupt should not be used during reception of a Start Frame because SCIh uses an SCIg interrupt request.
The two ways of dealing with this are described below. When a receive error is detected, clear the error flag of the SCIg and initialize the control section of the SCIh.

(1) Set the SCR.RIE bit of the SCIg to 0 to disable the output of interrupt requests. Check the error flags in the SSR register for SCIg on completion of the reception of a Start Frame, because an ERI interrupt is not generated if a receive error occurs. After reception of the Start Frame is completed, set the SCR.RIE bit of the SCIg to 1 by the time the first byte of the Information Frame is received.

(2) Set the SCR.RIE bit of the SCIg to 1 to disable RXI interrupts and enable ERI interrupts for ICU.
Clear the IRn.IR flag to enable the acceptance of RXI interrupts by ICU by the time the first byte of the Information Frame is received after the completion of Start Frame reception.

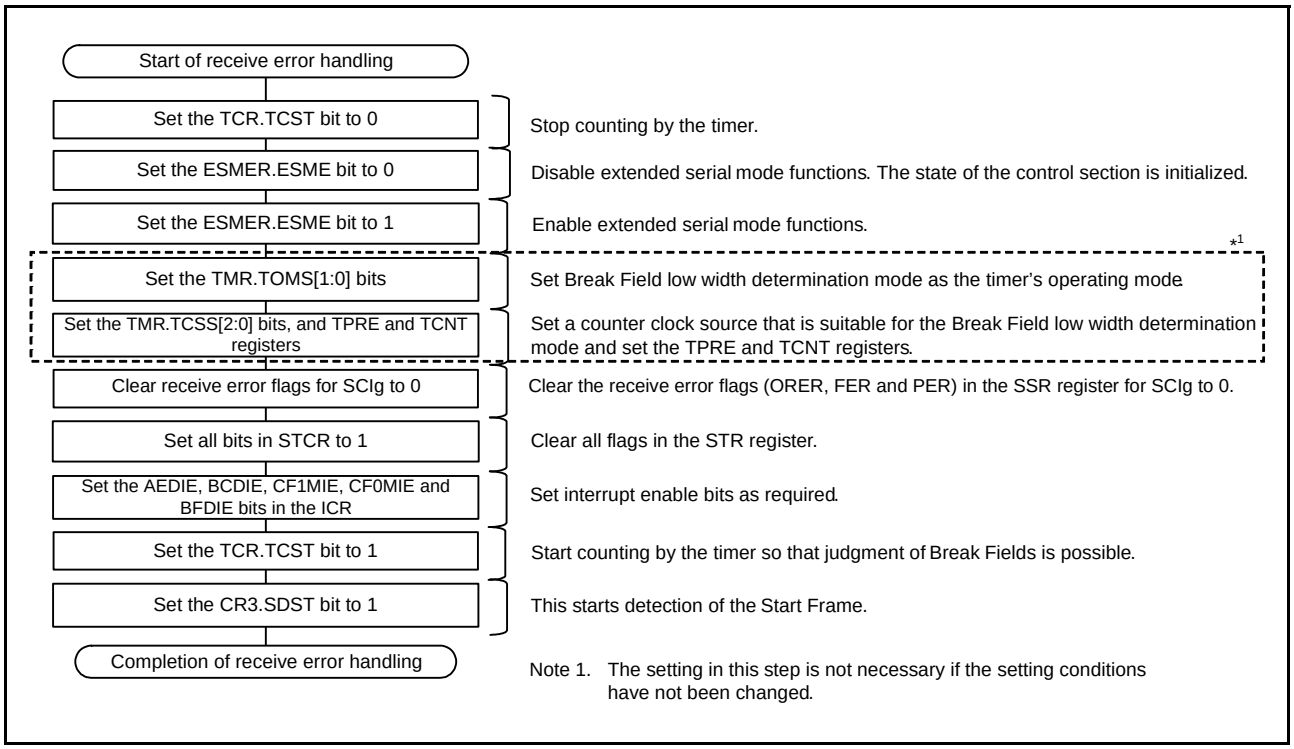


Figure 40.80 Example of Flowchart for Receive Error Handling (during Reception of the Start Frame)

40.14.14 Note on Transmit Enable Bit (TE Bit)

When setting the SCR.TE bit to 0 (serial transmission is disabled) while the pin function is “TXDn”, output of the pin becomes high impedance.

Prevent the TXDn line from becoming high impedance by any of the following ways:

(1) Connect a pull-up resistor to the TXDn line.

(2) Change the pin function to “general-purpose I/O port, output” before setting the SCR.TE bit to 0.
Set the SCR.TE bit to 1 before changing the pin function to “TXDn”.

40.14.15 Note on Stopping Reception When Using the RTS Function in Asynchronous Mode

One clock cycle of PCLK is required for the time from setting the SCR.RE bit to 0 to stopping the RTS signal generator in asynchronous mode.

When reading the RDR (or RDRL) register after setting the SCR.RE bit to 0, confirm that the RE bit has been set to 0 before reading the RDR (or RDRL) register to prevent these two processes from being performed consecutively.

41. FIFO Embedded Serial Communications Interface (SCIFA)

This MCU has four channels of serial communications interface with FIFO (SCIFA) that support both asynchronous and clock synchronous serial communication. The SCIFA has 16-stage transmit FIFO buffers and 16-stage receive FIFO buffers which allows the MCU to perform continuous high-speed communication.

41.1 Overview

Table 41.1 lists the SCIFA Specifications.

Table 41.1 SCIFA Specifications

Item		Description
Channels		SCIFA8, SCIFA9, SCIFA10, and SCIFA11
Communication methods		Asynchronous and clock synchronous
Full duplex communication		Transmitter: Continuous data transmission possible using 16-stage FIFO buffering Receiver: Continuous data reception possible using 16-stage FIFO buffering
Bit order		LSB first or MSB first
Interrupt sources		TEIF : Transmit end
		TXIF : Transmit FIFO data empty
		RXIF : Receive FIFO data full
		DRIF : Receive data ready (only valid in asynchronous communication mode)
		ERIF : Receive error
		BRIF : Break detect or overrun error
Asynchronous mode	Character length	7 or 8 bits
	Stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or none
	Receive error detection	A parity error, overrun error, or framing error can be detected as a receive error.
	Hardware flow control	Controls data transmission and reception using the CTS# and RTS# pins.
	Break signal detection	Break signal detection by hardware
	Clock source	Internal clock or external clock selectable
	Noise cancellation	Incorporates a digital noise filter in the RXD pin input path.
Clock synchronous mode	Character length	8 bits
	Receive error detection	An overrun error can be detected as a receive error.
	Clock source	Internal clock or external clock selectable
Bit rate modulation function		Errors can be reduced using the on-chip baud rate generator's output correction

Figure 41.1 shows the SCIFA Block Diagram.

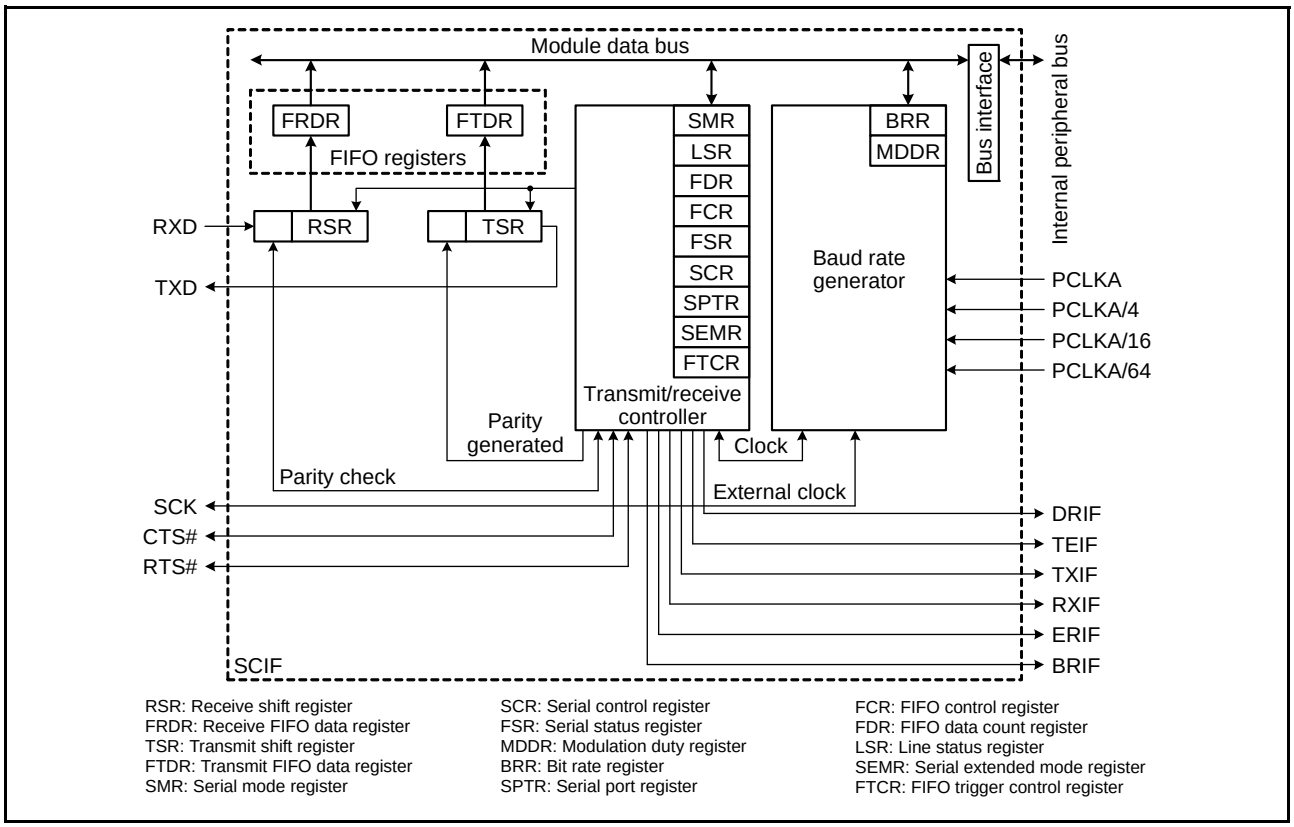


Figure 41.1 SCIFA Block Diagram

Table 41.2 shows the SCIFA I/O Pins.

Table 41.2 SCIFA I/O Pins

Channel	Pin Symbol *1	I/O	Function
SCIFA8	SCK8	I/O	Transmit/receive clock I/O, general output
	RXD8	Input	Receive data
	TXD8	Output	Transmit data
	CTS8#	I/O	Input for hardware flow control (clear to send), general output
	RTS8#	Output	Output for hardware flow control (request to send), general output
SCIFA9	SCK9	I/O	Transmit/receive clock I/O, general output
	RXD9	Input	Receive data
	TXD9	Output	Transmit data
	CTS9#	I/O	Input for hardware flow control (clear to send), general output
	RTS9#	Output	Output for hardware flow control (request to send), general output
SCIFA10	SCK10	I/O	Transmit/receive clock I/O, general output
	RXD10	Input	Receive data
	TXD10	Output	Transmit data
	CTS10#	I/O	Input for hardware flow control (clear to send), general output
	RTS10#	Output	Output for hardware flow control (request to send), general output
SCIFA11	SCK11	I/O	Transmit/receive clock I/O, general output
	RXD11	Input	Receive data
	TXD11	Output	Transmit data
	CTS11#	I/O	Input for hardware flow control (clear to send), general output
	RTS11#	Output	Output for hardware flow control (request to send), general output

Note 1. In this section, the channel number is omitted and pin symbols are abbreviated to SCK, RXD, TXD, CTS#, and RTS#.

41.2 Register Descriptions

41.2.1 Receive Shift Register (RSR)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Symbol								
Value after reset	0	0	0	0	0	0	0	0

The RSR register is a shift register used for receiving serial data. The RSR register converts serial data input from the RXD pin to parallel data. After receiving 1 byte of serial data, the SCIFA transfers the received data to the FRDR register. The CPU cannot directly access the RSR register.

41.2.2 Receive FIFO Data Register (FRDR)

Addresses: SCIFA8.FRDR 000D 000Ah, SCIFA9.FRDR 000D 002Ah, SCIFA10.FRDR 000D 004Ah, SCIFA11.FRDR 000D 006Ah

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Symbol								
Value after reset	0	0	0	0	0	0	0	0

The FRDR register is a 16-stage FIFO buffer used for storing received serial data. After 1 byte of serial data is received, the SCIFA transfers the received data from the RSR register into the FRDR register. Data can continuously be received until the 16-stage FIFO buffer is full. If the FRDR register is read when there is no received data in the FRDR register, the read value is undefined. When FRDR register is full of received data, subsequent received serial data is lost. The FRDR register is a read-only register.

41.2.3 Transmit Shift Register (TSR)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Symbol								
Value after reset	1	1	1	1	1	1	1	1

The TSR register is a shift register used for transmitting serial data. The SCIFA temporarily transfers transmit data from the FTDR register to the TSR register, and then transmits 1 bit of serial data at a time from the TXD pin. After transmitting 1 byte of serial data, the SCIFA transfers the next transmit data from the FTDR register to the TSR register. The CPU cannot directly access the TSR register.

41.2.4 Transmit FIFO Data Register (FTDR)

Addresses: SCIFA8.FTDR 000D 0006h, SCIFA9.FTDR 000D 0026h, SCIFA10.FTDR 000D 0046h, SCIFA11.FTDR 000D 0066h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Symbol								
Value after reset	1	1	1	1	1	1	1	1

The FTDR register is a 16-stage FIFO buffer used for storing data for serial transmission. When the TSR register becomes empty, transmit data written to the FTDR register is transferred to the TSR register by the SCIFA. Data can be continuously transmitted until the 16-stage FIFO buffer is empty. When the FTDR register is full of transmit data, no more data can be written. Even when attempting to write data, the data written is ignored. The FTDR register is a write-only register.

41.2.5 Serial Mode Register (SMR)

Addresses: SCIFA8.SMR 000D 0000h, SCIFA9.SMR 000D 0020h, SCIFA10.SMR 000D 0040h, SCIFA11.SMR 000D 0060h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	—	—	—	—	—	—	—	—	CM	CHR	PE	PM	STOP	—	CKS[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select Bits	b1 b0 0 0: PCLKA 0 1: PCLKA/4 1 0: PCLKA/16 1 1: PCLKA/64	R/W
b2	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b3	STOP	Stop Bit Length Select Bit	0: 1 stop bit 1: 2 stop bits	R/W
b4	PM	Parity Type Select Bit	0: Even parity 1: Odd parity	R/W
b5	PE	Parity Control Bit	0: Parity bit is neither added nor checked 1: Parity bit is added and checked	R/W
b6	CHR	Character Length Select Bit	0: 8 bits 1: 7 bits *1	R/W
b7	CM	Communication Mode Select Bit	0: Asynchronous mode 1: Clock synchronous mode	R/W
b15 to b8	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. When the CHR bit is 1, the MSB (b7) in the FTDR register is not transmitted.

The SMR register designates the serial communication data format, and designates the clock source for the baud rate generator.

CKS[1:0] Bits (Clock Select Bits)

These bits select the clock source for the baud rate generator. Refer to section 41.2.8, Bit Rate Register (BRR) for relation between the clock source, bit rate register settings, and bit rates.

STOP Bit (Stop Bit Length Select Bit)

While in asynchronous mode, this bit selects the stop bit length from 1 bit or 2 bits. A stop bit is only added when transmitting in asynchronous mode. The bit setting value is ignored in clock synchronous mode because no stop bits are added. When receiving data, only the first bit of the stop bit is checked, regardless of the stop bit setting. If the second bit of the stop bit is 1, it is treated as a stop bit, but if the second bit of the stop bit is 0, it is treated as the start bit of the next transmitted character.

PM Bit (Parity Type Select Bit)

In asynchronous mode, this bit selects how many parity bits are added during transmission, performs parity check when receiving data, and selects even or odd parity. This bit is only valid when the PE bit is 1 while in asynchronous mode. This bit is invalid when the PE bit is 0 while in clock synchronous mode.

PE Bit (Parity Control Bit)

In asynchronous mode, this bit selects whether to add a parity bit during data transmission, and whether to check the parity on data reception. Regardless of the PE bit setting, a parity bit is neither added nor checked in clock synchronous mode.

CHR Bit (Character Length Select Bit)

In asynchronous mode, this bit selects the valid bit length of transfer data from 7 bits or 8 bits. Regardless of the CHR bit setting, the data length is always 8 bits in clock synchronous mode.

CM Bit (Communication Mode Select Bit)

This bit selects the SCIFA operating mode from asynchronous mode or clock synchronous mode.

41.2.6 Serial Control Register (SCR)

Addresses SCIFA8.SCR 000D 0004h, SCIFA9.SCR 000D 0024h, SCIFA10.SCR 000D 0044h, SCIFA11.SCR 000D 0064h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	—	—	—	—	—	—	—	—	TIE	RIE	TE	RE	REIE	TEIE	CKE[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable Bits	In asynchronous mode b1 b0 0 0: The baud rate generator output is the clock source, and the SCK pin status is determined by the SPTR.SCKIO and SCKDT bit settings. 0 1: The baud rate generator output is the clock source. When the SEMR.ABCS0 bit is 0, the SCK pin outputs a clock 16 times faster than the bit rate; when the SEMR.ABCS0 bit is 1, the SCK pin outputs a clock eight times faster than the bit rate. 1 0: The external clock is the clock source. When the SEMR.ABCS0 bit is 0, an external clock 16 times faster than the bit rate is input to the SCK pin; when the SEMR.ABCS0 bit is 1, an external clock eight times faster than the bit rate is input to the SCK pin. 1 1: Do not set this value. In clock synchronous mode b1 b0 0 0: The baud rate generator output is the clock source, and the synchronous clock is output by the SCK pin. 0 1: The baud rate generator output is the clock source, and the synchronous clock is output by the SCK pin. 1 0: The external clock is the clock source, and the synchronous clock is input to the SCK pin. 1 1: Do not set this value.	R/W
b2	TEIE	Transmit End Interrupt Request Enable Bit	0: TEIF interrupt request disabled 1: TEIF interrupt request enabled	R/W
b3	REIE	Receive Error Interrupt Request Enable Bit	0: ERIF interrupt request and BRIF interrupt request disabled 1: ERIF interrupt request and BRIF interrupt request enabled	R/W
b4	RE	Reception Enable Bit	0: Serial reception disabled 1: Serial reception enabled	R/W
b5	TE	Transmission Enable Bit	0: Serial transmission disabled 1: Serial transmission enabled	R/W
b6	RIE	Receive Interrupt Request Enable Bit	0: RXIF interrupt request, DRIF interrupt request, ERIF interrupt request, and BRIF interrupt request disabled 1: RXIF interrupt request, DRIF interrupt request, ERIF interrupt request, and BRIF interrupt request enabled	R/W
b7	TIE	Transmit Interrupt Request Enable Bit	0: TXIF interrupt disabled 1: TXIF interrupt enabled	R/W
b15 to b8	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

The SCR register designates the transmission and reception operations, enables and disables interrupt requests, and designates the clock source for data transfer.

CKE[1:0] Bits (Clock Enable Bits)

These bits select the clock source for the SCIFA. These bits also select whether the clock is output from the SCK pin or input to the SCK pin. To output a synchronous clock in clock synchronous mode, set the SMR.CM bit to 1 and then set the CKE[1:0] bits.

TEIE Bit (Transmit End Interrupt Request Enable Bit)

Set this bit to enable and disable generation of the TEIF interrupt request. To cancel the TEIF interrupt request, confirm that the FSR.TEND flag is 1, and then either set the TEND flag to 0 or set the TEIE bit to 0.

REIE Bit (Receive Error Interrupt Request Enable Bit)

Set this bit to enable and disable generation of the ERIF interrupt request and BRIF interrupt request. To cancel the ERIF and BRIF interrupt requests, confirm that the FSR.ER, BRK, or ORER flag is 1, set the ER, BRK, or ORER flag to 0, and then set bits RIE and REIE to 0.

RE Bit (Reception Enable Bit)

Set this bit to enable and disable serial reception. When the RE bit is 1, data reception starts when either a start bit is detected in asynchronous mode, or a synchronous clock is detected in clock synchronous mode. Before setting the RE bit to 1, select the communication format in the SMR register, and in the FCR register, set the receive FIFO and perform a reset.

Note: Flags FSR.DR, ER, BRK, RDF, FER, PER, and the LSR.ORER flag do not become 0 even when the RE bit is set to 0.

TE Bit (Transmission Enable Bit)

Set this bit to enable and disable serial transmission. When the TE bit is set to 1 and transmit data is written to the FTDR register, serial transmission starts. Before setting the TE bit to 1, select the communication format in the SMR register, and in the FCR register, set the transmit FIFO and perform a reset.

RIE Bit (Receive Interrupt Request Enable Bit)

This bit enables and disables generation of the RXIF interrupt request, DRIF interrupt request, ERIF interrupt request, and BRIF interrupt request. To cancel an RXIF interrupt request, confirm that the FSR.RDF flag is 1 before setting it to 0, or set the RIE bit to 0. To cancel the DRIF interrupt request, confirm that the FSR.DR flag is 1 before setting it to 0, or set the RIE bit to 0. To cancel the ERIF interrupt request and BRIF interrupt request, set bits RIE and REIE to 0.

TIE Bit (Transmit Interrupt Request Enable Bit)

This bit enables and disables generation of the TXIF interrupt request. To cancel the TXIF interrupt request, write transmit data that is equal to or greater than the transmit FIFO threshold value to the FTDR register, confirm that the FSR.TDFE flag is 1 before setting it to 0, or set the TIE bit to 0.

41.2.7 Serial Status Register (FSR)

Addresses SCIFA8.FSR 000D 0008h, SCIFA9.FSR 000D 0028h, SCIFA10.FSR 000D 0048h, SCIFA11.FSR 000D 0068h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	—	—	—	—	—	—	—	—	ER	TEND	TDFE	BRK	FER	PER	RDF	DR
Value after reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DR	Receive Data Ready Flag *1	0: All receive data is read from the FRDR register (receive FIFO is empty) during data reception or after data reception is completed normally 1: After data reception is completed normally, the next receive data is not received for a given period	R(/W) *2
b1	RDF	Receive FIFO Data Full Flag	0: The amount of receive data stored in the FRDR register is less than the receive FIFO threshold 1: The amount of receive data stored in the FRDR register is the receive FIFO threshold or more	R(/W) *2
b2	PER	Parity Error Flag *1, *3	0: The start data in the FRDR register is not in the parity error state 1: The start data in the FRDR register is in the parity error state	R
b3	FER	Framing Error Flag *1, *3	0: The start data in the FRDR register is not in the framing error state 1: The start data in the FRDR register is in the framing error state	R
b4	BRK	Break Signal Detection Flag	0: Break signal not detected 1: Break signal detected *4	R(/W) *2
b5	TDFE	Transmit FIFO Data Empty Flag	0: The amount of transmit data written to the FTDR register is more than the transmit FIFO threshold 1: The amount of transmit data written to the FTDR register is the transmit FIFO threshold or less	R(/W) *2
b6	TEND	Transmission Complete Flag	0: Waiting for data to transmit or data is being transmitted 1: Transmission complete	R(/W) *2
b7	ER	Reception Error Flag *1	0: Data is being received or reception was completed normally 1: A framing error or parity error occurred during data reception	R(/W) *2
b15 to b8	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. This bit is valid in asynchronous mode.
Note 2. This flag can only be written to when it is being cleared. To clear this flag, confirm that the flag is 1 and then set it to 0.
Note 3. These flags are updated each time data is read from the FRDR register. When using the DTC or DMAC to read the received data, these flags cannot be used to detect errors.
Note 4. When a break signal is detected, storing receive data (00h) to the FRDR register is stopped. When the break signal has ended and the RXD signal is “marked”, storing receive data to the FRDR register resumes.

The FSR register indicates the transmission and reception status. The ER, TEND, TDFE, BRK, RDF, and DR flags cannot be written. To set these flags to 0, confirm that they are 1, and then write 0.
Because the PER and FER flags are updated each time data is read from the FRDR register, there is no way to detect errors after the data has been read. When using the DTC or DMAC to read the received data, use flags PER[3:0] and FER[3:0] in the LSR register. If any of the values of these flags are decremented after the DTC or DMA transfer, this means that the corresponding error has occurred in the read data.

DR Flag (Receive Data Ready Flag)

After receiving data in asynchronous mode, when the amount of data stored in the FRDR register is less than the receive FIFO threshold value, if the next data is not received even after 15 elementary time units (ETUs) elapse from the time the last stop bit is detected, then this flag becomes 1. However, this flag does not become 1 in clock synchronous mode.

— This flag becomes 1 under the following condition:

- When the data stored to the FRDR register is less than the receive FIFO threshold value, the next data is not received even after 15 ETUs *1 elapse from the time the last stop bit is detected.

— This flag becomes 0 under any of the following conditions:

- After confirming that the DR flag is 1, it is set to 0.
- All receive data in the FRDR register is read.

Note 1. 15 ETUs is equivalent to 1.5 frames of an 8-bit, 1 stop bit format.

RDF Flag (Receive FIFO Data Full Flag)

When receive data is transferred to the FRDR register, this flag indicates that the amount of data stored in the FRDR register is the receive FIFO threshold value or more.

— This flag becomes 1 under the following condition:

- Receive data stored in the FRDR register is the receive FIFO threshold value or more. *1

— This flag becomes 0 under any of the following conditions:

- After confirming that the RDF flag is 1, it is set to 0.
- DTC or DMA transfer is used to read the receive data from the FRDR register (do not set the RDF flag to 0 during DTC or DMA transfer).

Note 1. Since the FRDR register is a 16-byte FIFO register, the maximum amount of data that can be read when RDF flag is 1 is indicated by the FDR.R[4:0] flags. After reading all the data in the FRDR register, continuing a read access results in an undefined value.

PER Flag (Parity Error Flag)

When in asynchronous mode, this flag indicates whether there is a parity error in the next data to be read from the FRDR register.

— This flag becomes 1 under the following condition:

- There is a parity error in the next data to be read from the FRDR register.

— This flag becomes 0 under the following condition:

- There is no parity error in the next data to be read from the FRDR register.

FER Flag (Framing Error Flag)

When in asynchronous mode, this flag indicates whether there is a framing error in the next data to be read from the FRDR register.

— This flag becomes 1 under the following condition:

- There is a framing error in the next data to be read from the FRDR register.

— This flag becomes 0 under the following condition:

- There is no framing error in the next data to be read from the FRDR register.

BRK Flag (Break Signal Detection Flag)

This flag indicates that a break signal was detected while receiving data.

— This flag becomes 1 under the following condition:

- Data that includes a framing error is received, followed by at least one frame length of a “space” (low) being received.

— This flag becomes 0 under the following condition:

- After confirming the BRK flag is 1, it is set to 0.

TDFE Flag (Transmit FIFO Data Empty Flag)

This flag indicates that data has been transferred from the FTDR register to the TSR register, it indicates the amount of data in FTDR has become the transmit FIFO threshold value or lower, and it indicates that transmit data can be written to the FTDR register.

— This flag becomes 1 under any of the following conditions:

- The SCR.TE bit is 0.
- The amount of transmit data written to the FTDR register is less than or equal to the transmit FIFO threshold value.

— This flag becomes 0 under any of the following conditions:

- After confirming that the TDFE flag is 1, it is set to 0.
- DTC or DMA transfer *1 is used to write transmit data to the FTDR register *2

Note 1. The last block transferred during block transfer.
Note 2. Do not clear the TDFE flag during DTC or DMA transfer.
Note 3. Because the FTDR register is a 16-byte FIFO register, when the TDFE flag is 1, the maximum amount of data that can be written to the FTDR register is (16 - FDR.T[4:0]). All other data written to the FTDR register above that value is ignored.

TEND Flag (Transmission Complete Flag)

This flag indicates that the FTDR register does not contain valid data when transmitting the last bit of a serial character, so the transmission is halted.

— This flag becomes 1 under the following condition:

- When transmitting one frame of data, there is no transmit data in the FTDR register when the last bit is transmitted.

— This flag becomes 0 under any of the following conditions:

- Transmit data is written to the FTDR register.
- After confirming that the TEND flag is 1, it is set to 0.

ER Flag (Reception Error Flag)

This flag indicates that a reception error (framing error or parity error) occurred in the frame that was just received. *1

— This flag becomes 1 under any of the following conditions:

- The stop bit in the received frame is 0. *2
- During data reception, the SMR.PM bit setting did not match the combined number of 1's in the receive data and the parity bit.

— This flag becomes 0 under the following condition:

- After confirming that the ER flag is 1, it is set to 0.

Note 1. The ER flag value is not affected by setting the SCR.RE bit to 0. Even if a receive error occurs, the receive data is transferred to the FRDR register, and reception continues. Read the FSR.FER and PER flags to check if the data read from the FRDR register includes a receive error.
Note 2. When using 2 stop bits, only the first stop bit is checked; the second stop bit is not checked.

41.2.8 Bit Rate Register (BRR)

Addresses: SCIFA8.BRR 000D 0002h, SCIFA9.BRR 000D 0022h, SCIFA10.BRR 000D 0042h, SCIFA11.BRR 000D 0062h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Symbol								
Value after reset	1	1	1	1	1	1	1	1

The BRR register is used to adjust the bit rate. The BRR register is located in the same address as the MDDR register. The BRR register is can be accessed when the SEMR.MDDRS bit is 0. Rewrite this register when bits SCR.TE and RE are both 0.

The BRR register setting value can be calculated using the formulas in Table 41.3.

Table 41.3 Formulas for Calculating the BRR Register Setting Value

Mode		Operate the SCIFA at 16 Times the Bit Rate (SEMR.ABCS0 bit is 0)	Operate the SCIFA at 8 Times the Bit Rate (SEMR.ABCS0 bit is 1)
Asynchronous mode	Baud rate generator is in normal mode (SEMR.BGDM bit is 0)	$N = \frac{f_{(PCLKA)} \times 10^6}{2^{2n} \times 2 \times 16 \times B} - 1$	$N = \frac{f_{(PCLKA)} \times 10^6}{2^{2n} \times 2 \times 8 \times B} - 1$
	Baud rate generator is in high-speed mode (SEMR.BGDM bit is 1)	$N = \frac{f_{(PCLKA)} \times 10^6}{2^{2n} \times 16 \times B} - 1$	$N = \frac{f_{(PCLKA)} \times 10^6}{2^{2n} \times 8 \times B} - 1$
Clock synchronous mode		$N = \frac{f_{(PCLKA)} \times 10^6}{2^{2n} \times 4 \times B} - 1$	

N: BRR register setting value ($0 \le N \le 255$)
$f_{(PCLKA)}$: Frequency of the peripheral module clock [MHz]
n: SMR.CKS[1:0] bit setting value (n = 0 to 3) (see Table 41.4)
B: Bit rate [bps] (set a value that satisfies the Electrical Characteristics)

Table 41.4 SMR.CKS[1:0] Bit Settings

SMR.CKS[1:0] Bit Setting Value	n	Clock Input to the Baud Rate Generator
00b	0	PCLKA
01b	1	PCLKA/4
10b	2	PCLKA/16
11b	3	PCLKA/64

The bit rate error [%] while in asynchronous mode can be calculated using the formulas in Table 41.5.

Table 41.5 Bit Rate Error While in Asynchronous Mode

Mode	Operate the SCIFA at 16 Times the Bit Rate (SEMR.ABCS0 bit is 0)	Operate the SCIFA at 8 Times the Bit Rate (SEMR.ABCS0 bit is 1)
Baud rate generator is in normal mode (SEMR.BGDM bit is 0)	$\left(\frac{f_{(PCLKA)} \times 10^6}{2^{2n} \times 2 \times 16 \times B \times (N+1)} - 1\right) \times 100$	$\left(\frac{f_{(PCLKA)} \times 10^6}{2^{2n} \times 2 \times 8 \times B \times (N+1)} - 1\right) \times 100$
Baud rate generator is in high-speed mode (SEMR.BGDM bit is 1)	$\left(\frac{f_{(PCLKA)} \times 10^6}{2^{2n} \times 16 \times B \times (N+1)} - 1\right) \times 100$	$\left(\frac{f_{(PCLKA)} \times 10^6}{2^{2n} \times 8 \times B \times (N+1)} - 1\right) \times 100$

Table 41.6 and Table 41.7 list examples of BRR register setting in asynchronous mode, and Table 41.8 lists examples of BRR register setting in clock synchronous mode.

Table 41.6 Bit Rates and BRR Register Settings in Asynchronous Mode *1 (1/2)

Bit Rate [bps]	PCLKA Frequency [MHz]																				
	10			12			14			16			18			20			22		
	n	N	Error [%]	n	N	Error [%]	n	N	Error [%]	n	N	Error [%]	n	N	Error [%]	n	N	Error [%]	n	N	Error [%]
110	2	177	–0.25	2	212	0.03	2	248	–0.17	3	70	0.03	3	79	–0.12	3	88	–0.25	3	97	–0.35
150	2	129	0.16	2	155	0.16	2	181	0.16	2	207	0.16	2	233	0.16	3	64	0.16	3	71	–0.54
300	2	64	0.16	2	77	0.16	2	90	0.16	2	103	0.16	2	116	0.16	2	129	0.16	2	142	0.16
600	1	129	0.16	1	155	0.16	1	181	0.16	1	207	0.16	1	233	0.16	2	61	0.16	2	71	–0.54
1200	1	64	0.16	1	77	0.16	1	90	0.16	1	103	0.16	1	116	0.16	1	129	0.16	1	142	0.16
2400	0	129	0.16	0	155	0.16	0	181	0.16	0	207	0.16	0	233	0.16	1	64	0.16	1	71	–0.54
4800	0	64	0.16	0	77	0.16	0	90	0.16	0	103	0.16	0	116	0.16	0	129	0.16	0	142	0.16
9600	0	32	–1.36	0	38	0.16	0	45	–0.93	0	51	0.16	0	58	–0.69	0	64	0.16	0	71	–0.54
14400	0	21	–1.36	0	25	0.16	0	29	127	0	34	–0.79	0	38	0.16	0	42	0.94	0	47	–0.54
19200	0	15	1.73	0	19	–2.34	0	22	–0.93	0	25	0.16	0	28	1.02	0	32	–1.36	0	35	–0.54
28800	0	10	–1.36	0	12	0.16	0	14	1.27	0	16	2.12	0	19	–2.34	0	21	–1.36	0	23	–0.54
31250	0	9	0.00	0	11	0.00	0	13	0.00	0	15	0.00	0	17	0.00	0	19	0.00	0	21	0.00
38400	0	7	1.73	0	9	–2.34	0	10	3.57	0	12	0.16	0	14	–2.34	0	15	1.73	0	17	–0.54
115200	0	2	–9.58	0	2	8.51	0	3	–5.06	0	3	8.51	0	4	–2.34	0	4	8.51	0	5	–0.54
500000	0	0 *2	–37.5	0	0 *2	–25.0	0	0 *2	–12.5	0	0 *2	0.00	0	0 *2	12.5	0	0 *2	25.0	0	0 *2	37.5

Bit Rate [bps]	PCLKA Frequency [MHz]																				
	24			26			28			30			32			34			36		
	n	N	Error [%]	n	N	Error [%]	n	N	Error [%]	n	N	Error [%]	n	N	Error [%]	n	N	Error (%)	n	N	Error (%)
110	3	106	–0.44	3	114	0.36	3	123	0.23	3	132	0.13	3	141	0.03	3	150	–0.05	3	159	–0.12
150	3	77	0.16	3	84	–0.43	3	90	0.16	3	97	–0.35	3	103	0.16	3	110	–0.29	3	116	0.16
300	2	155	0.16	2	168	0.16	2	181	0.16	2	194	0.16	2	207	0.16	2	220	0.16	2	233	0.16
600	2	77	0.16	2	84	–0.43	2	90	0.16	2	97	–0.35	2	103	0.16	2	110	–0.29	2	116	0.16
1200	1	155	0.16	1	168	0.16	1	181	0.16	1	194	0.16	1	207	0.16	1	220	0.16	1	233	0.16
2400	1	77	0.16	1	84	–0.43	1	90	0.16	1	97	–0.35	1	103	0.16	1	110	–0.29	1	116	0.16
4800	0	155	0.16	0	168	0.16	0	181	0.16	0	194	0.16	0	207	0.16	0	220	0.16	0	233	0.16
9600	0	77	0.16	0	84	–0.43	0	90	0.16	0	97	–0.35	0	103	0.16	0	110	–0.29	0	116	0.16
14400	0	51	0.16	0	55	0.76	0	60	–0.39	0	64	0.16	0	68	0.64	0	73	–0.29	0	77	0.16
19200	0	38	0.16	0	41	0.76	0	45	–0.93	0	48	–0.35	0	51	0.16	0	54	0.62	0	58	–0.69
28800	0	25	0.16	0	27	0.76	0	29	1.27	0	32	–1.36	0	34	–0.79	0	36	–0.29	0	38	0.16
31250	0	23	0.00	0	25	0.00	0	27	0.00	0	29	0.00	0	31	0.00	0	33	0.00	0	35	0.00
38400	0	19	–2.34	0	20	0.76	0	22	–0.93	0	23	1.73	0	25	0.16	0	27	–1.18	0	28	1.02
115200	0	6	–6.99	0	6	0.76	0	7	–5.06	0	7	1.73	0	8	–3.55	0	8	2.48	0	9	–2.34
500000	0	1	–25.0	0	1	–18.8	0	1	–12.5	0	1	–6.25	0	1	0.00	0	1	6.25	0	1	12.5

Note 1. These values assume bits SEMR.ABCS0 and SEMR.BGDM are both 0. When either the SEMR.ABCS0 bit or SEMR.BGDM bit is set to 1, the bit rate is doubled. When bits SEMR.ABCS0 and SEMR.BGDM are both 1, the bit rate is quadrupled. Configure settings so the range of error is no greater than 1%.

Note 2. Continuous transmit/receive is not possible.

Table 41.7 Bit Rates and BRR Settings *1 in Asynchronous Mode (2/2)

Bit Rate [bps]	PCLKA (MHz)																				
	38			40			42			44			46			48			50		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	168	–0.19	3	177	–0.25	3	185	0.23	3	194	0.16	3	203	0.09	3	212	0.03	3	221	–0.02
150	3	123	–0.24	3	129	0.16	3	136	–0.21	3	142	0.16	3	149	–0.17	3	155	0.16	3	162	–0.15
300	2	246	0.16	2	64	0.16	2	67	0.53	3	71	–0.54	3	74	–0.17	3	77	0.16	3	80	0.47
600	2	123	–0.24	2	129	0.16	2	136	–0.21	2	142	0.16	2	149	–0.17	2	155	0.16	2	162	–0.15
1200	1	246	0.16	1	64	0.16	1	67	0.53	2	71	–0.54	2	74	–0.17	2	77	0.16	2	80	0.47
2400	1	123	–0.24	1	129	0.16	1	136	–0.21	1	142	0.16	1	149	–0.17	1	155	0.16	1	162	–0.15
4800	0	246	0.16	0	64	0.16	0	67	0.53	1	71	–0.54	1	74	–0.17	1	77	0.16	1	80	0.47
9600	0	123	–0.24	0	129	0.16	0	136	–0.21	0	142	0.16	0	149	–0.17	0	155	0.16	0	162	–0.15
14400	0	81	0.57	0	86	–0.22	0	90	0.16	0	94	0.51	0	99	–0.17	0	103	0.16	0	108	–0.45
19200	0	61	–0.24	0	64	0.16	0	67	0.53	0	71	–0.54	0	74	–0.17	0	77	0.16	0	80	0.47
28800	0	40	0.57	0	42	0.94	0	45	–0.93	0	47	–0.54	0	49	–0.17	0	51	0.16	0	53	0.47
31250	0	37	0.00	0	39	0.00	0	41	0.00	0	43	0.00	0	45	0.00	0	47	0.00	0	49	0.00
38400	0	30	–0.24	0	32	–1.36	0	33	0.53	0	35	–0.54	0	36	1.18	0	38	0.16	0	40	–0.76
115200	0	9	3.08	0	10	–1.36	0	10	3.57	0	11	–0.54	0	11	3.99	0	12	0.16	0	13	–3.12
500000	0	1	18.8	0	2	–16.7	0	2	–12.5	0	2	–8.33	0	2	–4.17	0	2	0.00	0	2	4.17

Bit Rate [bps]	PCLKA (MHz)											
	60			80			100			120		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	255	4.04									
150	3	194	0.16	3	255	1.73						
300	3	97	–0.35	3	129	0.16	3	162	–0.15	3	194	0.16
600	3	48	–0.35	3	64	0.16	3	80	0.47	3	97	–0.35
1200	2	97	–0.35	2	129	0.16	2	162	–0.15	3	48	–0.35
2400	2	48	–0.35	2	64	0.16	2	80	0.47	2	97	–0.35
4800	1	97	–0.35	1	129	0.16	1	162	–0.15	2	48	–0.35
9600	0	194	0.16	1	64	0.16	1	80	0.47	1	97	–0.35
14400	0	129	0.16	0	173	–0.22	1	53	0.47	1	64	0.16
19200	0	97	–0.35	0	129	0.16	0	162	–0.15	1	48	–0.35
28800	0	64	0.16	0	86	–0.22	0	108	–0.45	0	129	0.16
31250	0	59	0.00	0	79	0.00	0	99	0.00	0	119	0.00
38400	0	48	–0.35	0	64	0.16	0	80	0.47	0	97	–0.35
115200	0	15	1.73	0	21	–1.36	0	26	0.47	0	32	–1.36
500000	0	3	–6.25	0	4	0.00	0	5	4.17	0	7	–6.25

Note 1. These values assume bits SEMR.ABCS0 and SEMR.BGDM are both 0. When either the SEMR.ABCS0 bit or SEMR.BGDM bit is set to 1, the bit rate is doubled. When bits SEMR.ABCS0 and SEMR.BGDM are both 1, the bit rate is quadrupled. Configure settings so the range of error is no greater than 1%.

Table 41.8 Bit Rates and BRR Settings *1 in Clock Synchronous Mode

Bit Rate [bps]	PCLKA (MHz)																							
	10		12		14		16		18		20		22		24		26		28		30		32		34	
	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N
250	3	155	3	187	3	218	3	249																		
500	3	77	3	93	3	108	3	169	3	140	3	155	3	171	3	187	3	202	3	218	3	233	3	249		
1000	2	155	2	187	2	218	2	249	3	69	3	77	3	85	3	93	3	101	3	108	3	116	3	124	3	132
2500	1	249	2	74	2	87	2	99	2	112	2	124	2	137	2	149	2	162	2	174	2	187	2	199	2	212
5000	1	124	1	149	1	174	1	199	1	224	1	249	2	68	2	74	2	80	2	87	2	93	2	99	2	105
10000	0	249	1	74	1	87	1	99	1	112	1	124	1	137	1	149	1	162	1	174	1	187	1	199	1	212
25000	0	99	0	119	0	139	0	159	0	179	0	199	0	219	0	239	1	64	1	69	1	74	1	79	1	84
50000	0	49	0	59	0	69	0	79	0	89	0	99	0	109	0	119	0	129	0	139	0	149	0	159	0	169
100000	0	24	0	29	0	34	0	39	0	44	0	49	0	54	0	59	0	64	0	69	0	74	0	79	0	84
250000	0	9	0	11	0	13	0	15	0	17	0	19	0	21	0	23	0	25	0	27	0	29	0	31	0	33
500000	0	4	0	5	0	6	0	7	0	8	0	9	0	10	0	11	0	12	0	13	0	14	0	15	0	16
1000000	—	—	0	2	—	—	0	3	—	—	0	4	—	—	0	5	—	—	0	6	—	—	0	7	—	—
2500000	0	0 *2	—	—	—	—	—	—	—	—	0	1	—	—	—	—	—	—	—	—	0	2	—	—	—	—
5000000			—	—	—	—	—	—	—	—	0	0 *2	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit Rate [bps]	PCLKA (MHz)																			
	38		40		42		44		46		48		50		60		80		100		120	
	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N
250																						
500																						
1000	3	147	3	155	3	163	3	171	3	179	3	187	3	194	3	233						
2500	2	237	2	249	3	65	3	68	3	71	3	74	3	77	3	93	3	124	3	155	3	187
5000	2	118	2	124	2	130	2	137	2	143	2	149	2	155	2	187	2	249	3	77	3	93
10000	1	237	1	249	2	65	2	68	2	71	2	74	2	77	2	93	2	124	2	155	2	187
25000	1	94	1	99	1	104	1	109	1	114	1	119	1	124	1	149	1	199	1	249	2	74
50000	0	189	0	199	0	209	0	219	0	229	0	239	0	249	1	74	1	99	1	124	1	149
100000	0	94	0	99	0	104	0	109	0	114	0	119	0	124	0	149	0	199	0	249	1	74
250000	0	37	0	39	0	41	0	43	0	45	0	47	0	49	0	59	0	79	0	99	0	119
500000	0	18	0	19	0	20	0	21	0	22	0	23	0	24	0	29	0	39	0	49	0	59
1000000	—	—	0	9	—	—	0	10	—	—	0	11	—	—	0	14	0	19	0	24	0	29
2500000	—	—	0	3	—	—	—	—	—	—	—	—	0	4	0	5	0	7	0	9	0	11
5000000	—	—	0	1	—	—	—	—	—	—	—	—	—	—	0	2	0	3	0	4	0	5

Blank: Setting is prohibited.

—: Setting is possible, but causes a bit rate error.

Note 1. Configure settings so the range of error is no greater than 1%.

Note 2. Continuous transmit/receive is not possible.

Table 41.9 lists the maximum bit rates for various frequencies in asynchronous mode and clock synchronous mode when the baud rate generator is used.

Table 41.9 Maximum Bit Rates for Various Frequencies with Baud Rate Generator

PCLKA [MHz]	Asynchronous Mode *1			Clock Synchronous Mode *2					
				Discontinuous Transmission/Reception			Continuous Transmission/Reception		
	Maximum Bit Rate [bps]	Settings		Maximum bit rate [bps]	Settings		Maximum bit rate [bps]	Settings	
		n	N		n	N		n	N
10	312,500	0	0	2,500,000	0	0	1,250,000	0	1
12	375,000	0	0	3,000,000	0	0	1,500,000	0	1
14	437,500	0	0	3,500,000	0	0	1,750,000	0	1
16	500,000	0	0	4,000,000	0	0	2,000,000	0	1
18	562,500	0	0	4,500,000	0	0	2,250,000	0	1
20	625,000	0	0	5,000,000	0	0	2,500,000	0	1
22	687,500	0	0	5,500,000	0	0	2,750,000	0	1
24	750,000	0	0	6,000,000	0	0	3,000,000	0	1
26	812,500	0	0	6,500,000	0	0	3,250,000	0	1
28	875,000	0	0	7,000,000	0	0	3,500,000	0	1
30	937,500	0	0	7,500,000	0	0	3,750,000	0	1
32	1,000,000	0	0	8,000,000	0	0	4,000,000	0	1
34	1,062,500	0	0	8,500,000	0	0	4,250,000	0	1
36	1,125,000	0	0	9,000,000	0	0	4,500,000	0	1
38	1,187,500	0	0	9,500,000	0	0	4,750,000	0	1
40	1,250,000	0	0	10,000,000	0	0	5,000,000	0	1
42	1,312,500	0	0	10,500,000	0	0	5,250,000	0	1
44	1,375,000	0	0	11,000,000	0	0	5,500,000	0	1
46	1,437,500	0	0	11,500,000	0	0	5,750,000	0	1
48	1,500,000	0	0	12,000,000	0	0	6,000,000	0	1
50	1,562,500	0	0	—	—	—	—	—	—
60	1,875,000	0	0	—	—	—	—	—	—
80	2,500,000	0	0	—	—	—	—	—	—
100	3,125,000	0	0	—	—	—	—	—	—
120	3,750,000	0	0	—	—	—	—	—	—

Note 1. The maximum bit rate in asynchronous mode assume bits SEMR.ABCS0 and SEMR.BGDM are both 0. If either the SEMR.ABCS0 bit or SEMR.BGDM bit is 1, the bit rate is doubled. When bits SEMR.ABCS0 and SEMR.BGDM are both 1, the bit rate is quadrupled. Configure settings so the range of error falls within 1%.

Note 2. The maximum bit rate setting should satisfy the electrical characteristics.

Table 41.10 lists the maximum rates for external clock inputs in asynchronous mode and clock synchronous mode.

Table 41.10 Maximum Bit Rates with External Clock Input

PCLKA [MHz]	External Input Clock [MHz]		Maximum Bit Rate [bps]	
	Asynchronous mode	Clock synchronous mode	Asynchronous mode *1	Clock synchronous mode *2
10	2.5	1.6667	156,250	1,666,667
12	3.0	2.0000	187,500	2,000,000
14	3.5	2.3333	218,750	2,333,333
16	4.0	2.6667	250,000	2,666,667
18	4.5	3.0000	281,250	3,000,000
20	5.0	3.3333	312,500	3,333,333
22	5.5	3.6667	343,750	3,666,667
24	6.0	4.0000	375,000	4,000,000
26	6.5	4.3333	406,250	4,333,333
28	7.0	4.6667	437,500	4,666,667
30	7.5	5.0000	468,750	5,000,000
32	8.0	5.3333	500,000	5,333,333
34	8.5	5.6667	531,250	5,666,667
36	9.0	6.0000	562,500	6,000,000
38	9.5	6.3333	593,750	6,333,333
40	10.0	6.6667	625,000	6,666,667
42	10.5	7.0000	656,250	7,000,000
44	11.0	7.3333	687,500	7,333,333
46	11.5	7.6667	718,750	7,666,667
48	12.0	8.0000	750,000	8,000,000
50	12.5	8.3333	781,250	8,333,333
60	15.0	5	937,500	5,000,000
80	20.0	6.6667	1,250,000	6,666,667
100	25.0	8.3333	1,562,500	8,333,333
120	30.0	—	1,875,000	—

Note 1. These are the values when the SEMR.ABCS0 bit is 0. The bit rate is doubled with the SEMR.ABCS0 bit is 1.
Note 2. The maximum bit rate setting should satisfy the electrical characteristics.

41.2.9 Modulation Duty Register (MDDR)

Addresses: SCIFA8.MDDR 000D 0002h, SCIFA9.MDDR 000D 0022h, SCIFA10.MDDR 000D 0042h, SCIFA11.MDDR 000D 0062h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
	1							
Value after reset	1	1	1	1	1	1	1	1

The MDDR register corrects the bit rate adjusted by the BRR register. When the SEMR.BRME bit is set to 1, the bit rate generated by the on-chip baud rate generator is corrected. The MDDR register is located in the same address as the BRR register, and the MDDR register can be accessed when the SEMR.MDDRS bit is 1. Rewrite this register when bits SCR.TE and RE are both 0. Bit 7 in this register is fixed to 1.

The formulae in Table 41.11 can be used to calculate the bit rate when using the bit rate modulation function.

Table 41.11 Calculating the Bit Rate When Using the Bit Rate Modulation Function

Mode		Operate the SCIFA at 16 Times the Bit Rate (SEMR.ABCS0 bit is 0)	Operate the SCIFA at 8 Times the Bit Rate (SEMR.ABCS0 bit is 1)
Asynchronous mode	Baud rate generator is in normal mode (SEMR.BGDM bit is 0)	$B = \frac{f_{(PCLKA)} \times 10^6}{2^{2n} \times 2 \times 16 \times \frac{256}{MDDR} \times (N+1)}$	$B = \frac{f_{(PCLKA)} \times 10^6}{2^{2n} \times 2 \times 8 \times \frac{256}{MDDR} \times (N+1)}$
	Baud rate generator is in high-speed mode (SEMR.BGDM bit is 1)	$B = \frac{f_{(PCLKA)} \times 10^6}{2^{2n} \times 16 \times \frac{256}{MDDR} \times (N+1)}$	$B = \frac{f_{(PCLKA)} \times 10^6}{2^{2n} \times 8 \times \frac{256}{MDDR} \times (N+1)}$
Clock synchronous mode		$B = \frac{f_{(PCLKA)} \times 10^6}{2^{2n} \times 4 \times \frac{256}{MDDR} \times (N+1)}$	

B: Bit rate [bps]
$f_{(PCLKA)}$: Frequency of the peripheral module clock [MHz]
n: SMR.CKS[1:0] bit setting value (n = 0 to 3) (see Table 41.4.).
MDDR: MDDR register setting value ($128 \le MDDR \le 256$)
N: BRR register setting value ($0 \le N \le 255$) (the bit rate value must satisfy the electrical characteristics).

The formulae in Table 41.12 can be used to calculate the average bit rate error when using the bit rate modulation function in asynchronous mode.

Table 41.12 Bit Rate Error When Using the Bit Rate Modulation Function in Asynchronous Mode

Mode	Operate the SCIFA at 16 Times the Bit Rate (SEMR.ABCS0 bit is 0)	Operate the SCIFA at 8 Times the Bit Rate (SEMR.ABCS0 bit is 1)
Baud rate generator normal mode (SEMR.BGDM bit is 0)	$\left(\frac{f_{(PCLKA)} \times 10^6}{2^{2n} \times 2 \times 16 \times B \times \frac{256}{MDDR} \times (N+1)} - 1\right) \times 100$	$\left(\frac{f_{(PCLKA)} \times 10^6}{2^{2n} \times 2 \times 8 \times B \times \frac{256}{MDDR} \times (N+1)} - 1\right) \times 100$
Baud rate generator double speed mode (SEMR.BGDM bit is 1)	$\left(\frac{f_{(PCLKA)} \times 10^6}{2^{2n} \times 16 \times B \times \frac{256}{MDDR} \times (N+1)} - 1\right) \times 100$	$\left(\frac{f_{(PCLKA)} \times 10^6}{2^{2n} \times 8 \times B \times \frac{256}{MDDR} \times (N+1)} - 1\right) \times 100$

41.2.10 FIFO Control Register (FCR)

Addresses: SCIFA8.FCR 000D 000Ch, SCIFA9.FCR 000D 002Ch, SCIFA10.FCR 000D 004Ch, SCIFA11.FCR 000D 006Ch

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	—	—	—	—	—	RSTRG[2:0]			RTRG[1:0]		TTRG[1:0]		MCE	TFRST	RFRST	LOOP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description		R/W
b0	LOOP	Loopback Test Control Bit	0: Loopback test disabled 1: Loopback test enabled		R/W
b1	RFRST	FRDR Register Reset Bit	0: FRDR register is not reset 1: FRDR register is reset		R/W
b2	TFRST	FTDR Register Reset Bit	0: FTDR register is not reset 1: FTDR register is reset		R/W
b3	MCE	Flow Control Bit	0: Hardware flow control by the modem control signal is disabled 1: Hardware flow control by the modem control signal is enabled		R/W
b5, b4	TTRG[1:0]	Transmit FIFO Threshold Setting Bits	b7 b6 0 0: 8 (8) *1 0 1: 4 (12) *1	b7 b6 1 0: 2 (14) *1 1 1: 0 (16) *1	R/W
b7, b6	RTRG[1:0]	Receive FIFO Threshold Setting Bits	Asynchronous mode b7 b6 0 0: 1 0 1: 4 1 0: 8 1 1: 14	Clock synchronous mode b7 b6 0 0: 1 0 1: 2 1 0: 8 1 1: 14	R/W
b10 to b8	RSTRG[2:0]	RTS# Output Threshold Setting Bits	b10 b8 0 0 0: 15 0 0 1: 1 0 1 0: 4 0 1 1: 6	b10 b8 1 0 0: 8 1 0 1: 10 1 1 0: 12 1 1 1: 14	R/W
b15 to b11	—	Reserved	These bits are 0 when read. Set them to 0 when writing.		R/W

Note 1. Values in parentheses represent the number of empty bytes in the FTDR register when the TDFE flag becomes 1.

The FCR register is used to reset the amount of data stored in the FTDR and FRDR registers, set the transmit FIFO threshold value, and set the receive FIFO threshold value. This register also enables and disables loopback testing.

LOOP Bit (Loopback Test Control Bit)

When the LOOP bit is set to 1, the TXD and RXD pins are connected internally, and a loopback test is enabled.

RFRST Bit (FRDR Register Reset Bit)

When the RFRST bit is set to 1, the FRDR register is reset and emptied of all receive data. After setting the RFRST bit to 1, set it back to 0.

TFRST Bit (FTDR Register Reset Bit)

When the TFRST bit is set to 1, the FTDR register is reset and emptied of all transmit data. After setting the TFRST bit to 1, set it back to 0.

MCE Bit (Flow Control Bit)

This bit enables and disables the hardware flow control by the modem control signals (CTS# and RTS#). Set the MCE bit to 0 in clock synchronous mode.

Note: Regardless of the input value, the level of the CTS# pin has no effect on transmit operation of frames being transmitted. The level of RTS# pin has no effect on the receive operation.

TTRG[1:0] Bits (Transmit FIFO Threshold Setting Bits)

When the FTCR.TTRGS bit is 0, the transmit FIFO threshold value is determined by the TTRG[1:0] bit setting. When the FTCR.TTRGS bit is 1, the transmit FIFO threshold value is determined by the FTCR.TFTC[4:0] bit setting. When the amount of data stored in the FTDR register becomes the transmit FIFO threshold value or less, and the FSR.TDFE flag becomes 1. At that time, if the SCR.TIE bit is 1, a TXIF interrupt request is generated.

RTRG[1:0] Bits (Receive FIFO Threshold Setting Bits)

When the FTCR.RTRGS bit is 0, the receive FIFO threshold value is determined by the RTRG[1:0] bit setting. When the FTCR.RTRGS bit is 1, the receive FIFO threshold value is determined by the FTCR.RFTC[4:0] bit setting. When the amount of data stored in the FRDR register becomes the receive FIFO threshold value or more, the FSR.RDF flag becomes 1, and the FRDR register becomes read accessible. At that time, if the SCR.RIE bit is 1, an RXIF interrupt request is generated.

RSTRG[2:0] Bits (RTS# Output Threshold Setting Bits)

When the amount of receive data stored in the FRDR register becomes the RSTRG[2:0] bit value or higher, the RTS# signal becomes high. This bit is only valid when the MCE bit is set to 1 while in asynchronous mode.

41.2.11 FIFO Data Count Register (FDR)

Addresses: SCIFA8.FDR 000D 000Eh, SCIFA9.FDR 000D 002Eh, SCIFA10.FDR 000D 004Eh, SCIFA11.FDR 000D 006Eh

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	—	—	—	T[4:0]					—	—	—	R[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	R[4:0]	Receive FIFO Storage Data Count Flags	These flags indicate the amount of receive data stored in the FRDR register.	R
b7 to b5	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b12 to b8	T[4:0]	Transmit FIFO Storage Data Count Flags	These flags indicate the amount of untransmitted data stored in the FTDR register.	R
b15 to b13	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

The FDR register indicates the amount of data stored in the FTDR register and FRDR register.

R[4:0] Flags (Receive FIFO Storage Data Count Flags)

These flags indicate the amount of receive data stored in the FRDR register. 00h indicates there is no receive data, and 10h indicates the FRDR register is full.

T[4:0] Flags (Transmit FIFO Storage Data Count Flags)

These flags indicate the amount of non-transmitted data stored in the FTDR register. 00h indicates there is no transmit data, and 10h indicates the FTDR register is full.

41.2.12 Serial Port Register (SPTR)

Addresses: SCIFA8.SPTR 000D 0010h, SCIFA9.SPTR 000D 0030h, SCIFA10.SPTR 000D 0050h, SCIFA11.SPTR 000D 0070h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	—	—	—	—	—	—	—	—	RTS2IO	RTS2DT	CTS2IO	CTS2DT	SCKIO	SCKDT	SPB2IO	SPB2DT
Value after reset	0	0	0	0	0	0	0	0	0	x	0	x	0	x	0	x

x: undefined

Bit	Symbol	Bit Name	Description	R/W
b0	SPB2DT	Serial Port Break Data Bit	Combine bits SPB2DT, SPB2IO, and the SCR.TE bit to control the TXD pin. Refer to Table 41.13 for details. Reading the SPB2DT bit returns the status of the RXD pin.	R/W
b1	SPB2IO	Serial Port Break I/O Bit		R/W
b2	SCKDT	SCK Port Data Bit	When the SMR.CM bit is 0 (asynchronous mode), combine bits SCKDT, SCKIO, and SCR.CKE[1:0] to control the SCK pin. Refer to Table 41.17 for details. Reading the SCKDT bit returns the status of the SCK pin.	R/W
b3	SCKIO	SCK Port I/O Bit		R/W
b4	CTS2DT	CTS# Port Data Select Bit	Combine bits CTS2DT, CTS2IO, and FCR.MCE to control the CTS# pin. Refer to Table 41.14 for details. Reading the CTS2DT bit returns the status of the CTS# pin.	R/W
b5	CTS2IO	CTS# Port Output Designated Bit		R/W
b6	RTS2DT	RTS# Port Data Select Bit	Combine bits RTS2DT, RTS2IO, and FCR.MCE to control the RTS# pin. Refer to Table 41.15 for details. Reading the RTS2DT bit returns the status of the RTS# pin.	R/W
b7	RTS2IO	RTS# Port Output Designated Bit		R/W
b15 to b8	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note: Do not use the read-modify-write instructions for this register. Use the MOV instruction instead.

The SPTR register is used for directly controlling the SCIFA I/O pins by software.

SPB2DT Bit (Serial Port Break Data Bit)

This bit designates the output level of the TXD pin when the SCR.TE bit is 0. Refer to Table 41.13 for details. When this bit is read, the status of the RXD pin is read regardless of the SPB2IO bit value. However, the RXD pin function must be selected in the MPC module.

SPB2IO Bit (Serial Port Break I/O Bit)

This bit designates the TXD pin input and output when the SCR.TE bit is 0. When directly controlling the TXD port by software, set the SPB2IO bit to 1 (output).

Table 41.13 Controlling the TXD Pin

SCR.TE Bit Setting	SPB2IO Bit Setting	SPB2DT Bit Setting	TXD Pin Status
0	0	0 or 1	Hi-Z
0	1	0	Low output
0	1	1	High output
1	0 or 1	0 or 1	Pin for outputting transmit data

SCKDT Bit (SCK Port Data Bit)

When the SMR.CM bit is 0 (asynchronous mode) and the SCR.CKE[1:0] bits are 00b, this bit selects the output level for the SCK pin. See Table 41.17 for details. When this bit is read, the SCK pin status can be read despite the SCKIO bit value. However, the SCK pin function must be selected in the MPC module.

SCKIO Bit (SCK Port I/O Bit)

When the SMR.CM bit is 0 (asynchronous mode) and the SCR.CKE[1:0] bits are 00b, this bit designates the SCK pin to input/output. When using this function, set the SCKIO bit to 1 (output).

CTS2DT Bit (CTS# Port Data Select Bit)

When the FCR.MCE bit is 0, this bit selects the output level for the CTS# pin. Refer to Table 41.14 for details. When this bit is read, the CTS# pin status can be read despite the CTS2IO bit value. However, the CTS# pin function must be

selected in the MPC module.

CTS2IO Bit (CTS# Port Output Designated Bit)

This bit designates the CTS# pin input and output when the SCR.TE bit is 0. When using this function, set the CTS2IO bit to 1 (output).

Table 41.14 Controlling the CTS# Pin

FCR.MCE Bit Setting	CTS2IO Bit Setting	CTS2DT Bit Setting	CTS# Pin Status
0	0	0 or 1	Hi-Z
0	1	0	Low output
0	1	1	High output
1	0 or 1	0 or 1	Input to the flow control logic

RTS2DT Bit (RTS# Port Data Select Bit)

This bit designates the output level of the RTS# pin when the FCR.MCE bit is 0. Refer to Table 41.15 for details. When this bit is read, the RTS# pin status can be read despite the RTS2IO bit value. However, the RTS# pin function must be selected in the MPC module.

RTS2IO Bit (RTS# Port Output Designated Bit)

This bit designates the RTS# pin input and output when the FCR.MCE bit is 0. When using this function, set the RTS2IO bit to 1 (output).

Table 41.15 Controlling the RTS# Pin

FCR.MCE Bit Setting	RTS2IO Bit Setting	RTS2DT Bit Setting	RTS# Pin Status
0	0	0 or 1	Hi-Z
0	1	0	Low output
0	1	1	High output
1	0 or 1	0 or 1	Sequence output in accordance with flow control logic

41.2.13 Line Status Register (LSR)

Addresses: SCIFA8.LSR 000D 0012h, SCIFA9.LSR 000D 0032h, SCIFA10.LSR 000D 0052h, SCIFA11.LSR 000D 0072h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	—	—	—	—	PER[3:0]				—	—	FER[3:0]				—	ORER
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ORER	Overrun Error Flag	0: Overrun error did not occur *1 1: Overrun error occurred *2	R(/W) *3
b1	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b5 to b2	FER[3:0]	Framing Error Count Flags	Among the receive data that is stored in the FRDR register, these flags indicate the amount of data that is in the framing error state.	R
b7, b6	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b11 to b8	PER[3:0]	Parity Error Count Flags	Among the receive data that is stored in the FRDR register, these flags indicate the amount of data that is in the parity error state.	R
b15 to b12	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. The ORER flag does not become 0 even if the SCR.RE bit is set to 0.
Note 2. If an overrun error occurs, data already stored in the FRDR register is not overwritten, and the data that caused the error is discarded. After the ORER flag becomes 1, subsequent serial reception cannot continue.
Note 3. The ORER flag can be written only to clear the flag. To clear the flag, confirm that the flag is 1 before setting it to 0.

LSR is a 16-bit register. Flags PER[3:0] and FER[3:0] indicate the number of receive errors that occurred to the data stored in the FRDR register.

ORER Flag (Overrun Error Flag)

This flag indicates that reception stopped when an overrun error occurred during the receive operation.

— This flag becomes 1 under the following condition:

- While 16-byte data is being stored in the receive FIFO, the next serial reception is completed.

— This flag becomes 0 under the following condition:

- After confirming that the ORER flag is 1, it is set to 0.

Note: When the internal clock is selected while the SCIFA is in clock synchronous mode, the amount of receive data can be controlled, so no overrun occurs.

FER[3:0] Flags (Framing Error Count Flags)

When the FSR.ER flag is 1, these flags indicate the amount of data that is in the framing error state from among the receive data stored in the FRDR register. When the FRDR register receives 16 bytes of data and all of the received data is in the framing error state, the FER[3:0] flags become 0000b. When received data with a framing error is read from the FRDR register, the value of the FER[3:0] flags is decremented by one.

PER[3:0] Flags (Parity Error Count Flags)

When the FSR.ER flag is 1, these flags indicate the amount of data that is in the parity error state from among the receive data stored in the FRDR register. When the FRDR register receives 16 bytes of data and all of the received data is in the parity error state, the PER[3:0] flags become 0000b. When received data with a parity error is read from the FRDR register, the value of the PER[3:0] flags is decremented by one.

41.2.14 FIFO Trigger Control Register (FTCR)

Addresses: SCIFA8.FTCR 000D 0016h, SCIFA9.FTCR 000D 0036h, SCIFA10.FTCR 000D 0056h, SCIFA11.FTCR 000D 0076h

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	RTRGS	—	—	RFTC[4:0]					TTRGS	—	—	TFTC[4:0]				
Value after reset	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	TFTC[4:0]	Transmit FIFO Threshold Setting Bits	b4 b0 0 0 0 0 0: Transmit FIFO threshold value becomes 0 ⋮ 0 1 1 1 1: Transmit FIFO threshold value becomes 15 Do not set these bits to a value from 10000b to 11111b.	R/W
b6, b5	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b7	TTRGS	Transmit FIFO Threshold Select Bit	0: Transmit FIFO threshold value set in the FCR.TTRG[1:0] bits 1: Transmit FIFO threshold value set in the FTCR.TFTC[4:0]bits	R/W
b12 to b8	RFTC[4:0]	Receive FIFO Threshold Setting Bits	b12 b8 0 0 0 0 1: Receive FIFO threshold value becomes 1 ⋮ 0 1 1 1 1: Receive FIFO threshold value becomes 15 Do not set these bits to 00000b, or 10000b to 11111b.	R/W
b14, b13	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b15	RTRGS	Receive FIFO Threshold Select Bit	0: Receive FIFO threshold value set in the FCR.RTRG[1:0] bits 1: Receive FIFO threshold value set in the FTCR.RFTC[4:0] bits	R/W

The FTCR register is used for writing the threshold value to the transmit FIFO and reading the threshold value from the receive FIFO.

TFTC[4:0] Bits (Transmit FIFO Threshold Setting Bits)

These bits set the transmit FIFO threshold value when the TTRGS bit is 1. When the TTRGS bit is 0, the FCR.TTRG[1:0] bit value becomes valid, and the TFTC[4:0] bit value becomes invalid. When the amount of transmit data stored in the FTDR register (transmit FIFO) becomes less than or equal to the value set in the TFTC[4:0] bits, the FSR.TDFE flag becomes 1, and a write access is requested for transmit data. At this point, if the SCR.TIE bit is 1, a TXIF interrupt request is generated.

RFTC[4:0] Bits (Receive FIFO Threshold Setting Bits)

These bits set the receive FIFO threshold value when the RTRGS bit is 1. When the RTRGS bit is 0, the FCR.RTRG[1:0] bit value becomes valid, and the RFTC[4:0] bit value becomes invalid. When the amount of receive data stored in the FRDR register (receive FIFO) becomes equal to or more than the value set in the RFTC[4:0] bits, the FSR.RDF flag becomes 1, and a read access is requested for receive data. At this point, if the SCR.RIE bit is 1, an RXIF interrupt request is generated.

41.2.15 Serial Extended Mode Register (SEMR)

Addresses SCIFA8.SEMR 000D 0014h, SCIFA9.SEMR 000D 0034h, SCIFA10.SEMR 000D 0054h, SCIFA11.SEMR 000D 0074h

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	BGDM	—	BRME	MDDRS	DIR	NFEN	—	ABCS0
Value after reset	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ABCS0	Asynchronous Mode Base Clock Select Bit	0: 16 cycles of the baud rate generator's output is a 1-bit period 1: 8 cycles of the baud rate generator's output is a 1-bit period	R/W
b1	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b2	NFEN	Noise Canceller Control Bit	0: RXD pin noise canceller disabled 1: RXD pin noise canceller enabled	R/W
b3	DIR	Data Transfer Direction Select Bit	0: Data transfer with LSB first 1: Data transfer with MSB first	R/W
b4	MDDRS	Modulation Register Select Bit	0: BRR register is accessible 1: MDDR register is accessible	R/W
b5	BRME	Bit Rate Modulation Control Bit	0: Bit rate modulation function disabled 1: Bit rate modulation function enabled	R/W
b6	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b7	BGDM	Baud Rate Generator Operating Speed Select Bit	0: In baud rate generator normal mode, the baud rate generator operates on a clock signal produced by the clock source divided by 2 1: In baud rate generator double speed mode, the baud rate generator operates on the clock signal produced by the clock source (no division)	R/W

The SEMR register controls SCIFA operation.

ABCS0 Bit (Asynchronous Mode Base Clock Select Bit)

While in asynchronous mode, set this bit to select the correspondence between the base clock (baud rate generator output) and a 1-bit period. The ABCS0 bit is only valid in asynchronous mode.

NFEN Bit (Noise Canceller Control Bit)

Set this bit to enable and disable the noise cancelling function on input signals from the RXD pin. This function can only be enabled in asynchronous mode. Refer to section 41.3.5, Noise Cancellation for details. Set this bit to 0 in clock synchronous mode.

DIR Bit (Data Transfer Direction Select Bit)

Set this bit to select the bit order (LSB first or MSB first) of transmit/receive data. This bit is only valid when the character length is 8 bits.

MDDRS Bit (Modulation Register Select Bit)

The BRR register and MDDR register are allocated to the same address. Set this bit to select which register is accessed when accessing that address.

BGDM Bit (Baud Rate Generator Operating Speed Select Bit)

Set this bit to select the operating mode for the baud rate generator. When the BGDM bit is 1, the baud rate generator operates in double speed mode. The baud rate generator double speed mode is only enabled when the SCIFA is in asynchronous mode and bits SCR.CKE[1:0] are 00b. Set the BGDM bit to 0 when bits SCR.CKE[1:0] are 01b, 10b, or 11b, and when in clock synchronous mode.

41.3 Operation

The SCIFA can be operated in asynchronous mode or clock synchronous mode.

41.3.1 Operating the SCIFA in Asynchronous Mode

Figure 41.2 shows the general format for asynchronous serial communication. In asynchronous serial communication, the communication line is normally in the “marked” state (high). The SCIFA monitors the communication line. When the line becomes “space” (low), that is regarded as the start bit, and the SCIFA starts serial communication. Full duplex communication is available as the SCIFA’s transmitter and receiver operate independently. As the transmit buffer and receive buffer are each configured as 16-stage FIFO buffers, data can be written and read during transmission or reception, and transmission and reception can be set to continuous.
In asynchronous mode, the SCIFA synchronizes the receive data and base clock at the start bit’s falling edge. When bits SEMR.ABCS0 and SEMR.BGDM are 0, the base clock of the SCIFA becomes 16 times the frequency of a 1-bit period, and each bit of the receive data is sampled at the eighth clock, so receive data is acquired in the middle of every bit. When either the SEMR.ABCS0 bit or SEMR.BGDM bit is 1, the base clock of the SCIFA becomes eight times the frequency of a 1-bit period, and each bit of the receive data is sampled at the fourth clock, so receive data is acquired in the middle of every bit.

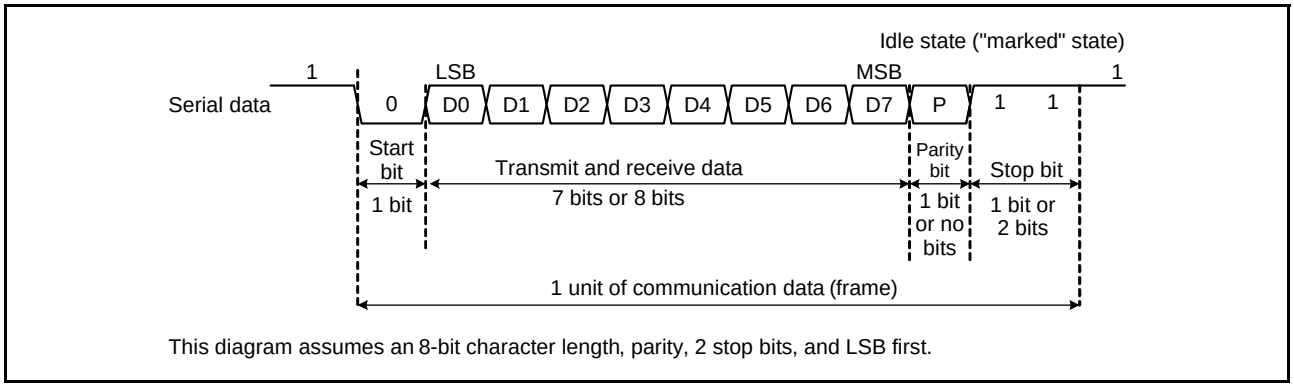


Figure 41.2 Data Format for Asynchronous Serial Communication

41.3.1.1 Serial Transmit/Receive Data Formats in Asynchronous Mode

Table 41.16 lists the formats that can be set in asynchronous mode. There are eight formats available, and the format can be selected by setting the SMR register.

Table 41.16 Serial Transmit/Receive Formats in Asynchronous Mode

SMR Register Setting				Serial Transmit/Receive Format and Frame Length											
CM	CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	Start	8-bit data								Stop		
			1	Start	8-bit data								Stop	Stop	
		1	0	Start	8-bit data								Parity	Stop	
			1	Start	8-bit data								Parity	Stop	Stop
	1	0	0	Start	7-bit data							Stop			
			1	Start	7-bit data							Stop	Stop		
		1	0	Start	7-bit data							Parity	Stop		
			1	Start	7-bit data							Parity	Stop	Stop	

41.3.1.2 Sampling Timing of Receive Data and the Reception Margin While in Asynchronous Mode

The SCIFA operates with a base clock frequency of eight times or 16 times the bit rate. The SCIFA imports receive data at the rising edge of the base clock's fourth clock or eighth clock. This timing is shown in Figure 41.3.

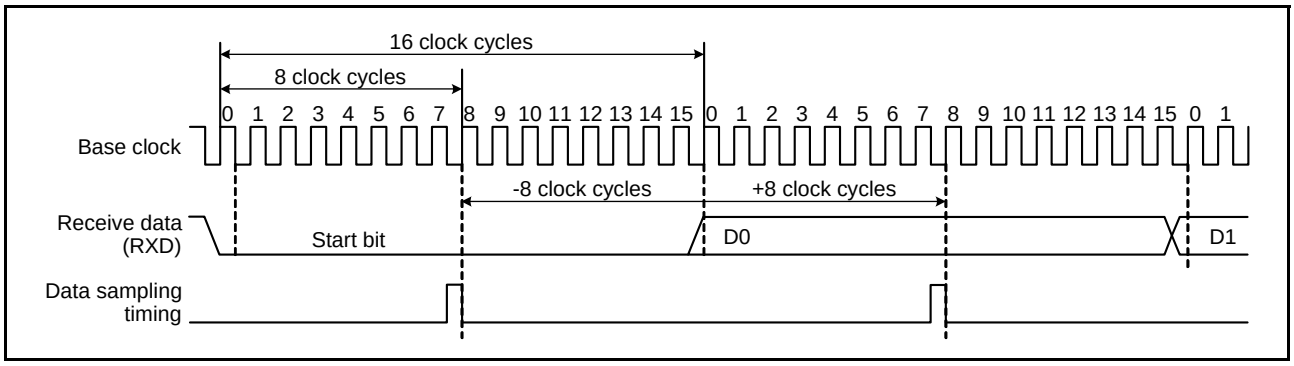


Figure 41.3 Sampling Timing and Reception Margin of Receive Data in Asynchronous Mode

The reception margin of receive data in asynchronous mode can be calculated using the following formula.

$$M = \left(0.5 - \frac{1}{2n} - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F)\right) \times 100[\%]$$

M: Reception margin [%]
N: Clock frequency ratio of the bit rate (N = 8 or 16)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of the difference in the clock frequencies
Assuming N = 16, F = 0, and D = 0.5, the following formula gives the reception margin as 46.875%.

$$M = \left(0.5 - \frac{1}{2 \times 16} - 0 - 0\right) \times 100$$

$$= 46.875$$

However, as this is a theoretical value, allow a 20% to 30% margin in system designs.

41.3.1.3 Clocks

The clock source for asynchronous mode can be selected from the internal clock generated by the baud rate generator, or the external clock input from the SCK pin. Refer to Table 41.17 for details on the clock source and register settings while the SCIFA is in asynchronous mode.
When inputting an external clock from the SCK pin, input a clock that has a frequency that is eight times or 16 times the bit rate used.
When operating the SCIFA with the internal clock, the clock can be output from the SCK pin. The frequency of the clock being output at this time is eight times or 16 times the bit rate.

Table 41.17 Setting Values to Registers in Clock Asynchronous Mode and Clock

SMR	SCR		SPTR		Clock Source	SCK Pin Function
CM	CKE[1]	CKE[0]	SCKIO	SCKDT		
0	0	0	0	0 or 1	Internal	Input pin (input signal is invalid)
			1	0		SCK pin outputs a low
			1	1		SCK pin outputs a high
		1	0 or 1	0 or 1		Outputs a clock with frequency that is 8 times or 16 times *1 the bit rate
	1	0	0 or 1	0 or 1	External	Inputs a clock with frequency that is 8 times or 16 times *1 the bit rate
		1	0 or 1	0 or 1	Do not set this value.	

Note 1. When using a clock that is 16 times the bit rate, set bits SEMR.ABCS0 and BGDM to 0. When using a clock that is eight times the bit rate, set the ABCS0 bit to 1 and set the BGDM bit to 0.

41.3.1.4 SCIFA Initialization in Asynchronous Mode

Figure 41.4 shows an example of the initialization when SCIFA is in asynchronous mode.
Set the SCR.TE and RE bits to 0 before initializing SCIFA. Also, set the SCR.TE and RE bits to 0 before changing the operating mode and communications format of SCIFA. When changing the SCIFA operating mode and communications format, follow the procedure in Figure 41.4.
The TSR register is initialized when the SCR.TE bit is set to 0. However, even if the SCR.RE bit is set to 0, the FSR.RDF, PER, FER flags, the LSR.ORER flag, and the FRDR register are not initialized, and their content is retained. To continue data reception, the FSR.RDF, PER, FER flags, the LSR.ORER flag, and the FRDR register must be initialized.
Typically, in order to set the SCR.TE bit to 0 to stop transmission, the user must first confirm that the FSR.TEND flag is 1 (transmission completed), but transmission can be stopped while data is being transmitted. In this case, the TXD pin status is dependent on the values of the SPTR.SPB2IO and SPB2DT bits. Also, after transmission is stopped, if transmission is restarted, set the FCR.TFRST bit to 1 and then reset the FTDR register.

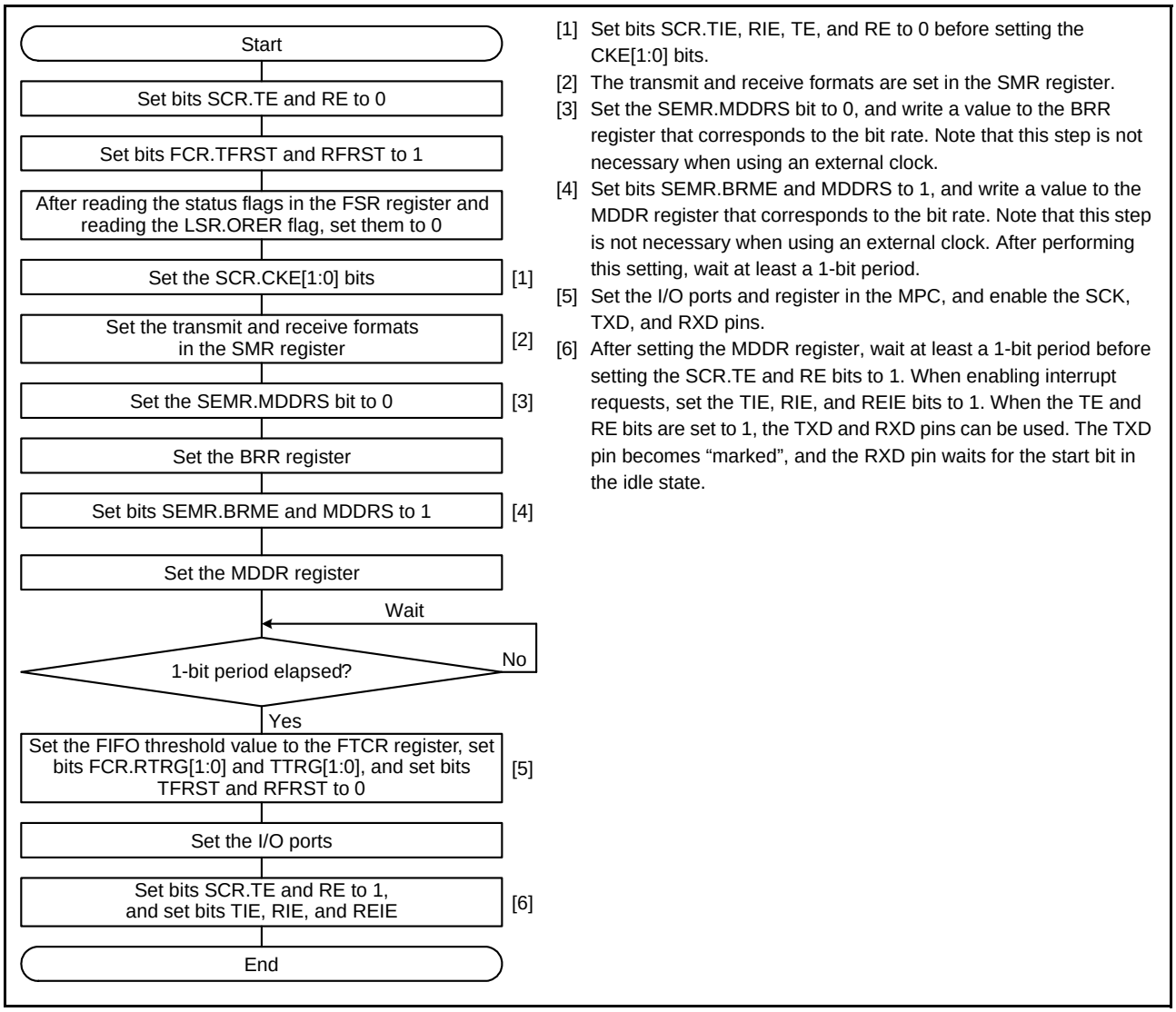


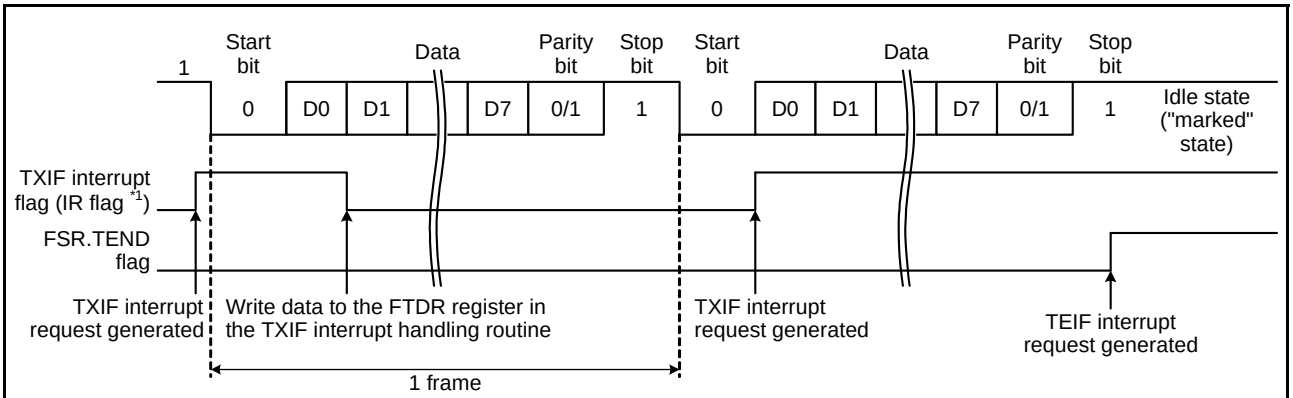
Figure 41.4 SCIFA Initialization in Asynchronous Mode

41.3.1.5 Transmitting Serial Data in Asynchronous Mode

This section describes data transmission in asynchronous mode.

1. When data is written to the FTDR register, the SCIFA transfers data from the FTDR register to the TSR register. Before writing transmit data to the FTDR register, confirm the FSR.TDFE flag is 1. The amount of transmit data bytes that can be written is calculated as (16 - FDR.T[4:0]).
2. When data transfer from the FTDR register to the TSR register starts, the SCIFA continuously transmits data until data written to the FTDR register is gone. When the amount of data written in the FTDR register is less than or equal to the transmit FIFO threshold value *1, the FSR.TDFE flag becomes 1. At this time, if the SCR.TIE bit is 1, the SCIFA generates the TXIF interrupt request.
3. The TXD pin outputs the start bit, transmit data, parity bit (when the SMR.PE bit is 1), and the stop bit (in that order).
4. When the stop bit is output, the SCIFA checks if there is any untransmitted data in the FTDR register. If the FTDR register contains data, it is transferred to the TSR register, and after the stop bit is output, the next frame of data starts transmitting. If there is no data in the FTDR register, set the FSR.TEND flag to 1 and output the stop bit before configuring the TXD pin to the “marked” state.
5. When the FCR.MCE bit is 1 (hardware flow control is enabled), the SCIFA starts or stops data transmission according to the value being input to the CTS# pin. If the CTS# pin becomes high during data transmission, when that frame is done transmitting, stop transmission, and configure the TXD pin to the “marked” state. If the CTS# pin becomes low while data transmission is stopped, SCIFA starts data transmission. Refer to Figure 41.5 for details on SCIFA operation when not using hardware flow control, and refer to Figure 41.6 for details on SCIFA operation when using hardware flow control.

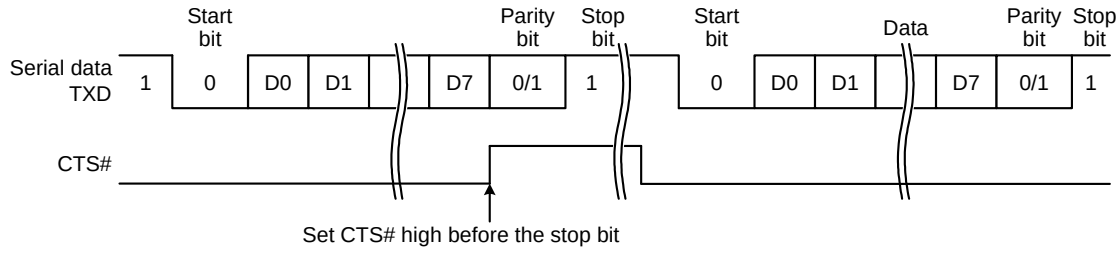
Note 1. When the FTCR.TTRGS bit is 0, the FCR.TTRG[1:0] bits determine the transmit FIFO threshold value; when the FTCR.TTRGS bit is 1, the FTCR.TFTC[4:0] bits determine the transmit FIFO threshold value.



The above diagram assumes 8-bit data, parity, 1 stop bit, LSB first, and no hardware flow control.

Note 1: Refer to the section 15. Interrupt Controller (ICUA) for the corresponding interrupt number.

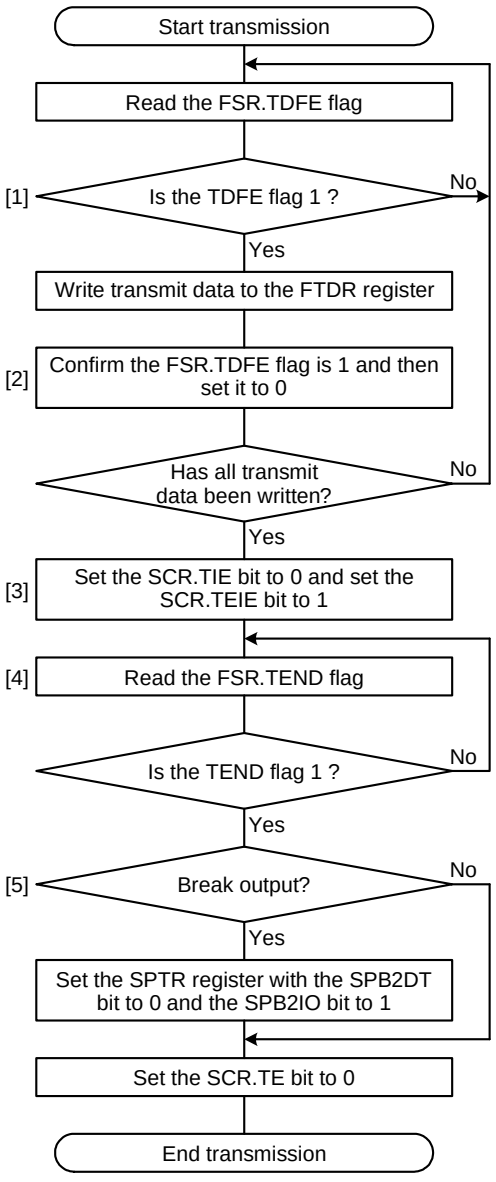
Figure 41.5 SCIFA Operation When Transmitting Data (1/2)



The above diagram assumes 8-bit data, parity, 1 stop bit, LSB first, and hardware flow control.

Figure 41.6 SCIFA Operation When Transmitting Data (2/2)

Figure 41.7 shows an example of Transmitting Serial Data in Asynchronous Mode.



[1] Confirm the status of the SCIFA and write the transmit data:
Read the FSR register, confirm that the TDFE flag is 1, then write transmit data to the FTDR register. The amount of data that can be write accesses is (16 – FDR.T[4:0]). The TXIF interrupt request can be used to determine if the TDFE flag has changed from 0 to 1.

[2] Procedure to continue serial transmission:
To continue serial transmission, confirm that the FSR.TDFE flag is 1, and then set it to 0. When using DMAC or DTC to write transmit data to the FTDR register, the TDFE flag automatically becomes 0, making this step unnecessary (do not manipulate the TDFE flag).

[3] Procedure to end serial transmission when using the TXIF interrupt or TEIF interrupt:
After writing the final transmit data to the FTDR register, set the SCR.TIE bit to 0 and set the SCR.TEIE bit to 1. When using the DMAC or DTC to write transmit data to the FTDR register, configure settings to enable the transfer end interrupt. When the last transmit data is written to the FTDR register, generate the CPU interrupt request, and rewrite the TIE and TEIE bit in the CPU interrupt handling. The procedure is not necessary when not using the TXIF interrupt request or TEIF interrupt request.

[4] Confirming serial transmission has ended:
Read the FSR register and confirm the TEND flag became 1. The TEIF interrupt request can be used to confirm that the TEND flag became 1.

[5] Outputting a break during serial transmission:
When sending a break during serial transmission, set the SPTR register with an immediate value of 0 for the SPB2DT bit and 1 for the SPB2IO bit before setting the SCR.TE bit to 0.*1

Note 1. The SPTR register cannot be rewritten in 1-bit units and requires 16-bit units instead.

Figure 41.7 Transmitting Serial Data in Asynchronous Mode

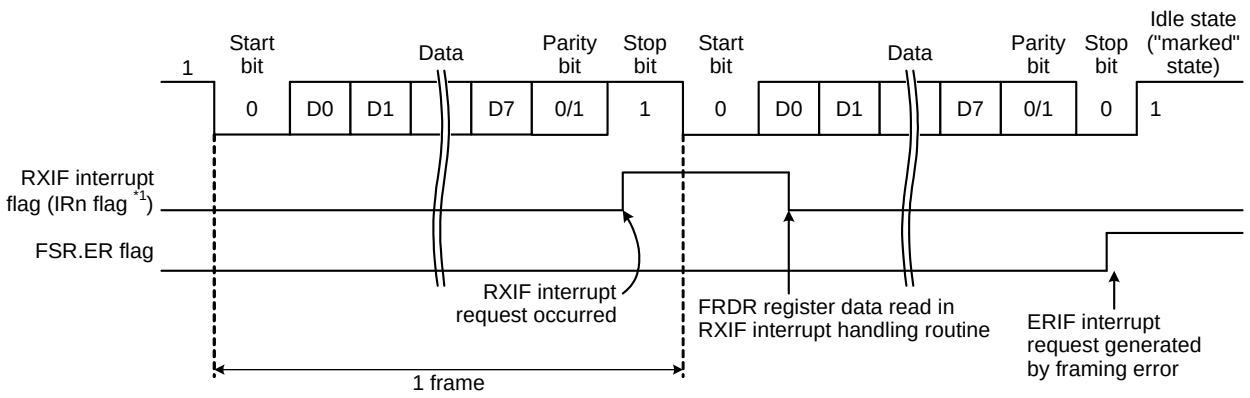
41.3.1.6 Receiving Serial Data in Asynchronous Mode

This section describes data reception in asynchronous mode.

1. When the SCIFA detects a start bit, reception starts, and the receive data is stored in the RSR register.
2. The SCIFA confirms the status of the flags in the FSR and LSR registers, confirms that a receive error (parity error, overrun error, or framing error) has not occurred, and also confirms that a break signal has not been detected. When these items have been confirmed, the SCIFA transfers the received data from the RSR register to the FRDR register.

Note: Received data is stored in the FRDR register even if a parity error or framing error occurs.

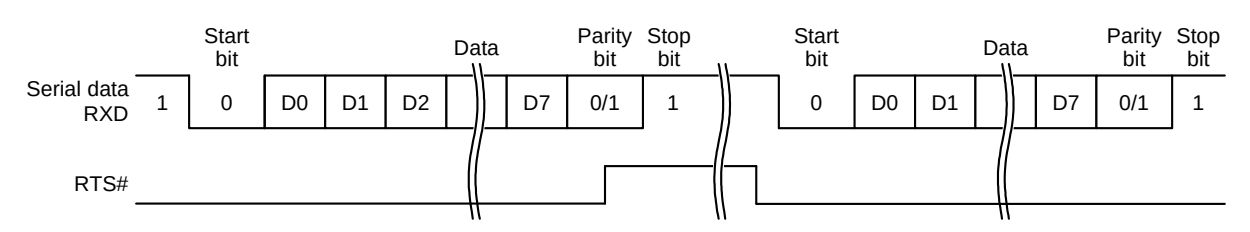
3. If the amount of data stored in the FRDR register is equal to or greater than the receive FIFO threshold value, the FSR.RDF flag becomes 1. At this time, if the SCR.RIE bit is 1, the SCIFA generates the RXIF interrupt request.
4. When the data stored in the FRDR register is less than the receive FIFO threshold value, if the next data is not received even after 15 ETUs elapse after the last stop bit in the received data, the FSR.DR flag becomes 1. At this time, if the SCR.RIE bit is 1, the SCIFA generates the DRIF interrupt request.
5. When the FSR.RE flag becomes 1, if the SCR.RIE bit or REIE bit is 1, the SCIFA generates the ERIF interrupt request. Also, when the FSR.BRK flag or LSR.ORER flag becomes 1, if the SCR.RIE bit or REIE bit is 1, the SCIFA generates the BRIF interrupt request.
6. When the FCR.MCE bit is 1 (hardware flow control enabled), if there is space in the FRDR register, the SCIFA outputs a low from the RTS# pin. When the RTS# pin is high, if the amount of data stored in the FRDR is equal to or greater than the FCR.RSTRG[2:0] bit setting, waiting for transmission is indicated. Refer to Figure 41.8 for details on SCIFA operation when not using hardware flow control, and refer to Figure 41.9 for details on SCIFA operation when using hardware flow control.



The above diagram assumes 8-bit data, parity, 1 stop bit, LSB first, and no hardware flow control.

Note 1: Refer to the section 15. Interrupt Controller (ICUA) for the corresponding interrupt number.

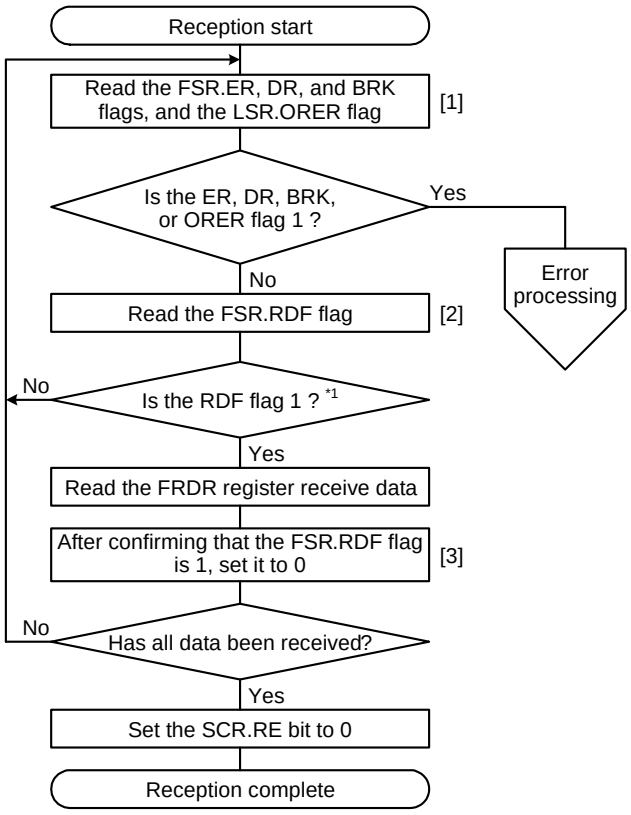
Figure 41.8 SCIFA Operation When Receiving Data (1/2)



The above diagram assumes 8-bit data, parity, 1 stop bit, LSB first, and hardware flow control.

Figure 41.9 SCIFA Operation When Receiving Data (2/2)

Figure 41.10 shows Receiving Serial Data in Asynchronous Mode (1/2).



[1] Receive error processing and break detection:
When a receive error occurs, read flags FSR.ER, DR, BRK, and the LSR.ORER flag to determine the type of error. After performing predetermined error processing, set the ER, DR, BRK, and ORER flags to 0. When a framing error occurs, the break can be detected by reading the RXD pin value. BRIF, ERIF, or DRIF interrupt requests indicate that the corresponding error has occurred.

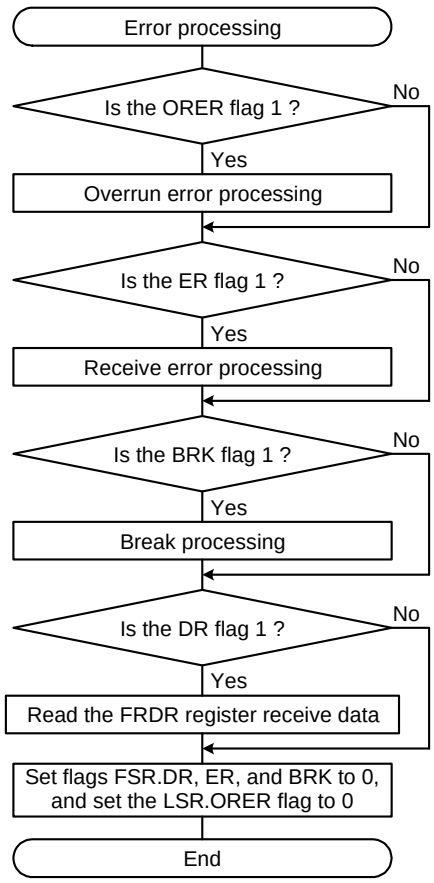
[2] Confirm the status of the SCIFA and read the receive data:
After reading the FSR register to confirm that the RDF flag is 1, read data stored in the FRDR register for at least the receive FIFO threshold value. The number of bytes of receive data that can be read can be confirmed by reading the FDR.R[4:0] flags. An RXIF interrupt request can be used to determine when the RDF flag changes from 0 to 1.

[3] Procedure to continue serial reception when using the CPU to read data from the FRDR register:
When continuing serial reception, after confirming that the FSR.RDF flag is 1, set it to 0. When triggering DMAC or DTC to read receive data from the FRDR register, the RDF flag automatically becomes 0, making this procedure unnecessary (do not manipulate the RDF flag).

Note 1. Set the amount of receive data to an integral multiple of the receive FIFO threshold value. When the amount of receive data is not set to an integral multiple of the receive FIFO threshold value, the program returns to step [1] and enters an endless loop, so timeout processing should be performed using an alternate timer, etc.

Figure 41.10 Receiving Serial Data in Asynchronous Mode (1/2)

Figure 41.11 shows Receiving Serial Data in Asynchronous Mode (2/2).



Note 1. Read the FSR.FER and PER flags to check if a framing error or parity error occurred in the receive data read from the FRDR register.

Note 2. When a break signal is received, if the BRK flag is 1, the receive data is not transferred to the FRDR register. Note that the last data in the FRDR register is 00h, and break data that includes a framing error is stored to the FRDR register.

Figure 41.11 Receiving Serial Data in Asynchronous Mode (2/2)

41.3.2 SCIFA Operation in Clock Synchronous Mode

Figure 41.12 shows a general format clock of the synchronous serial communication. In clock synchronous mode, data is transmitted and received during the period with the clock pulse. When transmitting data in clock synchronous mode, the SCIFA outputs data during the period from one falling edge to the next falling edge of the synchronous clock. After transmitting the 8th bit, the SCIFA holds the communication line at the value of the last bit output. When receiving data in clock synchronous mode, the SCIFA receives data in synchronization with the rising edge. The SCIFA's transmitter and receiver operate independently meaning full duplex communication is possible over the shared synchronous clock. Also, the transmit buffer and receive buffer are configured with a 16-stage FIFO buffer, meaning data can be read and written during transmission/reception, and continuous transmission/reception is possible.

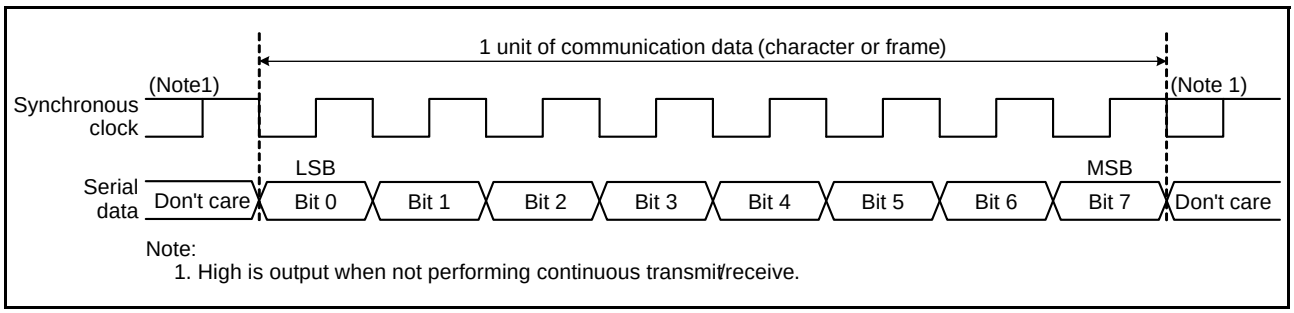


Figure 41.12 Data Format When Using Clock Synchronous Communication With LSB First

41.3.2.1 Format of Transmit/Receive Data in Clock Synchronous Mode

In clock synchronous mode, only 8 bits can be transmitted or received in 1 frame of data. The parity bit cannot be added.

41.3.2.2 Clocks

While in clock synchronous mode, the clock source is selectable from the internal clock generated by the baud rate generator, or the external clock input from the SCK pin.
Refer to Table 41.18 for information on the clock source and register setting values while in clock synchronous mode. The synchronous clock outputs eight pulses in one frame of data transmission and reception, and the synchronous clock is fixed high when data is not being transmitted or received. During a receive operation only, if the internal clock is selected as the clock source, while the SCR.RE bit is 1, a clock pulse is output until the amount of data in the receive FIFO reaches the receive FIFO threshold value.

Table 41.18 Clock Source and Register Settings in Clock Synchronous Mode

SMR	SCR		SPTR		Clock Source	SCK Pin Function
CM	CKE[1]	CKE[0]	SCKIO	SCKDT		
1	0	0 or 1	0 or 1	0 or 1	Internal	Outputs a synchronous clock
	1	0	0 or 1	0 or 1	External	Inputs a synchronous clock
		1	0 or 1	0 or 1	Do not set this value.	

41.3.2.3 SCIFA Initialization in Clock Synchronous Mode

Figure 41.13 shows SCIFA Initialization in Clock Synchronous Mode.

Before starting data transmission and reception, set bits SCR.TE and RE to 0, and initialize the SCIFA using the procedure in the figure below. When the SCR.TE bit is set to 0, the TSR register is initialized. However, even if the SCR.RE bit is set to 0, the FSR.RDF flag, LSR.ORER flag, and FRDR register are not initialized, and their values are retained. To continue data reception, the FSR.RDF flag, LSR.ORER flag, and FRDR register must be initialized.

If the SCIFA is not initialized as shown in the following figure, and the operating mode is changed from asynchronous mode to clock synchronous mode, confirm that the ORER, PER, and FER flags are set to 0.

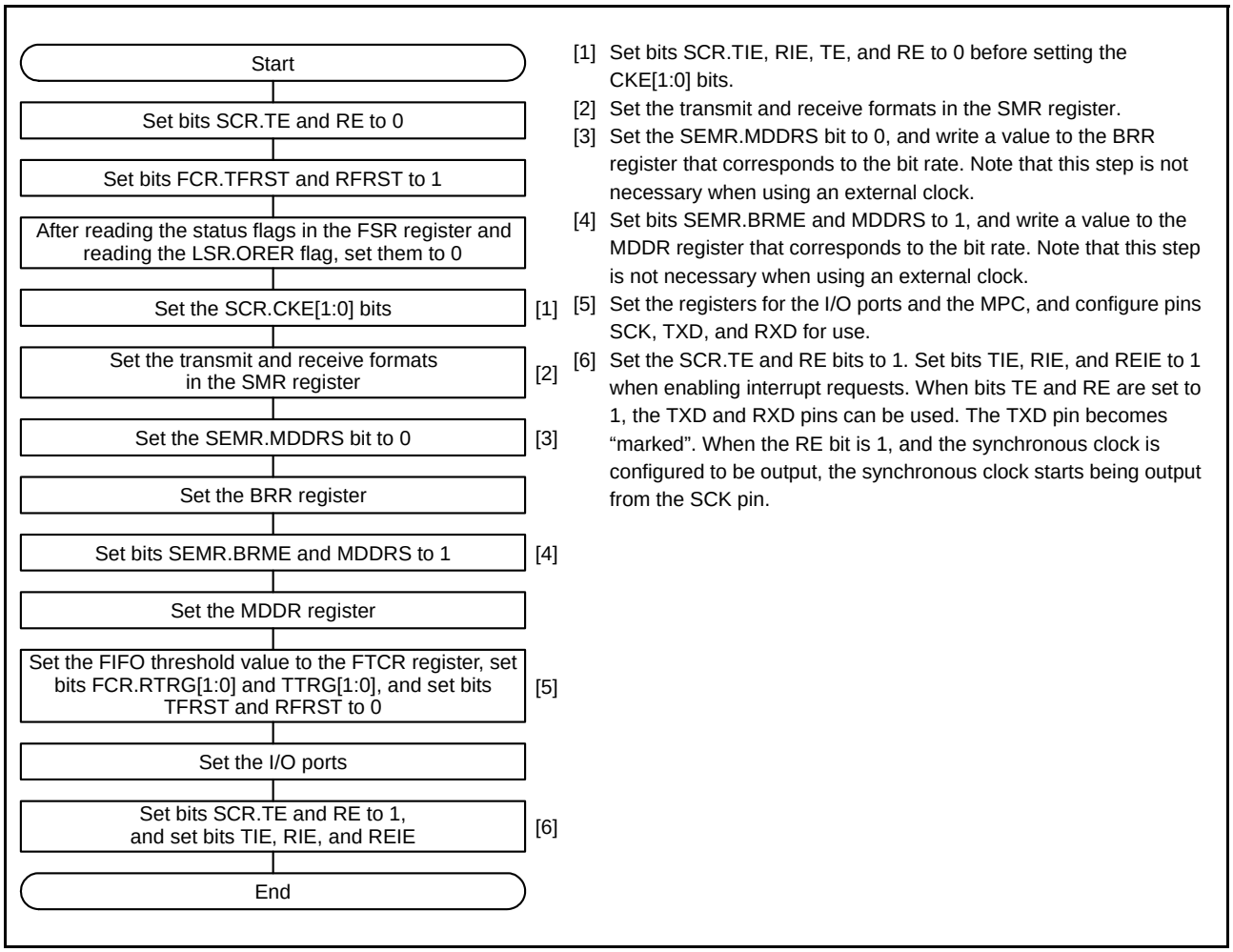


Figure 41.13 SCIFA Initialization in Clock Synchronous Mode

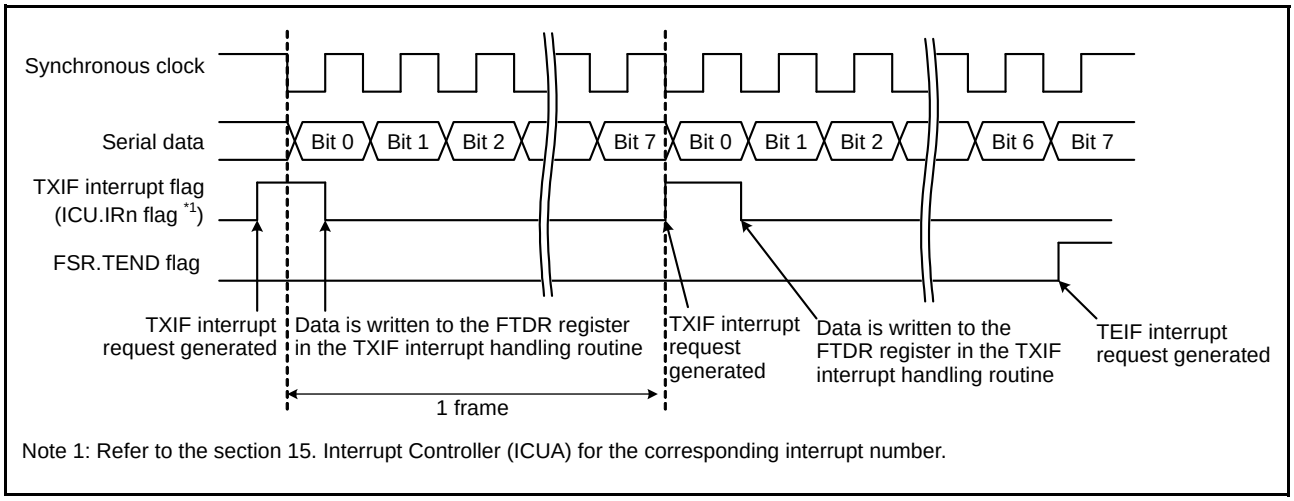
41.3.2.4 Transmitting Serial Data in Clock Synchronous Mode

This sections describes transmitting serial data in clock synchronous mode.

1. When data is written to the FTDR register, the SCIFA transfers data from the FTDR register to the TSR register. Confirm that the FSR.TDFE flag is 1 before writing transmit data to the FTDR register. The amount of transmit data that can be written to the FTDR register is calculated as (16 - FDR.T[4:0]).
2. When the data transfer from the FTDR register to the TSR register starts, the SCIFA continues transmitting data until data written to the FTDR register is gone. If the amount of data written to the FTDR register is less than or equal to the transmit FIFO threshold value *1, the FSR.TDFE flag becomes 1. At this point, if the SCR.TIE bit is 1, the SCIFA generates the TXIF interrupt request.
3. When a clock is output from the SCK pin, the SCIFA outputs eight pulses of the synchronous clock. When a clock is input to the SCK pin, the SCIFA outputs data synchronous to the input clock.
4. The SCIFA checks whether there is any untransmitted data in the FTDR register when the last bit is transmitted. If there is untransmitted data in the FTDR register, the data is transferred to the TSR register, and transmission of the next frame of data starts. If there is no untransmitted data in the FTDR register, the FSR.TEND flag becomes 1, and after the last bit is transmitted, the TXD pin retains the output level of the last bit. When transmission is complete, the SCK pin becomes “marked”.

Note 1. When the FTCR.TTRGS bit is 0, the transmit FIFO threshold value is determined by the FCR.TTRG[1:0] bits; when the FTCR.TTRGS bit is 1, the transmit FIFO threshold value is determined by the FTCR.TFTC[4:0] bits.

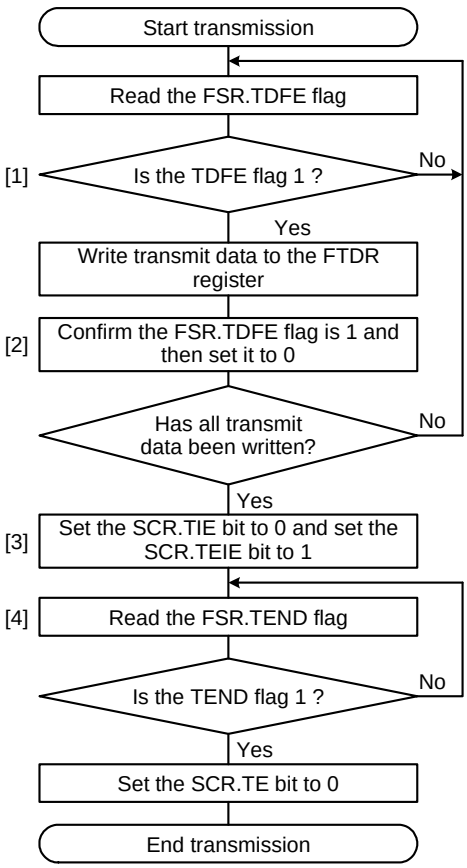
Figure 41.14 shows SCIFA Operation When Transmitting Data With LSB First.



Note 1: Refer to the section 15. Interrupt Controller (ICUA) for the corresponding interrupt number.

Figure 41.14 SCIFA Operation When Transmitting Data With LSB First

Figure 41.15 shows an example of Transmitting Serial Data in Clock Synchronous Mode.



[1] Confirm the status of the SCIFA and write transmit data:
Read the FSR register, confirm that the TDFE flag is 1, and then write the transmit data to the FTDR register. The amount of transmit data that can be written to the FTDR is calculated by (16 - FDR.T[4:0]). The TXIF interrupt request can be used to confirm that the TDFE flag changed from 0 to 1.

[2] Procedure to continue serial transmission when using the CPU to write to the FTDR register:
When continuing serial transmission, confirm that the FSR.TDFE flag is 1 and then set it to 0. When the DMAC or DTC is triggered to write transmit data to the FTDR register, the TDFE flag automatically becomes 0, making this step unnecessary (do not manipulate the TDFE flag).

[3] Procedure to end serial transmission when using the TXIF interrupt or TEIF interrupt:
When the last byte of transmit data is written to the FTDR register, set the SCR.TIE bit to 0 and set the SCR.TEIE bit to 1. When the DMAC or DTC is triggered to write transmit data to the FTDR register, enable the transfer end interrupt, and when the last byte of transmit data is written to the FTDR register, generate the CPU interrupt request, and rewrite bits TIE and TEIE in the CPU interrupt handling. This step is not necessary when the TXIF interrupt request and TEIF interrupt request are not used.

[4] Confirming serial transmission is complete:
Read the FSR register and confirm that the TEND flag is 1. The TEIF interrupt request can be used to confirm that the TEND flag is 1.

In step [1], the FDR.T[4:0] flags (amount of untransmitted data in the FTDR register) can be used to check the number of bytes of data that can be written to the FTDR register.

Figure 41.15 Transmitting Serial Data in Clock Synchronous Mode

41.3.2.5 Receiving Serial Data in Clock Synchronous Mode

Figure 41.16 shows an example of receiving data in clock synchronous mode.

1. The SCIFA synchronizes with the input or output of the synchronous clock, and writes receive data to the RSR register.
2. The SCIFA confirms the value of the LSR.ORER flag is 0, and confirms that the FRDR register has space available. After those items have been confirmed, the SCIFA sets the FSR.RDF flag to 1, and transfers receive data from the RSR register to the FRDR register.
3. When the amount of receive data in the FRDR register is equal to or greater than the receive FIFO threshold value, the FSR.RDF flag becomes 1. At this time, if the SCR.RIE bit is 1, the SCIFA generates the RXIF interrupt request. Also, when the LSR.ORER flag becomes 1, if the SCR.RIE bit or REIE bit is set to 1, the SCIFA generates the BRIF interrupt request.

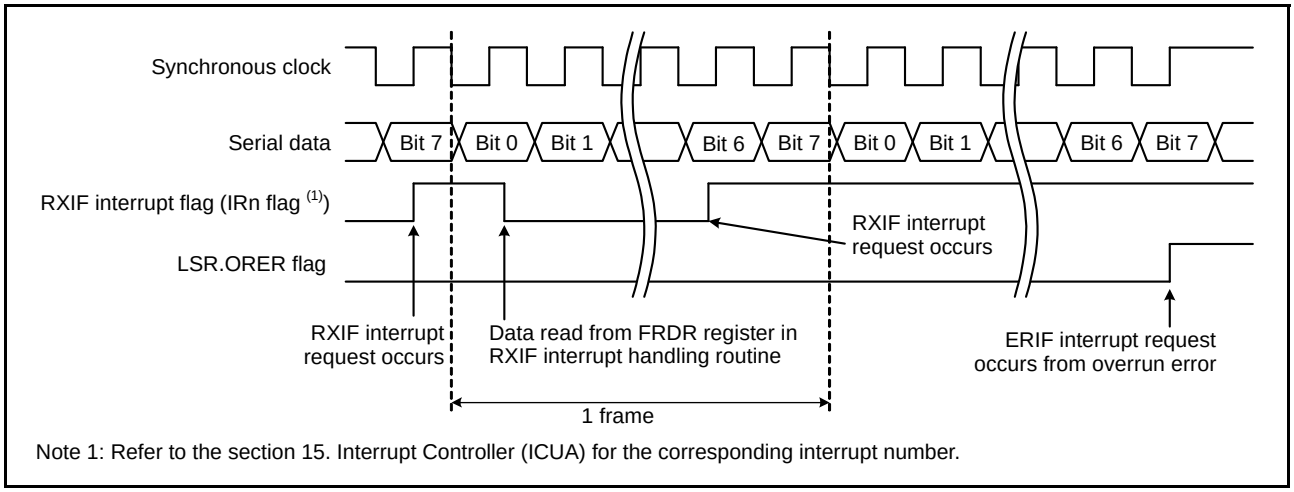
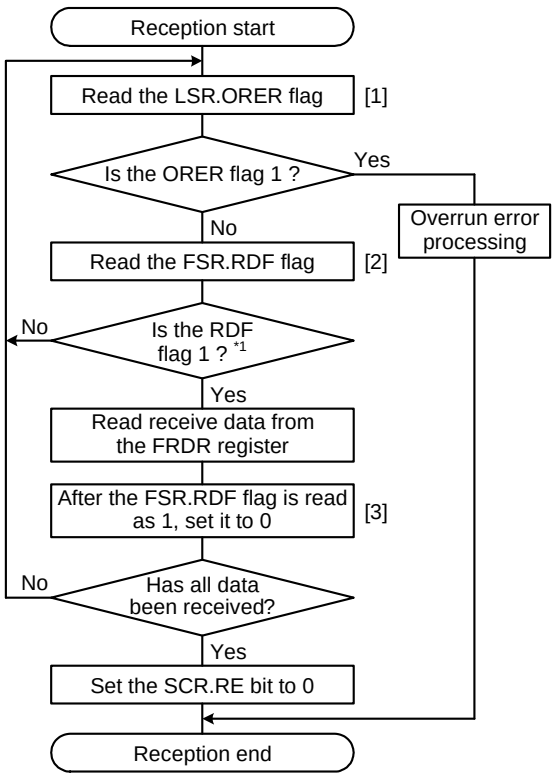


Figure 41.16 SCIFA Operation When Receiving Data With the LSB First

Figure 41.17 shows an example of Receiving Serial Data in Clock Synchronous Mode.



[1] Receive error processing:
When a receive error occurs, read the LSR.ORER flag, after performing predetermined error processing, set the ORER flag to 0. Data reception cannot be restarted while the ORER flag is 1. The BRIF interrupt request can be used to confirm that the ORER flag became 1.

[2] Confirm the status of SCIFA and read receive data:
Read the FSR register, confirm that the RDF flag is 1, and then read at least the receive FIFO threshold value of receive data stored in the FRDR register. Read the FDR.R[4:0] flags to confirm how much receive data can be read. The RXIF interrupt request can be used to confirm that the RDF flag changed from 0 to 1.

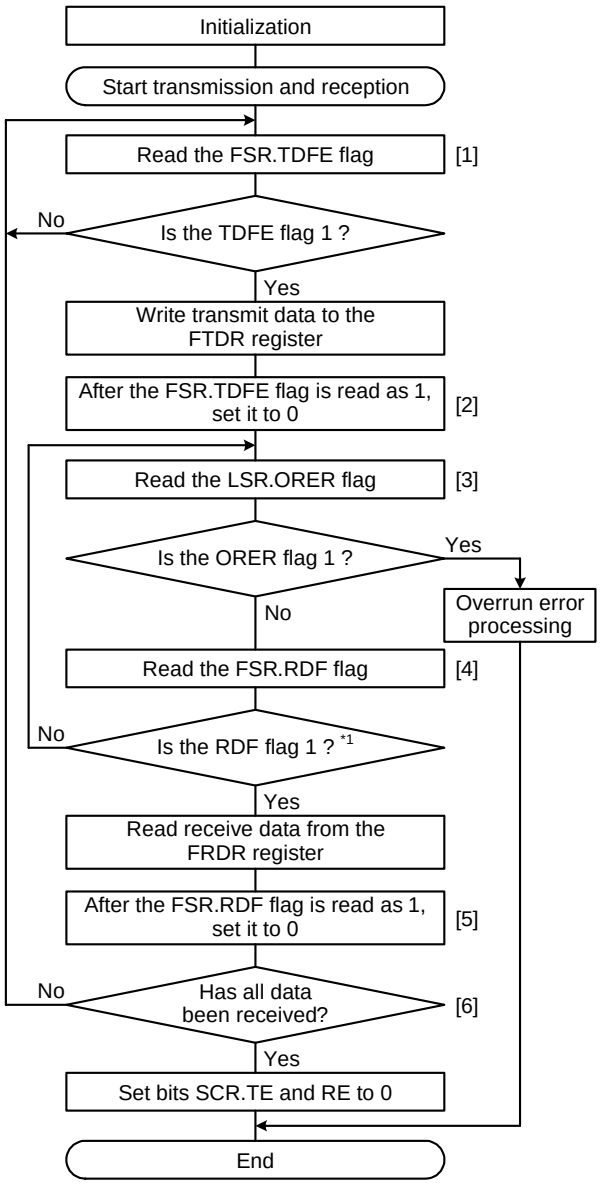
[3] Procedure to continue serial reception when using the CPU to read from the FRDR register:
When continuing serial reception, confirm that the FSR.RDF flag is 1 and then set it to 0. When the DMAC or DTC are triggered to read receive data from the FRDR register, the RDF flag automatically becomes 0, making this step unnecessary (do not rewrite the RDF flag).

Note 1. Set the amount of receive data to an integral multiple of the receive FIFO threshold value. When the amount of receive data is not set to an integral multiple of the receive FIFO threshold value, the program returns to step [1] and enters an endless loop, so timeout processing should be performed using a timer, etc.

Figure 41.17 Receiving Serial Data in Clock Synchronous Mode

41.3.2.6 Simultaneously Transmitting and Receiving Serial Data in Clock Synchronous Mode

Figure 41.18 shows and example of Simultaneously Transmitting and Receiving Serial Data in Clock Synchronous Mode. When simultaneously transmitting and receiving serial data in clock synchronous mode, enable transmission and reception, and then perform the procedure in the figure below.



[1] Confirm the status of the SCIFA and write transmit data:
Read the FSR register, confirm that the TDFE flag is 1, and then write the transmit data to the FTDR register. The TXIF interrupt request can be used to confirm that the TDFE flag changed from 0 to 1.

[2] Procedure to continue serial transmission when using the CPU to write to the FTDR register:
When continuing serial transmission, confirm that the FSR.TDFE flag is 1 and then set it to 0. When the DMAC or DTC is triggered to write transmit data to the FTDR register, the TDFE flag automatically becomes 0, making this step unnecessary (do not manipulate the TDFE flag).

[3] Receive error processing:
When receive error occurs, read the LSR.ORER flag, and after performing predetermined error processing, set the ORER flag to 0. Data reception cannot be restarted while the ORER flag is 1. The BRIF interrupt request can be used to confirm that the ORER flag became 1.

[4] Confirm the status of the SCIFA and read receive data:
Read the FSR register, confirm that the RDF flag is 1, and then read the receive data from the FRDR register. The RXIF interrupt request can be used to confirm that the RDF flag changed from 0 to 1.

[5] Procedure to continue serial reception when using the CPU to read from the FRDR register:
When continuing serial reception, confirm that the FSR.RDF flag is 1 and then set it to 0. When the DMAC or DTC is triggered to read receive data from the FRDR register, the RDF flag automatically becomes 0, making this step unnecessary (do not manipulate the RDF flag).

[6] Procedure to continue serial transmission and reception (supplement):
When continuing serial transmission and reception, read the FRDR register before the FRDR register overflows and the ORER flag becomes 1, and set the RDF flag to 0. Also, before transmitting the last bit of the last data written to the FTDR register, write data to the FTDR register, and set the TDFE flag to 0. Furthermore, when the DMAC or DTC is triggered to write data to the FTDR register or read data from the FRDR register, the TDFE and RDF flags automatically become 0, so they do not need to be cleared manually (do not manipulate the TDFE or RDF flag).

Note: When toggling from a transmit or receive operation to simultaneous transmission and reception, set bits TE and RE to 0, and then set them to 1 simultaneously.

Note 1. Set the amount of receive data to an integral multiple of the receive FIFO threshold value. When the amount of receive data is not set to an integral multiple of the receive FIFO threshold value, the program returns to step [1] and enters an endless loop, so timeout processing should be performed using a timer, etc.

Figure 41.18 Simultaneously Transmitting and Receiving Serial Data in Clock Synchronous Mode

41.3.3 Bit Rate Modulation Function

The bit rate modulation function corrects the bit rate by thinning out the specified amount of clocks from those input to the baud rate generator.

When the SEMR.BRME bit is 1, the baud rate generator validates and counts the average interval of the number of clocks set in the MDDR register out of the total 256 clocks input.

Figure 41.19 assumes the SCIFA is in asynchronous mode, bits SMR.CKS[1:0] are 00b, the BRR register is 00h, and the MDDR register is 160. In this example, the cycle of the base clock is evenly corrected to $\frac{256}{160}$, and the bit rate is corrected to $\frac{160}{256}$. Note that there is an imbalance in thinning out the internal clock, and expansion and contraction occur in the pulse width of the internal base clock.

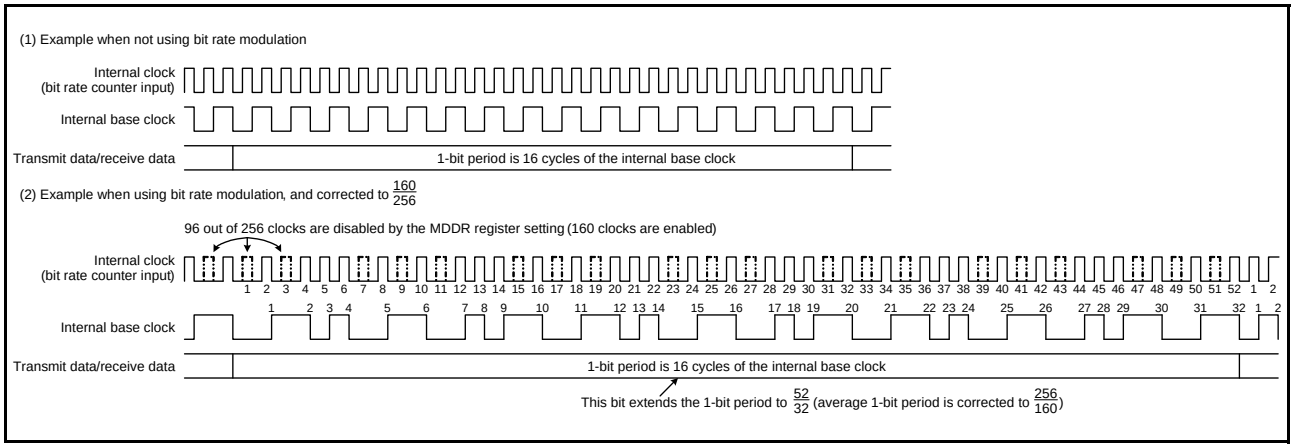


Figure 41.19 Example of the Base Clock When the Bit Modulation Function is Used

41.3.4 Relation Between the SPTR Register and SCIFA Pins

Figure 41.20 to Figure 41.23 show the relation between the SPTR register and the SCIFA pins.

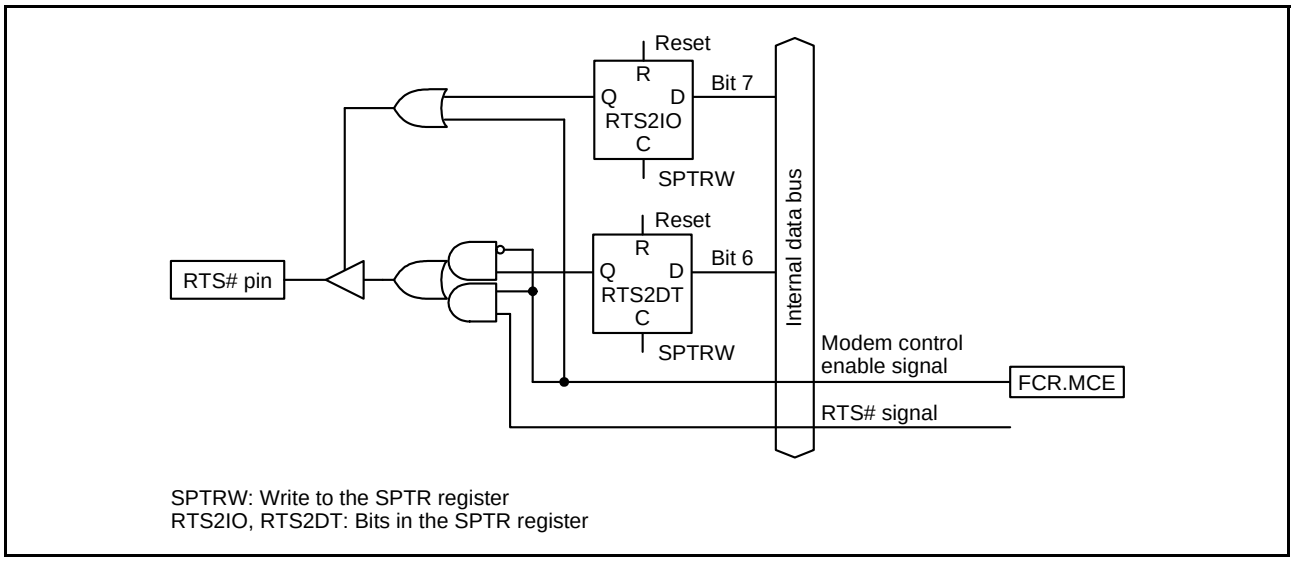


Figure 41.20 Relation Between the RTS# Pin and Bits RTS2IO and RTS2DT

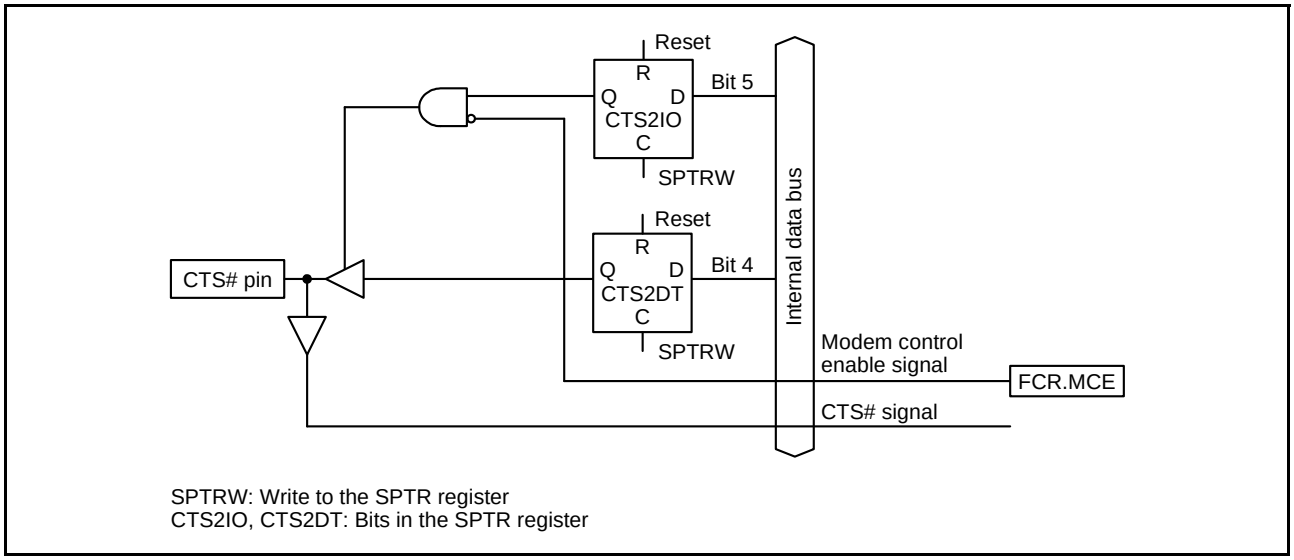


Figure 41.21 Relation Between the CTS# Pin and Bits CTS2IO and CTS2DT

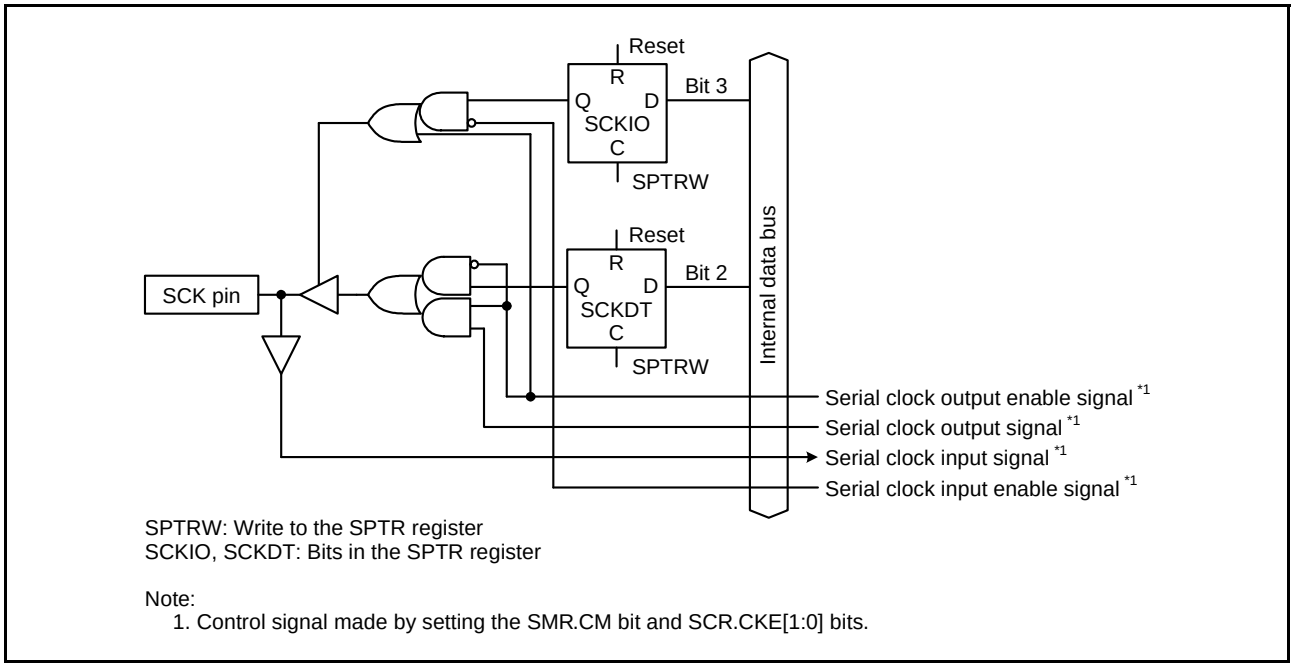


Figure 41.22 Relation Between the SCK Pin and Bits SCKIO and SCKDT

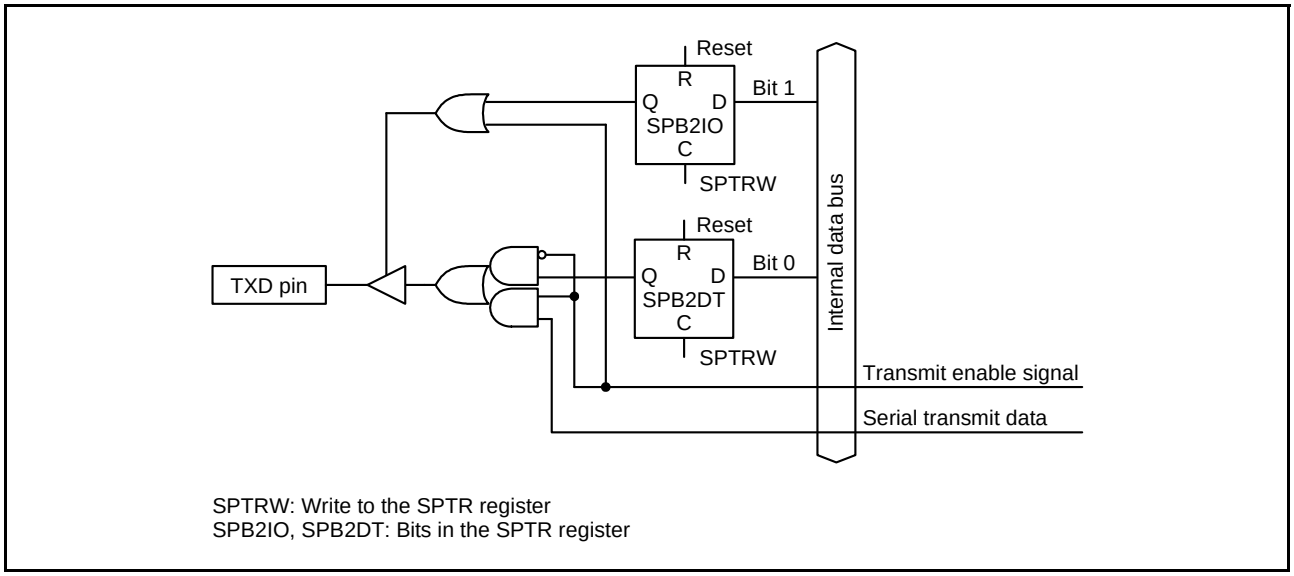


Figure 41.23 Relation Between the TXD Pin and Bits SPB2IO and SPB2DT

41.3.5 Noise Cancellation

Figure 41.24 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of a two-stage flip-flop circuit and a match detection circuit. When the input signals of the noise filter and the output signals of the two-stage flip-flop circuit all match, that level is conveyed as an internal signal. If the levels do not match, the previous value is retained. When the same level is retained for three or more cycles of the sampling clock for the noise filter, the SCIFA determines it to be a valid receive signal. When there is change in the signal level less than three cycles, the SCIFA determines the signal to be noise and not a receive signal.

In asynchronous mode, the noise cancellation can be applied to the RXD pin. The receive signal of the RXD pin is taken in to the flip-flop circuit of the noise filter on the clock with a frequency eight times or 16 times the bit rate *1.

When the noise filter is enabled, if the base clock is stopped and then restarted, noise filter operation resumes from the state where the base clock was stopped. If the SCR.RE bit is set to 0 while the base clock is being input, the noise filter sets the RXD internal signal to 0. The match detection circuit continues to operate even when reception is stopped, the three previous sampling results are output at the same time reception resumes.

Note 1. When the SEMR.ABCS0 bit and SEMR.BGDM bit are both 0, the frequency is 16 times the bit rate, and when either the SEMR.ABCS0 bit and SEMR.BGDM bit is 1, the frequency is eight times the bit rate.

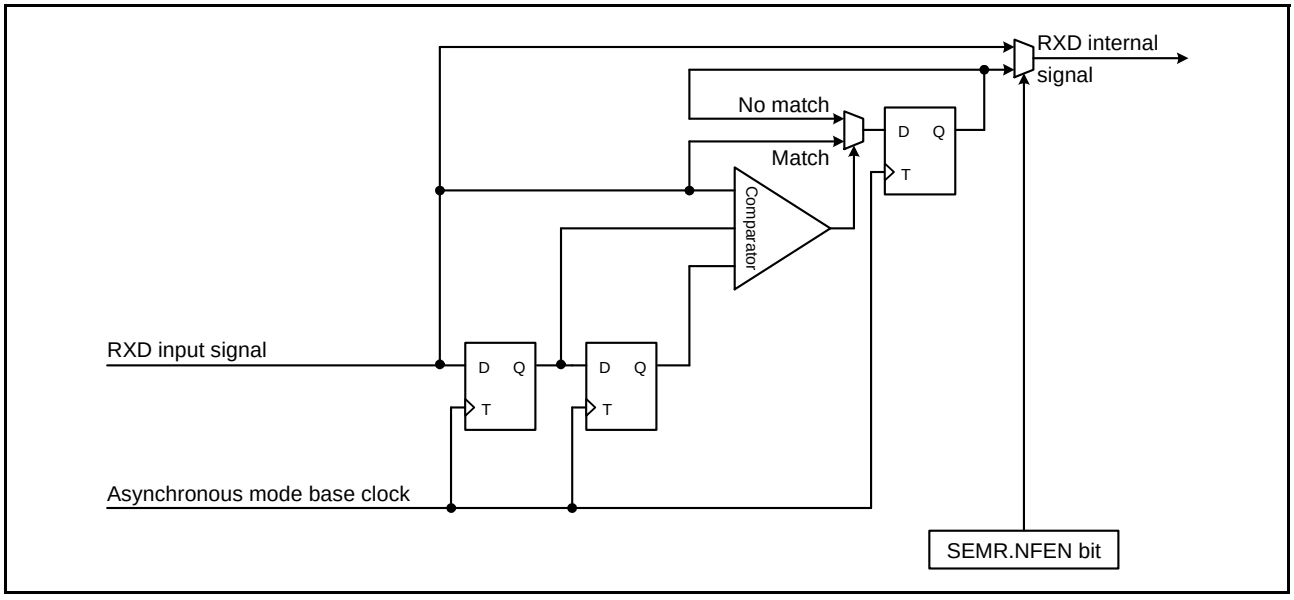


Figure 41.24 Block Diagram of the Digital Noise Filter Circuit

41.4 Interrupts

The SCIFA has six interrupt sources as listed in Table 41.19.
The interrupt sources can be enabled or disabled using bits in Table 41.19. Interrupt sources are individually input to the interrupt controller.
When transmitting data, if the amount of data written to the FTDR register is equal to or less than the transmit FIFO threshold value, the FSR.TDFE flag becomes 1, and the TXIF interrupt request is generated. When receiving data, if the amount of data stored in the FRDR is greater than or equal to the receive FIFO threshold value, the FSR.RDF flag becomes 1, and the RXIF interrupt request is generated.
While in asynchronous mode, if the amount of data stored in the FRDR register is less than the receive FIFO threshold value, and if the next data is not received even after 15 ETUs elapse *1 after the last stop bit, when the FSR.DR flag becomes 1, the SCIFA generates the DRIF interrupt request. While in clock synchronous mode, the DRIF interrupt request is not generated.
The SCIFA generates the BRIF interrupt request when the FSR.BRK flag and LSR.ORER flag become 1, it generates the ERIF interrupt request when the ER flag becomes 1, and it generates the TEIF interrupt request when the TEND flag becomes 1.
When setting the SCR.RIE bit to 0 and the REIE bit to 1, the ERIF interrupt request and BRIF interrupt request can be generated without generating the RXIF interrupt request.
The TXIF interrupt indicates that transmit data can be written to the FTDR register, and the RXIF interrupt indicates that there is receive data in the FRDR register.

Note 1. Equivalent to 1.5 frames of 8-bit data with a 1 stop bit format.

Table 41.19 SCIFA Interrupt Sources

Interrupt Source	Description	Interrupt Enable Bit	DMAC/DTC Triggerable	Priority Level
BRIF	Interrupt caused by break (BRK) or overrun (ORER)	RIE or REIE	No	High
ERIF	Interrupt caused by a framing error or parity error (ER)	RIE or REIE	No	↕
RXIF	Interrupt caused by receive FIFO data full (RDF)	RIE	Yes	
TXIF	Interrupt caused by transmit FIFO data empty (TDFE)	TIE	Yes	
TEIF	Interrupt caused by transmit end (TEND)	TEIE	No	
DRIF	Interrupt caused by receive data ready (DR)	RIE	No	Low

Note: When performing CPU processing, flags are cleared after blocks are transferred. When triggering the DTC or DMAC, flag access is disabled.

41.5 Notes on Using the SCIFA

41.5.1 Relation Between the FTDR Register and the FSR.TDFE Flag

When the amount of data written to the FTDR register is equal to or less than the transmit FIFO threshold value, if the FSR.TDFE flag is read as 1 and then set to 0, it becomes 1 again. In order to set the FSR.TDFE flag to 0, after the amount of data written to the FTDR register surpasses the transmit FIFO threshold value, confirm that the flag is 1 and then set it to 0. When using DTC or DMAC to write data to the FTDR register, when the DTC or DMA transfer is complete, the FSR.TDFE flag becomes 0. The amount of transmit data stored in the FTDR register can be checked by reading the FSR.T[4:0] flags.

41.5.2 Relation Between the FRDR Register and the FSR.RDF Flag

When the amount of data written to the FRDR register is equal to or more than the receive FIFO threshold value, if the FSR.RDF flag is read as 1 and then set to 0, it becomes 1 again. In order to set the FSR.RDF flag to 0, after the amount of data read from the FRDR register becomes less than the receive FIFO threshold value, confirm that the flag is 1 and then set it to 0. When using the DTC or DMAC to read data from the FRDR register, when the DTC or DMA transfer is complete, the FSR.RDF flag becomes 0. The amount of receive data stored in the FRDR register can be checked by reading the FDR.R[4:0] flags.

41.5.3 Detecting a Break Signal and Processing

Break signals are automatically detected by hardware, but a break signal can also be detected by directly reading the RXD pin value when a framing error is detected. In a break, all input from the RXD pin becomes low, the FER flag becomes 1, and the PER flag may become 1. If a break signal is detected, transfer of receive data to the FRDR register stops, but the SCIFA continues to monitor the RXD pin. Reception starts again when the break is released (when the RXD pin becomes high).

41.5.4 Outputting a Break Signal

The signal output from the TXD pin is dependent on the settings of the SCR.TE bit, SPTR.SPB2IO bit, and SPTR.SPB2DT bit. These bits can be set to output a break signal. Between initializing the SCIFA and setting the SCR.TE bit to 1, the TXD pin does not function as a transmit data output pin. Between initializing the SCIFA and setting the SCR.TE bit to 1, if the SPB2IO and SPB2DT bits are set to 1, output from the TXD pin can be configured as “marked” (high). To output a break signal during data transmission, after setting the SPB2IO bit to 1 and the SPB2DT bit to 0, set the SCR.TE bit to 0. When the SCR.TE bit is set to 0, irrelevant to the current transmission status, the transmitter is initialized, and the TXD pin outputs a low depending on the setting of the SPB2DT bit.

41.5.5 Note on the FSR.FER and PER Flags

The status indicated by the FSR.FER and PER flags is the status of the data in the FRDR register to be read next. If the CPU, DMAC, or DTC is used to read the FRDR register, the framing error and parity error information in the relevant data disappears. To check the framing error and parity error in the receive data, read the FSR register before reading the FRDR register.

41.5.6 Note on Inputting an External Clock in Clock Synchronous Mode

To set bits SCR.TE and RE to 1, wait at least four cycles of the PCLKA after the external clock (SCK) changes from low to high, and then set the pins. In order to input the external clock (start communication), set bits SCR.TE and RE to 1, and then wait for at least one cycle of the external clock before inputting the external clock.

41.5.7 Module-Stop Function Setting

SCIFA operation can be enabled or disabled by setting bits MSTPCRC.MSTPC24 (SCIFA11), MSTPC25 (SCIFA10), MSTPC26 (SCIFA9), and MSTPC27 (SCIFA8). After a reset, SCIFA operation is stopped. Registers in the SCIFA can be accessed by exiting the module-stop state. Refer to section 11, Low Power Consumption for details.

42. I2C-bus Interface (RIICa)

This MCU has a two-channel I2C-bus interface (RIIC0, RIIC2).
The RIIC module conforms with the NXP I2C-bus (Inter-IC bus) interface and provides a subset of its functions.
In this section, “PCLK” is used to refer to PCLKB.

42.1 Overview

Table 42.1 lists the specifications of the RIIC, Figure 42.1 shows a block diagram of the RIIC, and Figure 42.2 shows an example of I/O pin connections to external circuits (I2C-bus configuration example). Table 42.2 lists the I/O pins of the RIIC.

Table 42.1 RIIC Specifications (1/2)

Item	Description
Communications format	• I2C-bus format or SMBus format • Master mode or slave mode selectable • Automatic securing of the various setup times, hold times, and bus-free times for the transfer rate
Transfer rate	Fast-mode Plus is supported (up to 1 Mbps)
SCL clock	For master operation, the duty cycle of the SCL clock is selectable in the range from 4 to 96%.
Issuing and detecting conditions	Start, restart, and stop conditions are automatically generated. Start conditions (including restart conditions) and stop conditions are detectable.
Slave address	• Up to three different slave addresses can be set. • 7-bit and 10-bit address formats are supported (along with the use of both at once). • General call addresses, device ID addresses, and SMBus host addresses are detectable.
Acknowledgment	• For transmission, the acknowledge bit is automatically loaded. Transfer of the next data for transmission can be automatically suspended on detection of a not-acknowledge bit. • For reception, the acknowledge bit is automatically transmitted. If a wait between the eighth and ninth clock cycles has been selected, software control of the value in the acknowledge field in response to the received value is possible.
Wait function	• In reception, the following periods of waiting can be obtained by holding the SCL clock at the low level: Waiting between the eighth and ninth clock cycles Waiting between the ninth clock cycle and the first clock cycle of the next transfer
SDA output delay function	Timing of the output of transmitted data, including the acknowledge bit, can be delayed.
Arbitration	• For multi-master operation Operation to synchronize the SCL clock in cases of conflict with the SCL signal from another master is possible. When issuing the start condition would create conflict on the bus, loss of arbitration is detected by testing for non-matching between the internal signal for the SDA line and the level on the SDA line. In master operation, loss of arbitration is detected by testing for non-matching between the signal on the SDA line and the internal signal for the SDA line. • Loss of arbitration due to detection of the start condition while the bus is busy is detectable (to prevent the issuing of double start conditions). • Loss of arbitration in transfer of a not-acknowledge bit due to the internal signal for the SDA line and the level on the SDA line not matching is detectable. • Loss of arbitration due to non-matching of internal and line levels for data is detectable in slave transmission.
Timeout function	The internal timeout function is capable of detecting long-interval stop of the SCL clock.
Noise cancellation	The interface incorporates digital noise filters for both the SCL and SDA signals, and the width for noise cancellation by the filters is adjustable by software.
Interrupt sources	Four sources: • Error in transfer or occurrence of events Detection of arbitration, NACK, timeout, a start condition including a restart condition, or a stop condition • Receive data full (including matching with a slave address) • Transmit data empty (including matching with a slave address) • Transmit end

Table 42.1 RIIC Specifications (2/2)

Item	Description
Low power consumption function	Module stop state can be set.
RIIC operating modes	• Four Master transmit mode, master receive mode, slave transmit mode, and slave receive mode
Event link function (output)	• Four sources (RIIC0): Error in transfer or occurrence of events Detection of arbitration, NACK, timeout, a start condition including a restart condition, or a stop condition • Receive data full (including matching with a slave address) • Transmit data empty (including matching with a slave address) • Transmit end

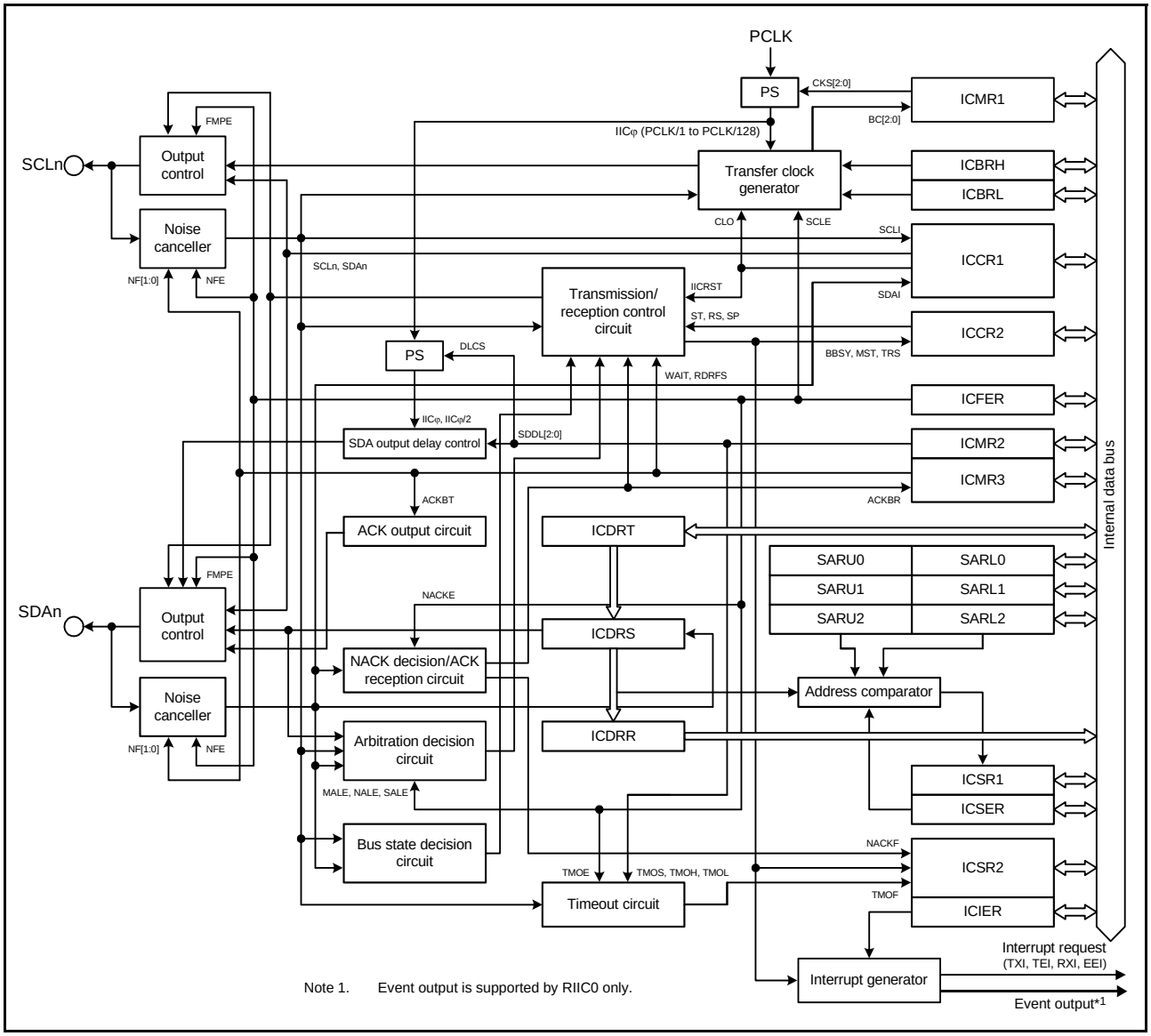


Figure 42.1 RIIC Block Diagram (n=0, 2)

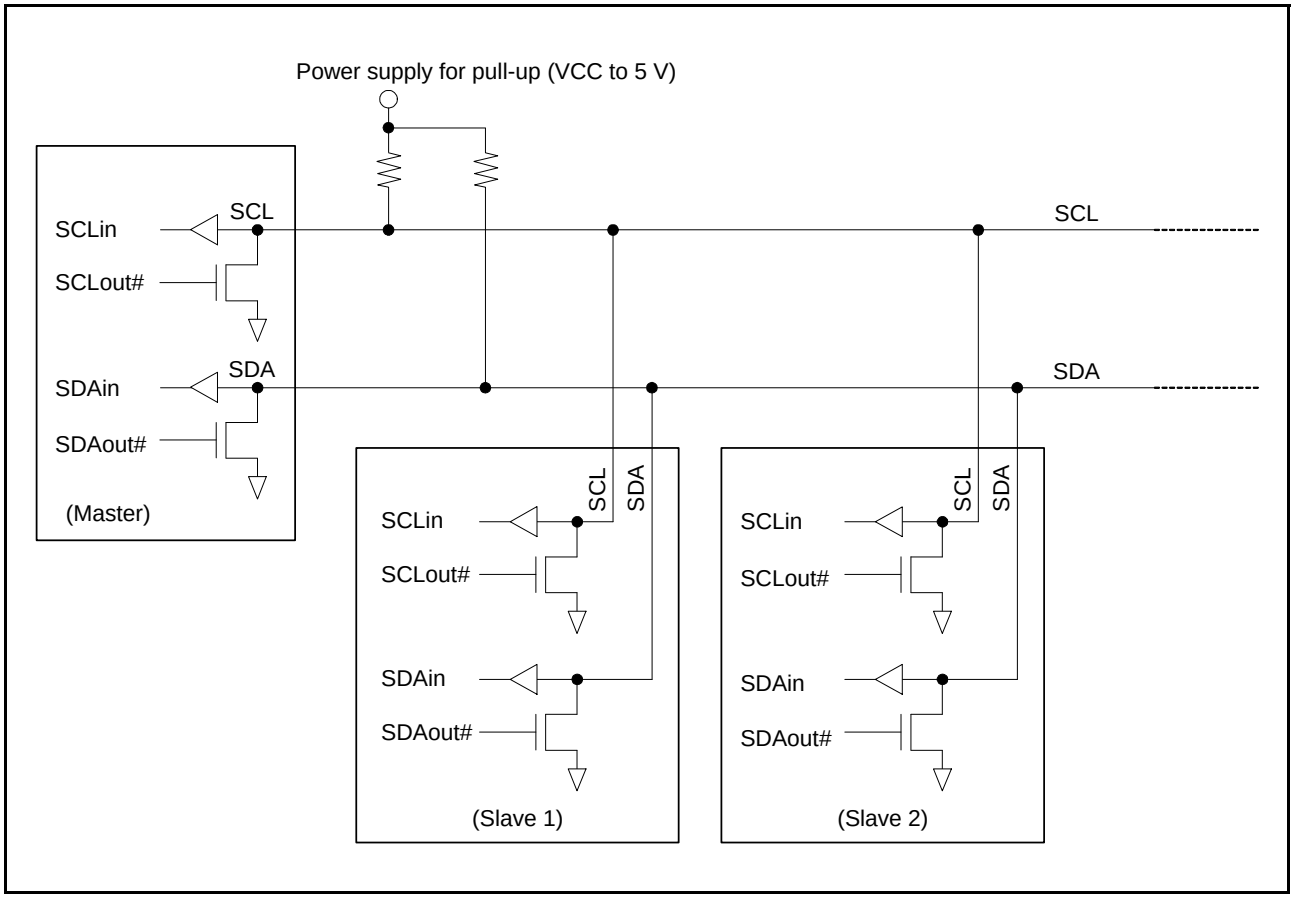


Figure 42.2 I/O Pin Connection to the External Circuit (I2C-bus Configuration Example)

The input level of the signals for RIIC is CMOS when I2C-bus is selected (ICMR3.SMBS bit is 0), or TTL when SMBus is selected (ICMR3.SMBS bit is 1).

Table 42.2 RIIC Pin Configuration

Channel	Pin Name	I/O	Function
RIIC0	SCL0	I/O	RIIC0 serial clock I/O pin
	SDA0	I/O	RIIC0 serial data I/O pin
RIIC2	SCL2	I/O	RIIC2 serial clock I/O pin
	SDA2	I/O	RIIC2 serial data I/O pin

42.2 Register Descriptions

42.2.1 I2C-bus Control Register 1 (ICCR1)

Address(es): RIIC0.ICCR1 0008 8300h, RIIC2.ICCR1 0008 8340h

	b7	b6	b5	b4	b3	b2	b1	b0
	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SDAI	SDA Line Monitor	0: SDAn line is low. 1: SDAn line is high.	R
b1	SCLI	SCL Line Monitor	0: SCLn line is low. 1: SCLn line is high.	R
b2	SDAO	SDA Output Control/Monitor	• Read: 0: The RIIC has driven the SDAn pin low. 1: The RIIC has released the SDAn pin. • Write: 0: The RIIC drives the SDAn pin low. 1: The RIIC releases the SDAn pin.	R/W
b3	SCLO	SCL Output Control/Monitor	• Read: 0: The RIIC has driven the SCLn pin low. 1: The RIIC has released the SCLn pin. • Write: 0: The RIIC drives the SCLn pin low. 1: The RIIC releases the SCLn pin. (High level output is achieved through an external pull-up resistor.)	R/W
b4	SOWP	SCLO/SDAO Write Protect	0: SCLO and SDAO bits can be written. 1: SCLO and SDAO bits are protected. (This bit is read as 1.)	R/W
b5	CLO	Extra SCL Clock Cycle Output	0: Does not output an extra SCL clock cycle (default). 1: Outputs an extra SCL clock cycle. (The CLO bit is cleared automatically after one clock cycle is output.)	R/W
b6	IICRST	I2C-bus Interface Internal Reset	0: Releases the RIIC reset or internal reset. 1: Initiates the RIIC reset or internal reset. (Clears the bit counter and the SCLn/SDAn output latch)	R/W
b7	ICE	I2C-bus Interface Enable	0: Disable (SCLn and SDAn pins in inactive state) 1: Enable (SCLn and SDAn pins in active state) (Combined with the IICRST bit to select either RIIC or internal reset.)	R/W

SDAO Bit (SDA Output Control/Monitor) and SCLO Bit (SCL Output Control/Monitor)

These bits are used to directly control the SDAn and SCLn signals output from the RIIC.

When writing to these bits, also write 0 to the SOWP bit.

The result of setting these bits is input to the RIIC via the input buffer. When slave mode is selected, a start condition may be detected and the bus may be released depending on the bit settings.

Do not rewrite these bits during a start condition, stop condition, restart condition, or during transmission or reception. Operation after rewriting under the above conditions is not guaranteed.

When reading these bits, the state of signals output from the RIIC can be read.

CLO Bit (Extra SCL Clock Cycle Output)

This bit is used to output an extra SCL clock cycle for debugging or error processing.
Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.
For details on this function, refer to section 42.11.2, Extra SCL Clock Cycle Output Function.

IICRST Bit (I2C-bus Interface Internal Reset)

This bit is used to reset the internal states of the RIIC.
Setting this bit to 1 initiates an RIIC reset or internal reset.
Whether an RIIC reset or internal reset is initiated is determined according to the combination with the ICE bit. Table 42.3 lists the resets of the RIIC.
The RIIC reset resets all registers and internal states of the RIIC, and the internal reset resets the bit counter (ICMR1.BC[2:0] bits), the I2C-bus shift register (ICDRS), and the I2C-bus status registers (ICSR1 and ICSR2) as well as the internal states of the RIIC. For the reset conditions for each register, refer to section 42.14, Resets and Register and Function States When Issuing Each Condition.
An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the RIIC without initializing the port settings and the control and setting registers of the RIIC when the bus or RIIC hangs up due to a communication error.
If the RIIC hangs up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the SCLn pin and SDAn pin at a high impedance.

Note: If an internal reset is initiated using the IICRST bit for a bus hang-up occurred during communication with the master device in slave mode, the states may become different between the slave device and the master device (due to the difference in the bit counter information). For this reason, do not initiate an internal reset in slave mode, but initiate restoration processing from the master device. If an internal reset is necessary because the RIIC hangs up with the SCLn line in a low level output state in slave mode, initiate an internal reset and then issue a restart condition from the master device or resume communication from the start condition issuance after issuing a stop condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start condition or restart condition from the master device, synchronization will be lost because the master and slave devices operate asynchronously.

Table 42.3 RIIC Resets

IICRST	ICE	State	Specifications
1	0	RIIC reset	Resets all registers and internal states of the RIIC.
	1	Internal reset	Resets the ICMR1.BC[2:0] bits, registers ICSR1, ICSR2, and ICDRS, and the internal states of the RIIC.

ICE Bit (I2C-bus Interface Enable)

This bit selects the active or inactive state of the SCLn and SDAn pins. It can also be combined with the IICRST bit to initiate two types of resets. See Table 42.3, RIIC Resets, for the types of resets.
Set the ICE bit to 1 when using the RIIC. The SCLn and SDAn pins are placed in the active state when the ICE bit is set to 1.
Set the ICE bit to 0 when the RIIC is not to be used. The SCLn and SDAn pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCLn or SDAn pin to the RIIC when setting up the multi-function pin controller (MPC). Note that the slave address comparison operation is carried out if the pins are assigned to the RIIC.

42.2.2 I2C-bus Control Register 2 (ICCR2)

Address(es): RIIC0.ICCR2 0008 8301h, RIIC2.ICCR2 0008 8341h

	b7	b6	b5	b4	b3	b2	b1	b0
	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	ST	Start Condition Issuance Request	0: Does not request to issue a start condition. 1: Requests to issue a start condition.	R/W
b2	RS	Restart Condition Issuance Request	0: Does not request to issue a restart condition. 1: Requests to issue a restart condition.	R/W
b3	SP	Stop Condition Issuance Request	0: Does not request to issue a stop condition. 1: Requests to issue a stop condition.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	TRS	Transmit/Receive Mode	0: Receive mode 1: Transmit mode	R/W*1
b6	MST	Master/Slave Mode	0: Slave mode 1: Master mode	R/W*1
b7	BBSY	Bus Busy Detection Flag	0: The I2C-bus is released (bus free state). 1: The I2C-bus is occupied (bus busy state).	R

Note 1. When the ICMR1.MTWP bit is set to 1, bits MST and TRS can be written to.

ST Bit (Start Condition Issuance Request)

This bit is used to request transition to master mode and issuance of a start condition.

When this bit is set to 1 to request to issue a start condition, a start condition is issued when the BBSY flag is set to 0 (bus free state).

For details on the start condition issuance, refer to section 42.10, Start Condition/Restart Condition/Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the ST bit

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition has been issued (a start condition is detected)
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: Set the ST bit to 1 (start condition issuance request) when the BBSY flag is set to 0 (bus free state). Note that arbitration may be lost due to a start condition issuance error if the ST bit is set to 1 (start condition issuance request) when the BBSY flag is set to 1 (bus busy state).

RS Bit (Restart Condition Issuance Request)

This bit is used to request that a restart condition be issued in master mode.
When this bit is set to 1 to request to issue a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode).
For details on the restart condition issuance, refer to section 42.10, Start Condition/Restart Condition/Stop Condition Issuing Function.
[Setting condition]

- When 1 is written to the RS bit with the ICCR2.BBSY flag set to 1

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition has been issued (a start condition is detected)
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: Do not set the RS bit to 1 while issuing a stop condition.
Note: If 1 (requests to issue a restart condition) is written to the RS bit in slave mode, the restart condition is not issued but the RS bit remains set to 1. If the operating mode changes to master mode with the bit not being cleared, note that the restart condition may be issued.

SP Bit (Stop Condition Issuance Request)

This bit is used to request that a stop condition be issued in master mode.
When this bit is set to 1 to request to issue a stop condition, a stop condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode).
For details on the stop condition issuance, refer to section 42.10, Start Condition/Restart Condition/Stop Condition Issuing Function.
[Setting condition]

- When 1 is written to the SP bit with both the BBSY flag and the ICCR2.MST bit set to 1

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition has been issued (a stop condition is detected)
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: Writing to the SP bit is not possible while the setting of the BBSY flag is 0 (bus free state).
Note: Do not set the SP bit to 1 while a restart condition is being issued.

TRS Bit (Transmit/Receive Mode)

This bit indicates transmit or receive mode.
The RIIC is in receive mode when the TRS bit is set to 0 and is in transmit mode when the bit is set to 1. Combination of this bit and the MST bit indicates the operating mode of the RIIC.
The value of TRS bit is automatically changed to 1 for transmit mode or 0 for receive mode by issuing or detection of a start condition and setting of the R/W# bit. Although writing to the TRS bit is possible when the ICMR1.MTWP bit is set to 1, writing to this bit is not necessary during normal usage.
[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When a restart condition is issued normally according to the restart condition issuance request (when a restart condition is detected with the RS bit set to 1)
- When the R/W# bit added to the slave address is set to 0 in master mode
- When the address received in slave mode matches the address enabled in the ICSER register, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the ICMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- The ICSR2.AL (arbitration-lost) flag being set to 1
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended
- In slave mode, a match between the received address and the address enabled in the ICSER register when the value of the received R/W# bit is 0 (including cases where the received address is the general call address)
- In slave mode, a restart condition is detected (a start condition is detected with ICCR2.BBSY flag is 1 and ICCR2.MST bit is 0)
- When 0 is written to the TRS bit with the ICMR1.MTWP bit set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

MST Bit (Master/Slave Mode)

This bit indicates master or slave mode.
The RIIC is in slave mode when the MST bit is set to 0 and is in master mode when the bit is set to 1. Combination of this bit and the TRS bit indicates the operating mode of the RIIC.
The value of the MST bit is automatically changed to 1 for master mode or 0 for slave mode by issuing of a start condition and issuing or detection of a stop condition, etc. Although writing to the MST bit is possible when the ICMR1.MTWP bit is set to 1, writing to this bit is not necessary during normal usage.
[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the ICMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When 0 is written to the MST bit with the ICMR1.MTWP bit set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

BBSY Flag (Bus Busy Detection Flag)

The BBSY flag indicates whether the I2C-bus is occupied (bus busy state) or released (bus free state).
This bit is set to 1 when the SDAn line changes from high to low under the condition of SCLn line = high, assuming that a start condition has been issued.
When the SDAn line changes from low to high under the condition of SCLn line = high, this bit is set to 0 after the bus free time (specified in the ICBRL register) start condition is not detected, assuming that a stop condition has been issued.
[Setting condition]

- When a start condition is detected

[Clearing conditions]

- When the bus free time (specified in the ICBRL register) start condition is not detected after detecting a stop condition
- When 1 is written to the ICCR1.IICRST bit with the ICCR1.ICE bit set to 0 (RIIC reset)

42.2.3 I2C-bus Mode Register 1 (ICMR1)

Address(es): RIIC0.ICMR1 0008 8302h, RIIC2.ICMR1 0008 8342h

	b7	b6	b5	b4	b3	b2	b1	b0
	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BC[2:0]	Bit Counter	b2 b0 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits	R/W*1
b3	BCWP	BC Write Protect	0: Enables a value to be written in the BC[2:0] bits. (This bit is read as 1.)	R/W*1
b6 to b4	CKS[2:0]	Internal Reference Clock Select	Select the internal reference clock (IICφ) source for the RIIC. b6 b4 0 0 0: PCLK/1 clock 0 0 1: PCLK/2 clock 0 1 0: PCLK/4 clock 0 1 1: PCLK/8 clock 1 0 0: PCLK/16 clock 1 0 1: PCLK/32 clock 1 1 0: PCLK/64 clock 1 1 1: PCLK/128 clock	R/W
b7	MTWP	MST/TRS Write Protect	0: Disables writing to the ICCR2.MST and TRS bits. 1: Enables writing to the ICCR2.MST and TRS bits.	R/W

Note 1. Rewrite the BC[2:0] bits and set the BCWP bit to 0 at the same time.

BC[2:0] Bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCLn line. Although these bits are writable and readable, it is not necessary to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one (data is transferred with an additional acknowledge bit) between transferred bytes when the SCLn line is at a low level.

The values of the BC[2:0] bits return to 000b at the end of a data transfer including the acknowledge bit or when a start condition including a restart condition is detected.

42.2.4 I2C-bus Mode Register 2 (ICMR2)

Address(es): RIIC0.ICMR2 0008 8303h, RIIC2.ICMR2 0008 8343h

	b7	b6	b5	b4	b3	b2	b1	b0
	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOS	Timeout Detection Time Select	0: Long mode is selected. 1: Short mode is selected.	R/W
b1	TMOL	Timeout L Count Control	0: Count-up is disabled while the SCLn line is at a low level. 1: Count-up is enabled while the SCLn line is at a low level.	R/W
b2	TMOH	Timeout H Count Control	0: Count-up is disabled while the SCLn line is at a high level. 1: Count-up is enabled while the SCLn line is at a high level.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	SDDL[2:0]	SDA Output Delay Counter	• When ICMR2.DLCS bit is 0 (IICφ) b6 b4 0 0 0: No output delay 0 0 1: 1 IICφ cycle 0 1 0: 2 IICφ cycles 0 1 1: 3 IICφ cycles 1 0 0: 4 IICφ cycles 1 0 1: 5 IICφ cycles 1 1 0: 6 IICφ cycles 1 1 1: 7 IICφ cycles • When ICMR2.DLCS bit is 1 (IICφ/2) b6 b4 0 0 0: No output delay 0 0 1: 1 or 2 IICφ cycles 0 1 0: 3 or 4 IICφ cycles 0 1 1: 5 or 6 IICφ cycles 1 0 0: 7 or 8 IICφ cycles 1 0 1: 9 or 10 IICφ cycles 1 1 0: 11 or 12 IICφ cycles 1 1 1: 13 or 14 IICφ cycles	R/W
b7	DLCS	SDA Output Delay Clock Source Select	0: The internal reference clock (IICφ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (IICφ/2) is selected as the clock source of the SDA output delay counter.*1	R/W

Note 1. The DLCS bit setting of 1 (IICφ/2) only becomes valid when SCL pin is low. When SCL pin is high, the DLCS bit setting of 1 becomes invalid and the clock source becomes the internal reference clock (IICφ).

TMOS Bit (Timeout Detection Time Select)

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (ICFER.TMOE bit is 1). When this bit is set to 0, long mode is selected. When this bit is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16-bit counter. In short mode, the counter functions as a 14-bit counter. While the SCLn line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IICφ) as a count source.
For details on the timeout function, refer to section 42.11.1, Timeout Function.

TMOL Bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCLn line is held low when the timeout function is enabled (ICFER.TMOE bit is 1).

TMOH Bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCLn line is held high when the timeout function is enabled (ICFER.TMOE bit is 1).

SDDL[2:0] Bits (SDA Output Delay Counter)

The SDA output can be delayed by the SDDL[2:0] setting. This counter works with the clock source selected by the DLCS bit. The setting of this function can be used for all types of SDA output, including the transmission of the acknowledge bit.

Set the SDA output delay time to meet the I2C-bus specification (within the data enable time/acknowledge enable time*1) or the SMBus specification (within the data hold time: 300 ns or more, and SCL-clock low-level period - the data setup time: 250 ns). Note that, if a value outside the specification is set, communication with communication devices may malfunction or it may seemingly become a start condition or stop condition depending on the bus state.

For details on this function, refer to section 42.5, SDA Output Delay Function.

Note 1. Data enable time/acknowledge enable time
3,450 ns (up to 100 kbps: Standard-mode (Sm))
900 ns (up to 400 kbps: Fast-mode (Fm))
450 ns (up to 1 Mbps: Fast-mode Plus (Fm+))

42.2.5 I2C-bus Mode Register 3 (ICMR3)

Address(es): RIIC0.ICMR3 0008 8304h, RIIC2.ICMR3 0008 8344h

	b7	b6	b5	b4	b3	b2	b1	b0
	SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	NF[1:0]	Noise Filter Stage Select	b1 b0 0 0: Noise of up to one IICφ cycle is filtered out (single-stage filter). 0 1: Noise of up to two IICφ cycles is filtered out (2-stage filter). 1 0: Noise of up to three IICφ cycles is filtered out (3-stage filter). 1 1: Noise of up to four IICφ cycles is filtered out (4-stage filter).	R/W
b2	ACKBR	Receive Acknowledge	0: 0 is received as the acknowledge bit (ACK reception). 1: 1 is received as the acknowledge bit (NACK reception).	R
b3	ACKBT	Transmit Acknowledge	0: 0 is sent as the acknowledge bit (ACK transmission). 1: 1 is sent as the acknowledge bit (NACK transmission).	R/W*1
b4	ACKWP	ACKBT Write Protect	0: Modification of the ACKBT bit is disabled. 1: Modification of the ACKBT bit is enabled.	R/W*1
b5	RDRFS	RDRF Flag Set Timing Select	0: The RDRF flag is set at the rising edge of the ninth SCL clock cycle. (The SCLn line is not held low at the falling edge of the eighth clock cycle.) 1: The RDRF flag is set at the rising edge of the eighth SCL clock cycle. (The SCLn line is held low at the falling edge of the eighth clock cycle.) Low-hold is released by writing a value to the ACKBT bit.	R/W*2
b6	WAIT	WAIT	0: No WAIT (The period between ninth clock cycle and first clock cycle is not held low.) 1: WAIT (The period between ninth clock cycle and first clock cycle is held low.) Low-hold is released by reading the ICDRR register.	R/W*2
b7	SMBS	SMBus/I2C-bus Select	0: The I2C-bus is selected. 1: The SMBus is selected.	R/W

Note 1. Write to the ACKBT bit only while the ACKWP bit is already 1. If it is attempted to write 1 to both the ACKWP and ACKBT bits at the same time, the ACKBT bit will not be set to 1.

Note 2. The WAIT and RDRFS bits are valid only in receive mode (invalid in transmit mode).

NF[1:0] Bits (Noise Filter Stage Select)

These bits are used to select the number of stages in the digital noise filter.

For details on the digital noise filter function, refer to section 42.6, Digital Noise Filter Circuit.

Note: Set the noise range to be filtered out by the noise filter within a range less than the SCLn line high-level period or low-level period. If the noise filter width is set to a value of [the shorter one of either SCL high width or SCL low width] – {1.5 × t_{IICcyc} (cycle time of internal reference clock (IICφ)) + 120 ns (pulse width suppressed by the analog noise filter, a reference value)} or a greater value, the SCL clock is regarded as noise by the noise filter function of the RIIC, which may prevent the RIIC from operating normally.

ACKBR Bit (Receive Acknowledge)

This bit is used to store the acknowledge bit information received from the receive device in transmit mode.
[Setting condition]

- When 1 is received as the acknowledge bit with the ICCR2.TRS bit set to 1

[Clearing conditions]

- When 0 is received as the acknowledge bit with the ICCR2.TRS bit set to 1
- When 1 is written to the ICCR1.IICRST bit while the ICCR1.ICE bit is 0 (RIIC reset)

ACKBT Bit (Transmit Acknowledge)

This bit is used to set the bit to be sent at the acknowledge timing in receive mode.
[Setting condition]

- When 1 is written to this bit with the ACKWP bit set to 1

[Clearing conditions]

- When 0 is written to this bit with the ACKWP bit set to 1
- When stop condition issuance is detected (when a stop condition is detected with the ICCR2.SP bit set to 1)
- When 1 is written to the ICCR1.IICRST bit while the ICCR1.ICE bit is 0 (RIIC reset)

ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.

RDRFS Bit (RDRF Flag Set Timing Select)

This bit is used to select the RDRF flag set timing in receive mode and also to select whether to hold the SCLn line low at the falling edge of the eighth SCL clock cycle.
When the RDRFS bit is 0, the SCLn line is not held low at the falling edge of the eighth SCL clock cycle, and the RDRF flag is set to 1 at the rising edge of the ninth SCL clock cycle.
When the RDRFS bit is 1, the RDRF flag is set to 1 at the rising edge of the eighth SCL clock cycle and the SCLn line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the SCLn line is released by writing a value to the ACKBT bit.
After data is received with this setting, the SCLn line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT bit is 0) or NACK (ACKBT bit is 1) according to receive data.

WAIT Bit (WAIT)

This bit is used to control whether to hold the period between the ninth SCL clock cycle and the first SCL clock cycle low until the I2C-bus receive data register (ICDRR) is completely read each time single-byte data is received in receive mode.
When the WAIT bit is 0, the receive operation is continued without holding the period between the ninth and the first SCL clock cycle low. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.
When the WAIT bit is 1, the SCLn line is held low from the falling edge of the ninth clock cycle until the ICDRR register value is read each time single-byte data is received. This enables receive operation in byte units.

Note: When the value of the WAIT bit is to be read, be sure to read the ICDRR register beforehand.

SMBS Bit (SMBus/I2C-bus Select)

Setting this bit to 1 selects the SMBus and enables the ICSER.HOAE bit.

42.2.6 I2C-bus Function Enable Register (ICFER)

Address(es): RIIC0.ICFER 0008 8305h, RIIC2.ICFER 0008 8345h

	b7	b6	b5	b4	b3	b2	b1	b0
	FMPE	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset:	0	1	1	1	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOE	Timeout Function Enable	0: The timeout function is disabled. 1: The timeout function is enabled.	R/W
b1	MALE	Master Arbitration-Lost Detection Enable	0: Master arbitration-lost detection is disabled. (Disables the arbitration-lost detection function and does not clear the ICCR2.MST and TRS bits automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the ICCR2.MST and TRS bits automatically when arbitration is lost.)	R/W
b2	NALE	NACK Transmission Arbitration-Lost Detection Enable	0: NACK transmission arbitration-lost detection is disabled. 1: NACK transmission arbitration-lost detection is enabled.	R/W
b3	SALE	Slave Arbitration-Lost Detection Enable	0: Slave arbitration-lost detection is disabled. 1: Slave arbitration-lost detection is enabled.	R/W
b4	NACKE	NACK Reception Transfer Suspension Enable	0: Transfer operation is not suspended during NACK reception (transfer suspension disabled). 1: Transfer operation is suspended during NACK reception (transfer suspension enabled).	R/W
b5	NFE	Digital Noise Filter Circuit Enable	0: No digital noise filter circuit is used. 1: A digital noise filter circuit is used.	R/W
b6	SCLE	SCL Synchronous Circuit Enable	0: No SCL synchronous circuit is used. 1: An SCL synchronous circuit is used.	R/W
b7	FMPE*1	Fast-Mode Plus Enable	0: No Fm+ slope control circuit is used for the SCLn pin and SDAn pin. 1: An Fm+ slope control circuit is used for the SCLn pin and SDAn pin.	R/W

Note 1. The Fast-mode Plus enable bit (FMPE) is only supported by RIIC0. In RIIC2, bit 7 is reserved.

TMOE Bit (Timeout Function Enable)

This bit is used to enable or disable the timeout function.

For details on the timeout function, refer to section 42.11.1, Timeout Function.

MALE Bit (Master Arbitration-Lost Detection Enable)

This bit is used to specify whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

NALE Bit (NACK Transmission Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode (such as when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with different number of receive bytes).

SALE Bit (Slave Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

NACKE Bit (NACK Reception Transfer Suspension Enable)

This bit is used to specify whether to continue or discontinue the transfer operation when NACK is received from the slave device in transmit mode. Normally, set this bit to 1.
When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended.
When the NACKE bit is 0, the next transfer operation is continued regardless of the received acknowledge content.
For details on the NACK reception transfer suspension function, refer to section 42.8.2, NACK Reception Transfer Suspension Function.

SCLE Bit (SCL Synchronous Circuit Enable)

This bit is used to specify whether to synchronize the SCL clock with the SCL input clock. Normally, set this bit to 1.
When the SCLE bit is set to 0 (no SCL synchronous circuit used), the RIIC does not synchronize the SCL clock with the SCL input clock. In this setting, the RIIC outputs the SCL clock with the transfer rate set in registers ICBRH and ICBRL regardless of the SCLn line state. For this reason, if the bus load of the I2C-bus line is much larger than the specification value or if the SCL clock output overlaps in multiple masters, the short-cycle SCL clock that does not meet the specification may be output. When no SCL synchronous circuit is used, it also affects the issuance of a start condition, restart condition, and stop condition, and the continuous output of extra SCL clock cycles.
This bit must not be set to 0 except for checking the output of the set transfer rate.

FMPE Bit (Fast-Mode Plus Enable)

This bit is used to specify whether to use a slope control circuit for Fast-mode Plus (Fm+).
When this bit is set to 1, a slope control circuit conforming to the Fast-mode Plus (Fm+) slope control specification (tof) of the I2C-bus is selected. When this bit is set to 0, a slope control circuit conforming to the Standard-mode (Sm) and Fast-mode (Fm) slope control specification (tof) of the I2C-bus is selected.
Set this bit to 1 when using the transmission rate within a range up to 1 Mbps (Fast-mode Plus (Fm+)) of the I2C-bus specification. Set this bit to 0 when using the transmission rate at other rates (up to 100 kbps (Sm), up to 400 kbps (Fm)) or for SMBus (10 to 100 kbps).

42.2.7 I2C-bus Status Enable Register (ICSER)

Address(es): RIIC0.ICSER 0008 8306h, RIIC2.ICSER 0008 8346h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
Value after reset:	0	0	0	0	1	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	SAR0E	Slave Address Register 0 Enable	0: Slave address in registers SARL0 and SARU0 is disabled. 1: Slave address in registers SARL0 and SARU0 is enabled.	R/W
b1	SAR1E	Slave Address Register 1 Enable	0: Slave address in registers SARL1 and SARU1 is disabled. 1: Slave address in registers SARL1 and SARU1 is enabled.	R/W
b2	SAR2E	Slave Address Register 2 Enable	0: Slave address in registers SARL2 and SARU2 is disabled. 1: Slave address in registers SARL2 and SARU2 is enabled.	R/W
b3	GCAE	General Call Address Enable	0: General call address detection is disabled. 1: General call address detection is enabled.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DIDE	Device-ID Address Detection Enable	0: Device-ID address detection is disabled. 1: Device-ID address detection is enabled.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOAE	Host Address Enable	0: Host address detection is disabled. 1: Host address detection is enabled.	R/W

SARyE Bit (Slave Address Register y Enable) (y = 0 to 2)

This bit is used to enable or disable the slave address set in registers SARLy and SARUy.

When this bit is set to 1, the slave address set in registers SARLy and SARUy is enabled and is compared with the received slave address.

When this bit is set to 0, the slave address set in registers SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

GCAE Bit (General Call Address Enable)

This bit is used to specify whether to ignore the general call address (0000 000b + 0 (write): All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the RIIC recognizes the received slave address as the general call address independently of the slave addresses set in registers SARLy and SARUy (y = 0 to 2) and performs data receive operation.

When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

DIDE Bit (Device-ID Address Detection Enable)

This bit is used to specify whether to recognize and execute the device-ID address when a device ID (1111 100b) is received in the first byte after a start condition or restart condition is detected.

When this bit is set to 1, if the received first byte matches the device ID, the RIIC recognizes that the device-ID address has been received. When the following R/W# bit is 0 (write), the RIIC recognizes the second and the following bytes as slave addresses and continues the receive operation.

When this bit is set to 0, the RIIC ignores the received first byte even if it matches the device ID address and recognizes the first byte as a normal slave address.

For details on the device-ID address detection, refer to section 42.7.3, Device-ID Address Detection.

HOAE Bit (Host Address Enable)

This bit is used to specify whether to ignore received host address (0001 000b) when the ICMR3.SMBS bit is 1.

When this bit is set to 1 while the ICMR3.SMBS bit is 1, if the received slave address matches the host address, the RIIC recognizes the received slave address as the host address independently of the slave addresses set in registers SARLy and SARUy (y = 0 to 2) and performs the receive operation.

When the ICMR3.SMBS bit or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

42.2.8 I2C-bus Interrupt Enable Register (ICIER)

Address(es): RIIC0.ICIER 0008 8307h, RIIC2.ICIER 0008 8347h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOIE	Timeout Interrupt Request Enable	0: Timeout interrupt (TMOI) request is disabled. 1: Timeout interrupt (TMOI) request is enabled.	R/W
b1	ALIE	Arbitration-Lost Interrupt Request Enable	0: Arbitration-lost interrupt (ALI) request is disabled. 1: Arbitration-lost interrupt (ALI) request is enabled.	R/W
b2	STIE	Start Condition Detection Interrupt Request Enable	0: Start condition detection interrupt (STI) request is disabled. 1: Start condition detection interrupt (STI) request is enabled.	R/W
b3	SPIE	Stop Condition Detection Interrupt Request Enable	0: Stop condition detection interrupt (SPI) request is disabled. 1: Stop condition detection interrupt (SPI) request is enabled.	R/W
b4	NAKIE	NACK Reception Interrupt Request Enable	0: NACK reception interrupt (NAKI) request is disabled. 1: NACK reception interrupt (NAKI) request is enabled.	R/W
b5	RIE	Receive Data Full Interrupt Request Enable	0: Receive data full interrupt (RXI) request is disabled. 1: Receive data full interrupt (RXI) request is enabled.	R/W
b6	TEIE	Transmit End Interrupt Request Enable	0: Transmit end interrupt (TEI) request is disabled. 1: Transmit end interrupt (TEI) request is enabled.	R/W
b7	TIE	Transmit Data Empty Interrupt Request Enable	0: Transmit data empty interrupt (TXI) request is disabled. 1: Transmit data empty interrupt (TXI) request is enabled.	R/W

TMOIE Bit (Timeout Interrupt Request Enable)

This bit is used to enable or disable timeout interrupt (TMOI) requests when the ICSR2.TMOF flag is set to 1. A TMOI interrupt request is canceled by setting the TMOF flag or the TMOIE bit to 0.

ALIE Bit (Arbitration-Lost Interrupt Request Enable)

This bit is used to enable or disable arbitration-lost interrupt (ALI) requests when the ICSR2.AL flag is set to 1. An ALI interrupt request is canceled by setting the AL flag or the ALIE bit to 0.

STIE Bit (Start Condition Detection Interrupt Request Enable)

This bit is used to enable or disable start condition detection interrupt (STI) requests when the ICSR2.START flag is set to 1. An STI interrupt request is canceled by setting the START flag or the STIE bit to 0.

SPIE Bit (Stop Condition Detection Interrupt Request Enable)

This bit is used to enable or disable stop condition detection interrupt (SPI) requests when the ICSR2.STOP flag is set to 1. An SPI interrupt request is canceled by setting the STOP flag or the SPIE bit to 0.

NAKIE Bit (NACK Reception Interrupt Request Enable)

This bit is used to enable or disable NACK reception interrupt (NAKI) requests when the ICSR2.NACKF flag is set to 1. An NAKI interrupt request is canceled by setting the NACKF flag or the NAKIE bit to 0.

RIE Bit (Receive Data Full Interrupt Request Enable)

This bit is used to enable or disable receive data full interrupt (RXI) requests when the ICSR2.RDRF flag is set to 1.

TEIE Bit (Transmit End Interrupt Request Enable)

This bit is used to enable or disable transmit end interrupt (TEI) requests when the ICSR2.TEND flag is set to 1. An TEI interrupt request is canceled by setting the TEND flag or the TEIE bit to 0.

TIE Bit (Transmit Data Empty Interrupt Request Enable)

This bit is used to enable or disable transmit data empty interrupt (TXI) requests when the ICSR2.TDRE flag is set to 1.

42.2.9 I2C-bus Status Register 1 (ICSR1)

Address(es): RIIC0.ICSR1 0008 8308h, RIIC2.ICSR1 0008 8348h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	AAS0	Slave Address 0 Detection Flag	0: Slave address 0 is not detected. 1: Slave address 0 is detected.	R/(W) *1
b1	AAS1	Slave Address 1 Detection Flag	0: Slave address 1 is not detected. 1: Slave address 1 is detected.	R/(W) *1
b2	AAS2	Slave Address 2 Detection Flag	0: Slave address 2 is not detected. 1: Slave address 2 is detected.	R/(W) *1
b3	GCA	General Call Address Detection Flag	0: General call address is not detected. 1: General call address is detected.	R/(W) *1
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DID	Device-ID Address Detection Flag	0: Device-ID command is not detected. 1: Device-ID command is detected. • This bit is set to 1 when the first byte received immediately after a start condition is detected matches a value of (device ID (1111 100b) + 0 (write)).	R/(W) *1
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOA	Host Address Detection Flag	0: Host address is not detected. 1: Host address is detected. • This bit is set to 1 when the received slave address matches the host address (0001 000b).	R/(W) *1

Note 1. Only 0 can be written to clear the flag.

AASy Flag (Slave Address y Detection Flag) (y = 0 to 2)

[Setting conditions]

For 7-bit address format: SARUy.FS bit = 0

- When the received slave address matches the SARLy.SVA[6:0] bits value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte.

For 10-bit address format: SARUy.FS bit = 1

- When the received slave address matches a value of (11110b + SARUy.SVA[1:0] bits) and the following address matches the SARLy value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the second byte.

[Clearing conditions]

- When 0 is written to the AASy flag after reading the AASy flag to be 1
- When a stop condition is detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

For 7-bit address format: SARUy.FS bit = 0

- When the received slave address does not match the SARLy.SVA[6:0] bits value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte.

For 10-bit address format: SARUy.FS bit = 1

- When the received slave address does not match a value of (11110b + SARUy.SVA[1:0] bits) with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte.
- When the received slave address matches a value of (11110b + SARUy.SVA[1:0] bits) and the following address does not match the SARLy value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the second byte.

GCA Flag (General Call Address Detection Flag)

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 (write)) with the ICSER.GCAE bit set to 1 (general call address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte.

[Clearing conditions]

- When 0 is written to the GCA flag after reading GCA flag to be 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000b + 0 (write)) with the ICSER.GCAE bit set to 1 (general call address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte.
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

DID Flag (Device-ID Address Detection Flag)

[Setting condition]

- When the first byte received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 (write)) with the ICSER.DIDE bit set to 1 (device-ID address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte.

[Clearing conditions]

- When 0 is written to the DID flag after reading DID flag to be 1
- When a stop condition is detected
- When the first byte received immediately after a start condition or restart condition is detected does not match a value of (device ID (1111 100b)) with the ICSER.DIDE bit set to 1 (device-ID address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte.
- When the first byte received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 (write)) and the second byte does not match any of slave addresses 0 to 2 with the ICSER.DIDE bit set to 1 (device-ID address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the second byte.
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

HOA Flag (Host Address Detection Flag)

[Setting condition]

- When the received slave address matches the host address (0001 000b) with the ICSER.HOAE bit set to 1 (host address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte.

[Clearing conditions]

- When 0 is written to the HOA flag after reading HOA flag to be 1
- When a stop condition is detected
- When the received slave address does not match the host address (0001 000b) with the ICSER.HOAE bit set to 1

(host address detection is enabled)
This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte.

- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

42.2.10 I2C-bus Status Register 2 (ICSR2)

Address(es): RIIC0.ICSR2 0008 8309h, RIIC2.ICSR2 0008 8349h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOF	Timeout Detection Flag	0: Timeout is not detected. 1: Timeout is detected.	R/(W) *1
b1	AL	Arbitration-Lost Flag	0: Arbitration is not lost. 1: Arbitration is lost.	R/(W) *1
b2	START	Start Condition Detection Flag	0: Start condition is not detected. 1: Start condition is detected.	R/(W) *1
b3	STOP	Stop Condition Detection Flag	0: Stop condition is not detected. 1: Stop condition is detected.	R/(W) *1
b4	NACKF	NACK Detection Flag	0: NACK is not detected. 1: NACK is detected.	R/(W) *1
b5	RDRF	Receive Data Full Flag	0: The ICDRR register contains no receive data. 1: The ICDRR register contains receive data.	R/(W) *1
b6	TEND	Transmit End Flag	0: Data is being transmitted. 1: Data has been transmitted.	R/(W) *1
b7	TDRE	Transmit Data Empty Flag	0: The ICDRT register contains transmit data. 1: The ICDRT register contains no transmit data.	R

Note 1. Only 0 can be written to clear the flag.

TMOF Flag (Timeout Detection Flag)

This flag is set to 1 when the RIIC recognizes timeout after the SCLn line state remains unchanged for a certain period.
[Setting condition]

- When the SCLn line state remains unchanged for the period specified by bits ICMR2.TMOH, TMOL, and TMOS while the ICFER.TMOE bit is 1 (the timeout function is enabled) in master mode or in slave mode and the received slave address matches.

[Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

AL Flag (Arbitration-Lost Flag)

This flag shows that bus mastership has been lost (loss in arbitration) due to a bus conflict or some other reason when a start condition is issued or an address and data are transmitted. The RIIC monitors the level on the SDAn line during transmission and, if the level on the line does not match the value of the bit being output, sets the value of the AL flag to 1 to indicate that the bus is occupied by another device.
The RIIC can also set the flag to indicate the detection of loss of arbitration during NACK transmission in master mode or during data transmission in slave mode.

[Setting conditions]

When master arbitration-lost detection is enabled: ICFER.MALE = 1

- When the internal SDA output state does not match the SDAn line level at the rising edge of SCL clock except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDAn line is driven low while the internal SDA output is at a high level (the SDAn pin is in the high-impedance state))
- When a start condition is detected while the ICCR2.ST bit is 1 (start condition issuance request) or the internal SDA output state does not match the SDAn line level
- When the ICCR2.ST bit is set to 1 (start condition issuance request) with the ICCR2.BBSY flag set to 1.

When NACK arbitration-lost detection is enabled: ICFER.NALE = 1

- When the internal SDA output state does not match the SDAn line level at the rising edge of SCL clock in the ACK period during NACK transmission in receive mode

When slave arbitration-lost detection is enabled: ICFER.SALE = 1

- When the internal SDA output state does not match the SDAn line level at the rising edge of SCL clock except for the ACK period during data transmission in slave transmit mode

[Clearing conditions]

- When 0 is written to the AL flag after reading AL = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Table 42.4 Relationship between Arbitration-Lost Generation Sources and Arbitration-Lost Enable Functions

ICFER			ICSR2		
MALE	NALE	SALE	AL	Error	Arbitration-Lost Generation Source
1	×	×	1	Start condition issuance error	When internal SDA output state does not match SDAn line level when a start condition is detected while the ICCR2.ST bit is 1
					When ICCR2.ST bit is set to 1 with ICCR2.BBSY flag set to 1
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode
×	1	×	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master receive mode or slave receive mode
×	×	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

×: Don't care

START Flag (Start Condition Detection Flag)

[Setting condition]

- When a start condition (or a restart condition) is detected

[Clearing conditions]

- When 0 is written to the START bit after reading START = 1
- When a stop condition is detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

STOP Flag (Stop Condition Detection Flag)

[Setting condition]

- When a stop condition is detected

[Clearing conditions]

- When 0 is written to the STOP bit after reading STOP = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

NACKF Flag (NACK Detection Flag)

[Setting condition]

- When acknowledge is not received (NACK is received) from the receive device in transmit mode with the ICFER.NACKE bit set to 1 (transfer suspension enabled)

[Clearing conditions]

- When 0 is written to the NACKF bit after reading NACKF = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: When the NACKF flag is set to 1, the RIIC suspends data transmission/reception. Writing to the ICDRT register in transmit mode or reading from the ICDRR register in receive mode with the NACKF flag set to 1 does not enable data transmit/receive operation. To restart data transmission/reception, set the NACKF flag to 0.

RDRF Flag (Receive Data Full Flag)

[Setting conditions]

- When receive data has been transferred from the ICDRS register to the ICDRR register
  This flag is set to 1 at the rising edge of the eighth or ninth SCL clock cycle (selected by the ICMR3.RDRFS bit)
- When the received slave address matches after a start condition (or a restart condition) is detected with the ICCR2.TRS bit set to 0

[Clearing conditions]

- When 0 is written to the RDRF bit after reading RDRF = 1
- When data is read from the ICDRR register
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

TEND Flag (Transmit End Flag)

[Setting condition]

- At the rising edge of the ninth SCL clock cycle while the TDRE flag is 1

[Clearing conditions]

- When 0 is written to the TEND bit after reading TEND = 1
- When data is written to the ICDRT register
- When a stop condition is detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

TDRE Flag (Transmit Data Empty Flag)

[Setting conditions]

- When data has been transferred from the ICDRT register to the ICDRS register and the ICDRT register becomes empty
- When the ICCR2.TRS bit is set to 1
- When the received slave address matches while the TRS bit is 1

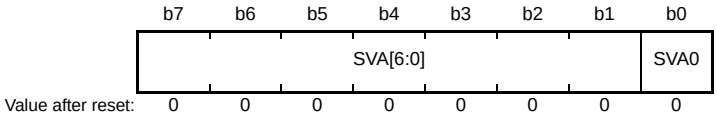
[Clearing conditions]

- When data is written to the ICDRT register
- When the ICCR2.TRS bit is set to 0
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: When the NACKF flag is set to 1 while the ICFER.NACKE bit is 1, the RIIC suspends data transmission/reception. Here, if the TDRE flag is 0 (next transmit data has been written), data is transferred to the ICDRS register and the ICDRT register becomes empty at the rising edge of the ninth clock cycle, but the TDRE flag is not set to 1.

42.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2)

Address(es): RIIC0.SARL0 0008 830Ah, RIIC2.SARL0 0008 834Ah, RIIC0.SARL1 0008 830Ch, RIIC2.SARL1 0008 834Ch, RIIC0.SARL2 0008 830Eh, RIIC2.SARL2 0008 834Eh



Bit	Symbol	Bit Name	Description	R/W
b0	SVA0	10-Bit Address LSB	A slave address is set.	R/W
b7 to b1	SVA[6:0]	7-Bit Address/10-Bit Address Lower Bits	A slave address is set.	R/W

SVA0 Bit (10-Bit Address LSB)

When the 10-bit address format is selected (SARUy.FS bit is 1), this bit functions as the LSB of a 10-bit address and forms the lower 8 bits of a 10-bit address in combination with the SVA[6:0] bits.

When the ICSER.SARyE bit is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, this bit is valid. While the SARUy.FS bit or SARyE bit is 0, the setting of this bit is ignored.

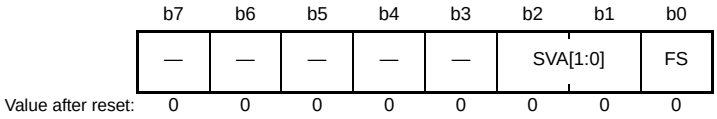
SVA[6:0] Bits (7-Bit Address/10-Bit Address Lower Bits)

When the 7-bit address format is selected (SARUy.FS bit is 0), these bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS bit is 1), these bits function as the lower 8 bits of a 10-bit address in combination with the SVA0 bit.

While the ICSER.SARyE bit is 0, the setting of these bits is ignored.

42.2.12 Slave Address Register Uy (SARUy) (y = 0 to 2)

Address(es): RIIC0.SARU0 0008 830Bh, RIIC2.SARU0 0008 834Bh, RIIC0.SARU1 0008 830Dh, RIIC2.SARU1 0008 834Dh, RIIC0.SARU2 0008 830Fh, RIIC2.SARU2 0008 834Fh



Bit	Symbol	Bit Name	Description	R/W
b0	FS	7-Bit/10-Bit Address Format Select	0: The 7-bit address format is selected. 1: The 10-bit address format is selected.	R/W
b2, b1	SVA[1:0]	10-Bit Address Upper Bits	A slave address is set.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FS Bit (7-Bit/10-Bit Address Format Select)

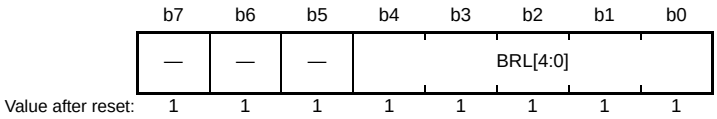
This bit is used to select 7-bit address or 10-bit address for slave address y (in registers SARLy and SARUy).
When the ICSER.SARyE bit is set to 1 (registers SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SARLy.SVA[6:0] bits setting is valid, and the settings of the SVA[1:0] bits and the SARLy.SVA0 bit are ignored.
When the ICSER.SARyE bit is set to 1 (registers SARLy and SARUy enabled) and the SARUy.FS bit is 1, the 10-bit address format is selected for slave address y and the settings of the SVA[1:0] bits and SARLy are valid.
While the ICSER.SARyE bit is 0 (registers SARLy and SARUy disabled), the setting of the SARUy.FS bit is invalid.

SVA[1:0] Bits (10-Bit Address Upper Bits)

When the 10-bit address format is selected (FS = 1), these bits function as the upper 2 bits of a 10-bit address.
When the ICSER.SARyE bit is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, these bits are valid.
While the SARUy.FS bit or SARyE bit is 0, the setting of these bits is ignored.

42.2.13 I2C-bus Bit Rate Low-Level Register (ICBRL)

Address(es): RIIC0.ICBRL 0008 8310h, RIIC2.ICBRL 0008 8350h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRL[4:0]	Bit Rate Low-Level Period	Low-level period of SCL clock	R/W
b7 to b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

ICBRL is a 5-bit register to set the low-level period of SCL clock.
It also works to generate the data setup time for automatic SCL low-hold operation (refer to section 42.8, Automatic Low-Hold Function for SCL); when the RIIC is used only in slave mode, this register needs to be set to a value longer than the data setup time*1.
ICBRL counts the low-level period with the internal reference clock (IICφ) specified by the ICMR1.CKS[2:0] bits.
If the digital noise filter is enabled (the ICFER.NFE bit is 1), set the ICBRL register to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICMR3.NF[1:0] bits.

Note 1. Data setup time (tSU: DAT)
250 ns (up to 100 kbps: Standard-mode (Sm))
100 ns (up to 400 kbps: Fast-mode (Fm))
50 ns (up to 1 Mbps: Fast-mode plus (Fm+))

42.2.14 I2C-bus Bit Rate High-Level Register (ICBRH)

Address(es): RIIC0.ICBRH 0008 8311h, RIIC2.ICBRH 0008 8351h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	BRH[4:0]				
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRH[4:0]	Bit Rate High-Level Period	High-level period of SCL clock	R/W
b7 to b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

ICBRH is a 5-bit register to set the high-level period of SCL clock. ICBRH is valid in master mode. If the RIIC is used only in slave mode, this register need not to set the high-level period.
ICBRH counts the high-level period with the internal reference clock (IICφ) specified by the ICMR1.CKS[2:0] bits.
If the digital noise filter is enabled (the ICFER.NFE bit is 1), set the ICBRH register to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICMR3.NF[1:0] bits.

The I2C transfer rate and the SCL clock duty are calculated using the following expression.
Transfer rate = 1 / {[(ICBRH + 1) + (ICBRL + 1)] / IICφ*1 + SCLn line rising time [tr] + SCLn line falling time [tf]}
Duty cycle = {SCLn line rising time [tr]*2 + (ICBRH + 1) / IICφ} / {SCLn line falling time [tf]*2 + (ICBRL + 1) / IICφ}

Note 1. IICφ = PCLK × Division ratio
Note 2. The SCLn line rising time [tr] and SCLn line falling time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, see the I2C-bus specification from NXP Semiconductors.

Table 42.5 lists examples of ICBRH/ICBRL settings.

Table 42.5 Examples of ICBRH/ICBRL Settings for Transfer Rate

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	8			10			12.5		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	100b	22 (F6h)	25 (F9h)	101b	13 (EDh)	15 (EFh)	101b	16 (F0h)	20 (F4h)
50	010b	16 (F0h)	19 (F3h)	010b	21 (F5h)	24 (F8h)	011b	12 (ECh)	15 (EFh)
100	001b	15 (EFh)	18 (F2h)	001b	19 (F3h)	23 (F7h)	001b	24 (F8h)	29 (FDh)
400	000b	4 (E4h)	10 (EAh)	000b	5 (E5h)	12 (ECh)	000b	7 (E7h)	16 (F0h)
1000	000b	2 (E2h)	3 (E3h)	000b	2 (E2h)	4 (E4h)	000b	3 (E3h)	6 (E6h)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	16			20			25		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	101b	22 (F6h)	25 (F9h)	110b	13 (EDh)	15 (EFh)	110b	16 (F0h)	20 (F4h)
50	011b	16 (F0h)	19 (F3h)	011b	21 (F5h)	24 (F8h)	100b	12 (ECh)	15 (EFh)
100	010b	15 (EFh)	18 (F2h)	010b	19 (F3h)	23 (F7h)	010b	24 (F8h)	29 (FDh)
400	000b	9 (E9h)	20 (F4h)	000b	11 (EBh)	25 (F9h)	001b	7 (E7h)	16 (F0h)
1000	000b	4 (E4h)	7 (E7h)	000b	5 (E5h)	9 (E9h)	000b	6 (E6h)	12 (ECh)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	30			32			33		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	110b	20 (F4h)	24 (F8h)	110b	22 (F6h)	25 (F9h)	110b	22 (F6h)	26 (FAh)
50	100b	15 (EFh)	18 (F2h)	100b	16 (F0h)	19 (F3h)	100b	17 (F1h)	20 (F4h)
100	010b	2 (E2h)	3 (E3h)	011b	15 (EFh)	18 (F2h)	011b	16 (F0h)	19 (F3h)
400	001b	8 (E8h)	19 (F3h)	001b	9 (E9h)	20 (F4h)	001b	9 (E9h)	21 (F5h)
1000	000b	7 (E7h)	14 (EEh)	000b	7 (E7h)	16 (F0h)	000b	8 (E8h)	16 (F0h)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	40			50			60		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	111b	13 (7Dh)	15 (7Fh)	111b	16 (F0h)	20 (F4h)	111b	20 (F4h)	24 (F8h)
50	100b	21 (F5h)	24 (F8h)	100b	26 (FAh)	31 (FFh)	101b	15 (EFh)	18 (F2h)
100	011b	19 (F3h)	23 (F7h)	011b	24 (F8h)	29 (FDh)	011b	2 (E2h)	3 (E3h)
400	001b	11 (7Bh)	25 (F9h)	010b	7 (E7h)	16 (F0h)	010b	8 (E8h)	19 (F3h)
1000	000b	9 (E9h)	20 (F4h)	000b	12 (ECh)	24 (F8h)	000b	15 (EFh)	29 (FDh)

Note: ICBRH/ICBRL settings in these tables are calculated using the following values:
SCLn line rising time (tr): 100 kbps or less (Sm): 1000 ns, 400 kbps or less (Fm): 300 ns, 1 Mbps or less (Fm+): 120 ns
SCLn line falling time (tf): 400 kbps or less (Sm/Fm): 300 ns, 1 Mbps or less (Fm+): 120 ns
For the specified values of SCLn line rising time (tr) and SCLn line falling time (tf), see the I2C-bus specification from NXP Semiconductors.

42.2.15 I2C-bus Transmit Data Register (ICDRT)

Address(es): RIIC0.ICDRT 0008 8312h, RIIC2.ICDRT 0008 8352h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

When the ICDRT register detects a space in the I2C-bus shift register (ICDRS), it transfers the transmit data that has been written to the ICDRT register to the ICDRS register and starts transmitting data in transmit mode.
The double-buffer structure of the ICDRT register and the ICDRS register allows continuous transmit operation if the next transmit data has been written to the ICDRT register while the ICDRS register data is being transmitted.
The ICDRT register can always be read and written. Write transmit data to the ICDRT register once when a transmit data empty interrupt (TXI) request is generated.

42.2.16 I2C-bus Receive Data Register (ICDRR)

Address(es): RIIC0.ICDRR 0008 8313h, RIIC2.ICDRR 0008 8353h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

When 1 byte of data has been received, the received data is transferred from the I2C-bus shift register (ICDRS) to the ICDRR register to enable the next data to be received.
The double-buffer structure of the ICDRS register and the ICDRR register allows continuous receive operation if the received data has been read from the ICDRR register while the ICDRS register is receiving data.
The ICDRR register cannot be written. Read data from the ICDRR register once when a receive data full interrupt (RXI) request is generated.
If the ICDRR register receives the next receive data before the current data is read from the ICDRR register (while the ICSR2.RDRF flag is 1), the RIIC automatically holds the SCL clock low one cycle before the RDRF flag is set to 1 next.

42.2.17 I2C-bus Shift Register (ICDRS)

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	—	—	—	—	—	—	—	—

The ICDRS register is an 8-bit shift register to transmit and receive data.
During transmission, transmit data is transferred from the ICDRT register to the ICDRS register and is sent from the SDAn pin. During reception, data is transferred from the ICDRS register to the ICDRR register after 1 byte of data has been received.
The ICDRS register cannot be accessed directly.

42.3 Operation

42.3.1 Communication Data Format

The I2C-bus format consists of 8-bit data and 1-bit acknowledge. The first byte following a start condition or restart condition is an address byte used to specify a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a stop condition is issued.
Figure 42.3 shows the I2C-bus format, and Figure 42.4 shows the I2C-bus timing.

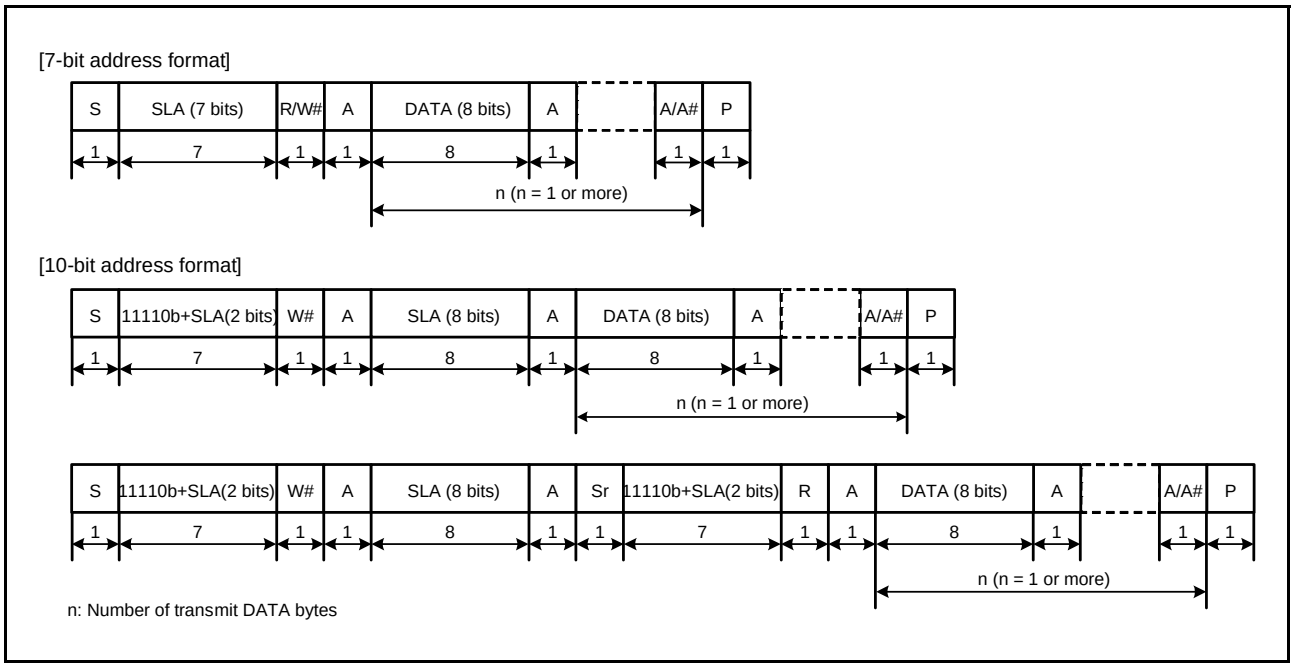


Figure 42.3 I2C-bus Format

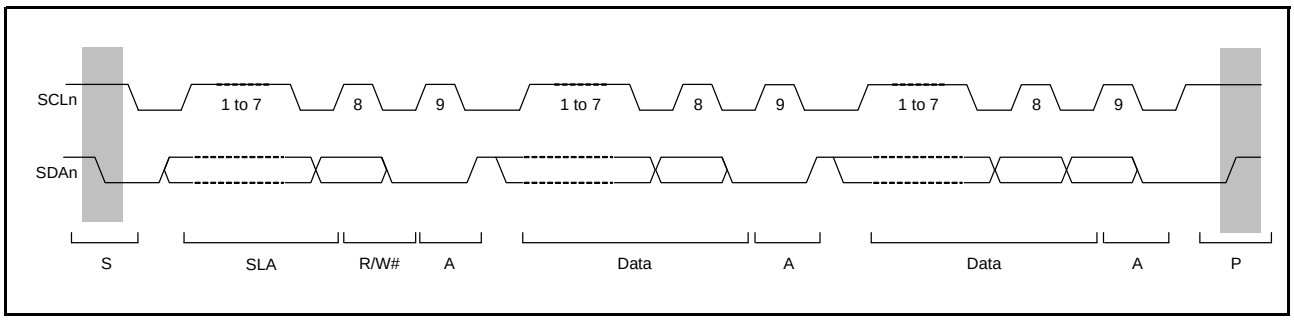


Figure 42.4 I2C-bus Timing (SLA = 7 Bits)

S: Start condition. The master device drives the SDAn line low from high level while the SCLn line is at a high level.

SLA: Slave address, by which the master device selects a slave device.

R/W#: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.

A: Acknowledge. The receive device drives the SDAn line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)

A#: Not Acknowledge. The receive device drives the SDAn line high.

Sr: Restart condition. The master device drives the SDAn line low from the high level after the setup time has elapsed with the SCLn line at the high level.

DATA: Transmitted or received data

P: Stop condition. The master device drives the SDAn line high from low level while the SCLn line is at a high level.

42.3.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in Figure 42.5.
Set the ICCR1.ICE bit to 1 (internal reset) after setting the ICCR1.IICRST bit to 1 (RIIC reset) with the ICCR1.ICE bit set to 0 (SCLn and SDAn pins in inactive state). This initializes the various flags and internal state of the ICSR1 register. After that, set registers SARLy, SARUy, ICSER, ICMR1, ICBRH, and ICBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see Figure 42.5). When the necessary register settings have been completed, set the ICCR1.IICRST bit to 0 (releases the RIIC reset). This step is not necessary if initialization of the RIIC has already been completed.

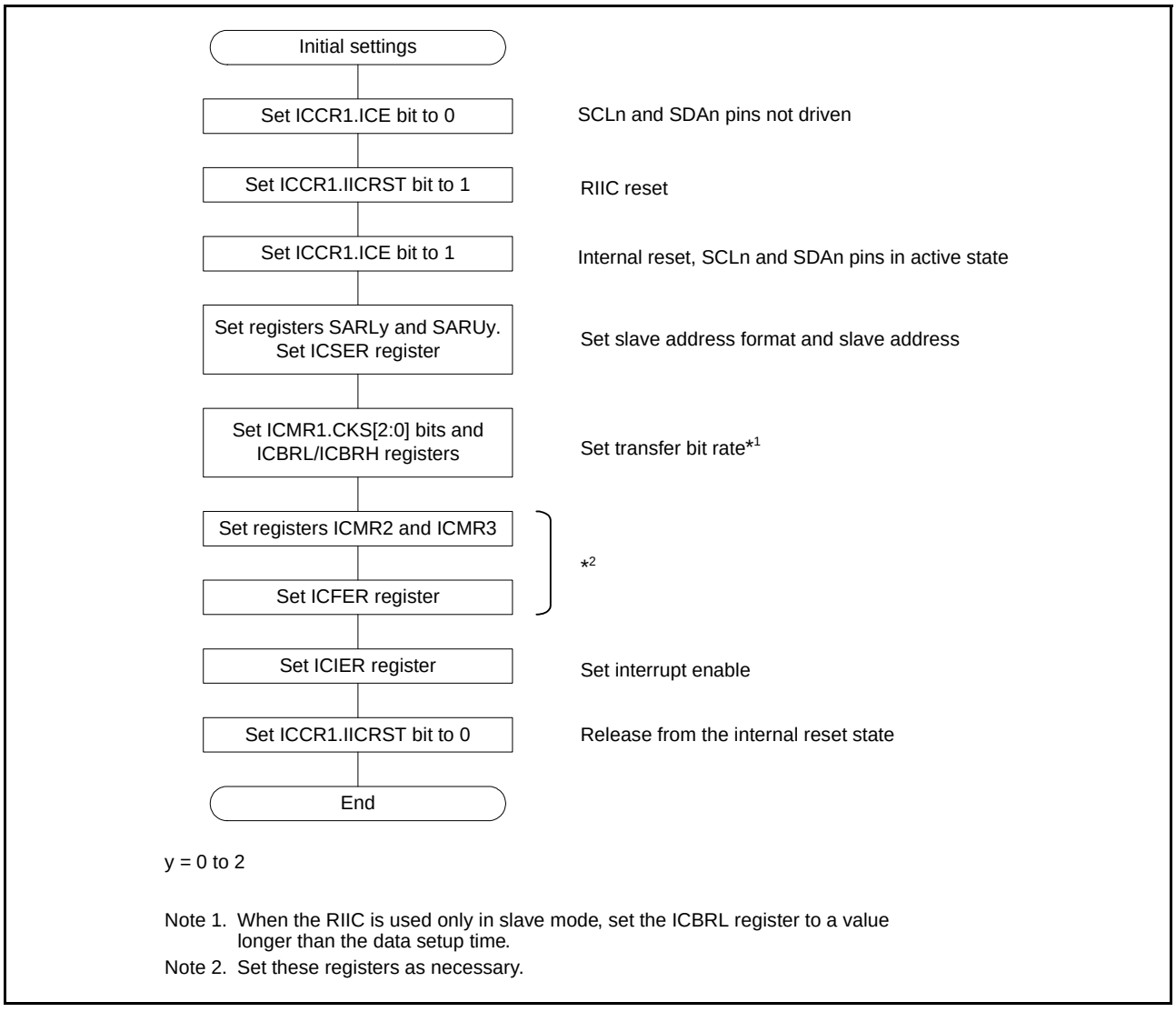


Figure 42.5 Example of RIIC Initialization Flowchart

42.3.3 Master Transmit Operation

In master transmit operation, the RIIC outputs the SCL clock and transmitted data signals as the master device, and the slave device returns acknowledgments. Figure 42.6 shows an example of usage of master transmission and Figure 42.7 to Figure 42.9 show the timing of operations in master transmission.
The following describes the procedure and operations for master transmission.

(1) Initial settings. For details, refer to section 42.3.2, Initial Settings.
(2) Read the ICCR2.BBSY flag to check that the bus is open, and then set the ICCR2.ST bit to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. At the same time, the BBSY flag and the ICSR2.START flag are automatically set to 1 and the ST bit is automatically set to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDAn line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and bits MST and TRS in the ICCR2 register are automatically set to 1, placing the RIIC in master transmit mode. The ICSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.
(3) Check that the ICSR2.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to the ICDRT register. Once the data for transmission are written to the ICDRT register, the TDRE flag is automatically set to 0, the data are transferred from the ICDRT register to the ICDRS register, and the TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RIIC continues in master transmit mode.
Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
For data transmission with an address in the 10-bit format, start by writing 1111 0b, the 2 higher-order bits of the slave address, and W to the ICDRT register as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to the ICDRT register.
(4) After confirming that the ICSR2.TDRE flag is 1, write the data for transmission to the ICDRT register. The RIIC automatically holds the SCLn line low until the data for transmission are ready or a stop condition is issued.
(5) After all bytes of data for transmission have been written to the ICDRT register, wait until the value of the ICSR2.TEND flag returns to 1, and then set the ICCR2.SP bit to 1 (stop condition issuance request). Upon receiving a stop condition issuance request, the RIIC issues the stop condition.
(6) Upon detecting the stop condition, the RIIC automatically sets bits MST and TRS in the ICCR2 register to 00b and enters slave receive mode. Furthermore, it automatically sets the TDRE and TEND flags to 0, and sets the ICSR2.STOP flag to 1.
(7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

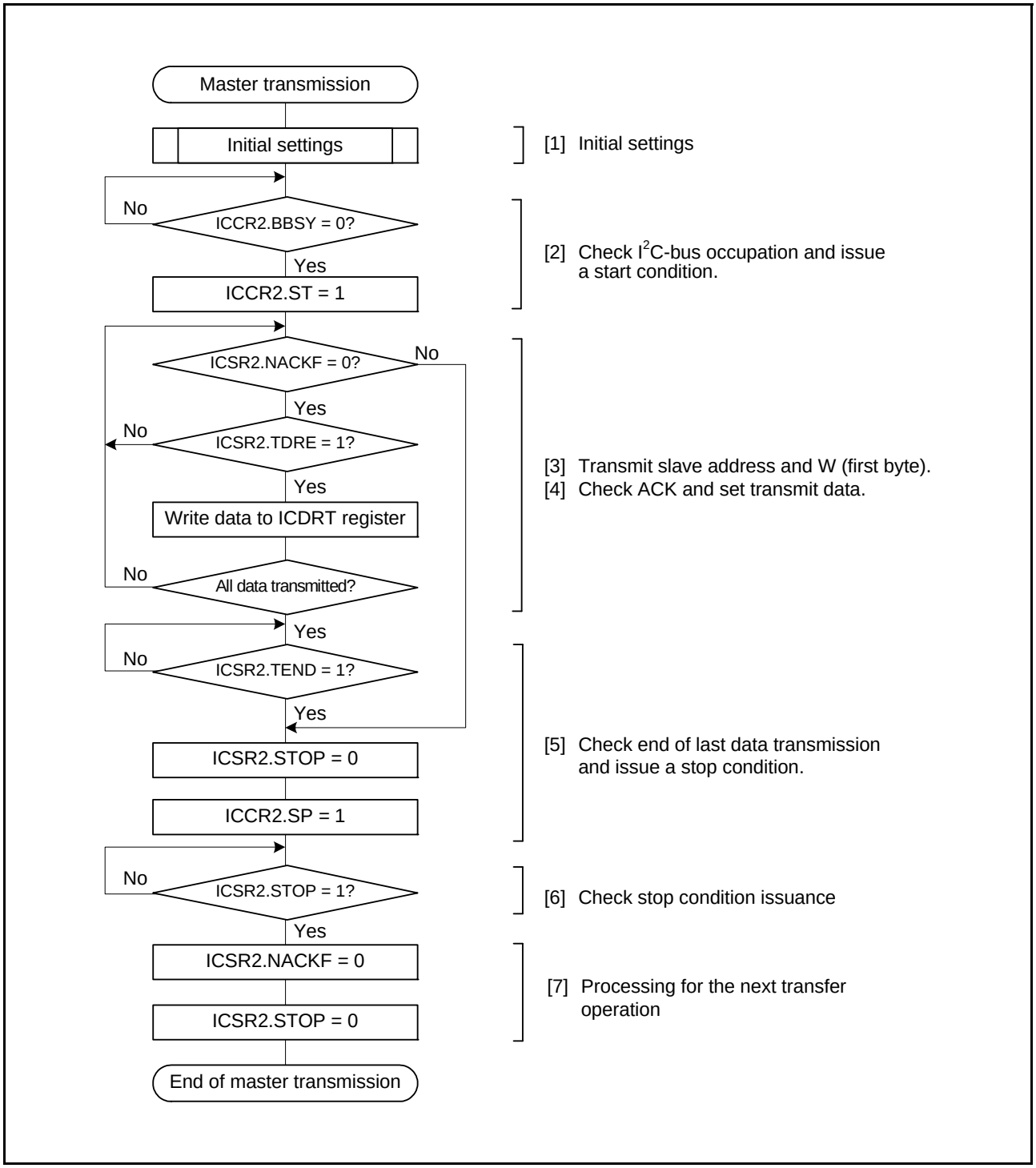


Figure 42.6 Example of Master Transmission Flowchart

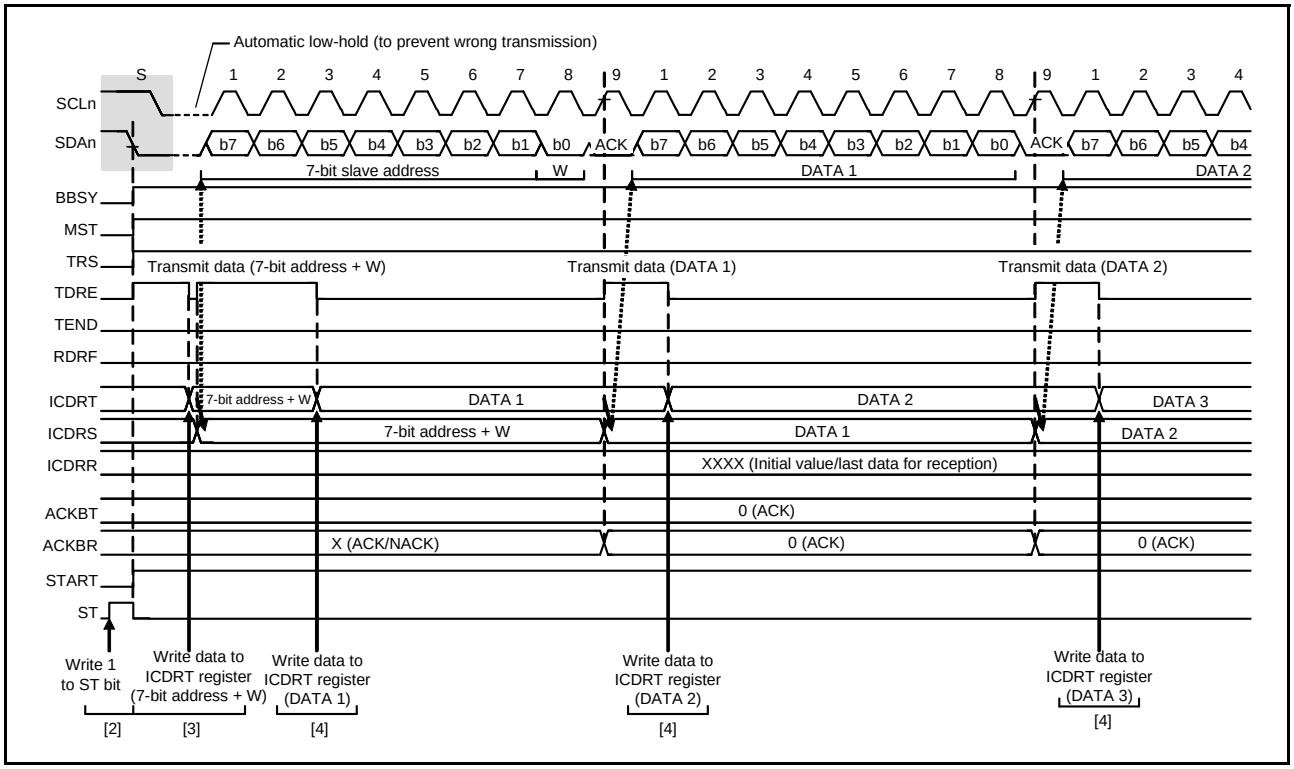


Figure 42.7 Master Transmit Operation Timing (1) (7-Bit Address Format)

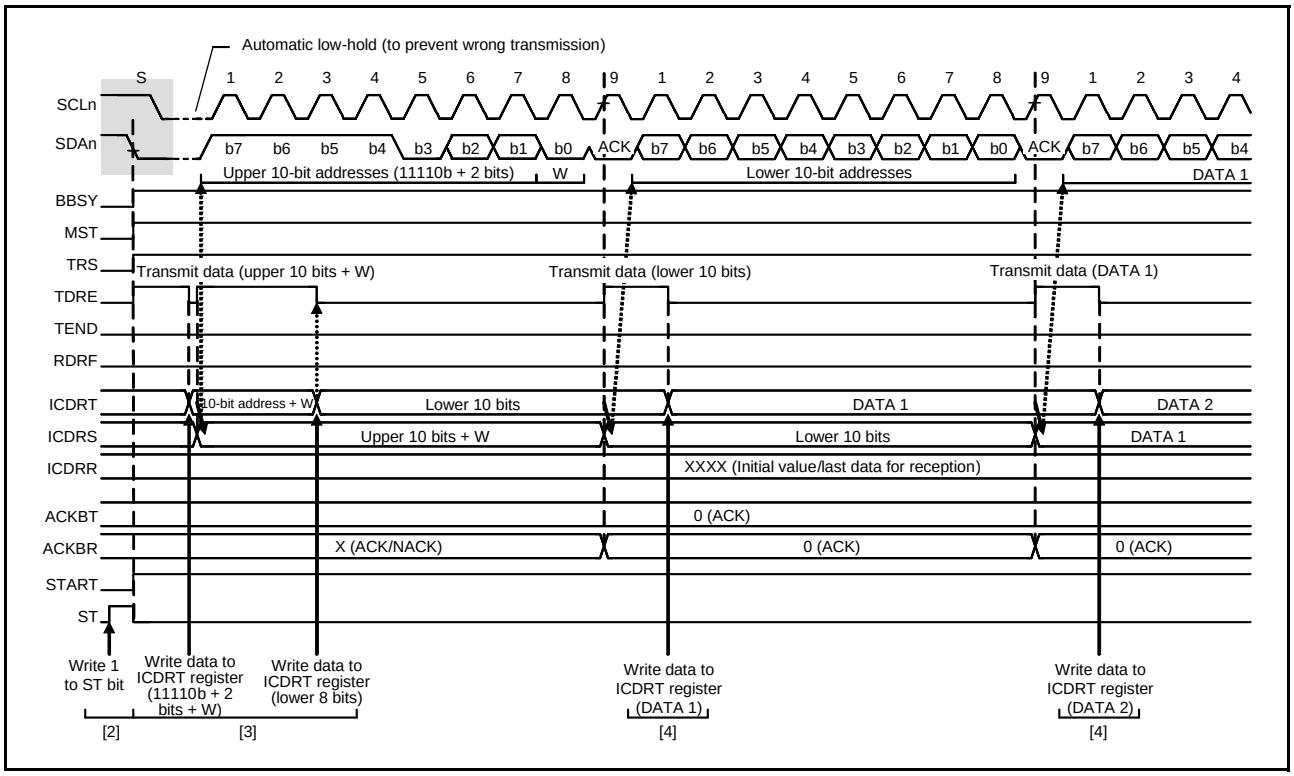


Figure 42.8 Master Transmit Operation Timing (2) (10-Bit Address Format)

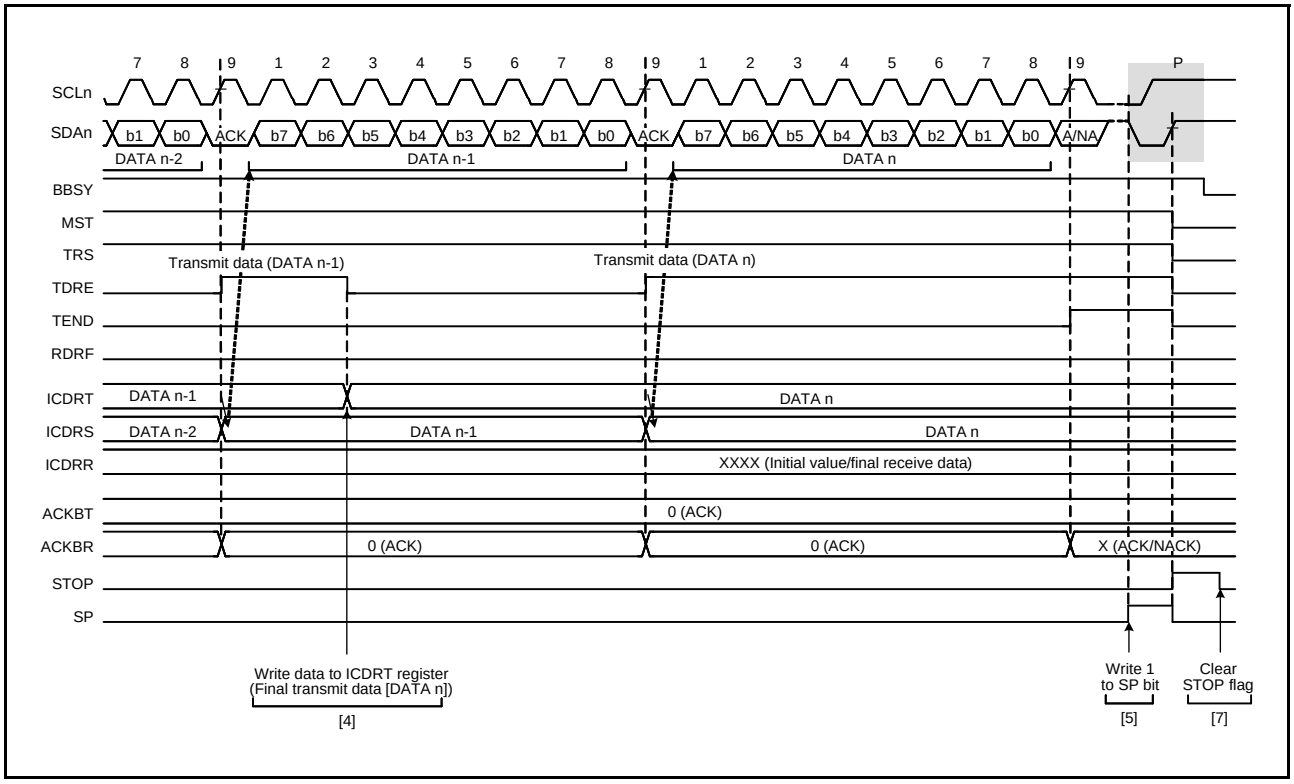


Figure 42.9 Master Transmit Operation Timing (3)

42.3.4 Master Receive Operation

In master receive operation, the RIIC as a master device outputs the SCL clock, receives data from the slave device, and returns acknowledgments. Because the RIIC must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.
Figure 42.10 and Figure 42.11 show examples of usage of master reception (7-bit address format) and Figure 42.12 to Figure 42.14 show the timing of operations in master reception.
The following describes the procedure and operations for master reception.

(1) Initial settings. For details, refer to section 42.3.2, Initial Settings.

(2) Read the ICCR2.BBSY flag to check that the bus is open, and then set the ICCR2.ST bit to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. When the RIIC detects the start condition, the BBSY flag and the ICSR2.START flag are automatically set to 1 and the ST bit is automatically set to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDAn line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and bits MST and TRS in the ICCR2 register are automatically set to 1, placing the RIIC in master transmit mode. The ICSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.

(3) Check that the ICSR2.TDRE flag is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to the ICDRT register. Once the data for transmission are written to the ICDRT register, the TDRE flag is automatically set to 0, the data are transferred from the ICDRT register to the ICDRS register, and the TDRE flag is again set to 1. Once the byte containing the slave address and R/W# bit has been transmitted, the value of the ICCR2.TRS bit is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the TRS bit is set to 0 on the rising edge of the ninth cycle of SCL clock, placing the RIIC in master receive mode. At this time, the TDRE flag is set to 0 and the ICSR2.RDRF flag is automatically set to 1.

Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a restart condition. After that, transmitting 1111 0b, the two higher-order bits of the slave address, and the R bit places the RIIC in master receive mode.

(4) Dummy read the ICDRR register after confirming that the ICSR2.RDRF flag is 1; this makes the RIIC start output of the SCL clock and start data reception.

(5) After 1 byte of data has been received, the ICSR2.RDRF flag is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the ICMR3.RDRFS bit. Reading the ICDRR register at this time will produce the received data, and the RDRF flag is automatically set to 0 at the same time. Furthermore, the value of the acknowledgment field received during the ninth cycle of SCL clock is returned as the value set in the ICMR3.ACKBT bit. Furthermore, if the next byte to be received is the next to last byte, set the ICMR3.WAIT bit to 1 (for wait insertion) before reading the ICDRR register (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the ICMR3.ACKBT bit to 1 (NACK) in step (6), due to other interrupts, etc., this fixes the SCLn line to the low level on the falling edge of the ninth clock cycle in reception of the last byte, so the state is such that issuing a stop condition is possible.

(6) When the ICMR3.RDRFS bit is 0 and the slave device must be notified that it is to end transfer for data reception after transfer of the next (final) byte, set the ICMR3.ACKBT bit to 1 (NACK).

(7) After reading the byte before last from the ICDRR register, if the value of the ICSR2.RDRF flag is confirmed to be 1, write 1 to the ICCR2.SP bit (stop condition issuance request) and then read the last byte from the ICDRR register. When the ICDRR register is read, the RIIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is completed or the SCLn line is released from the low-hold state.

(8) Upon detecting the stop condition, the RIIC automatically sets bits MST and TRS in the ICCR2 register to 00b and enters slave receive mode. Furthermore, detection of the stop condition leads to setting of the ICSR2.STOP flag to 1.

(9) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

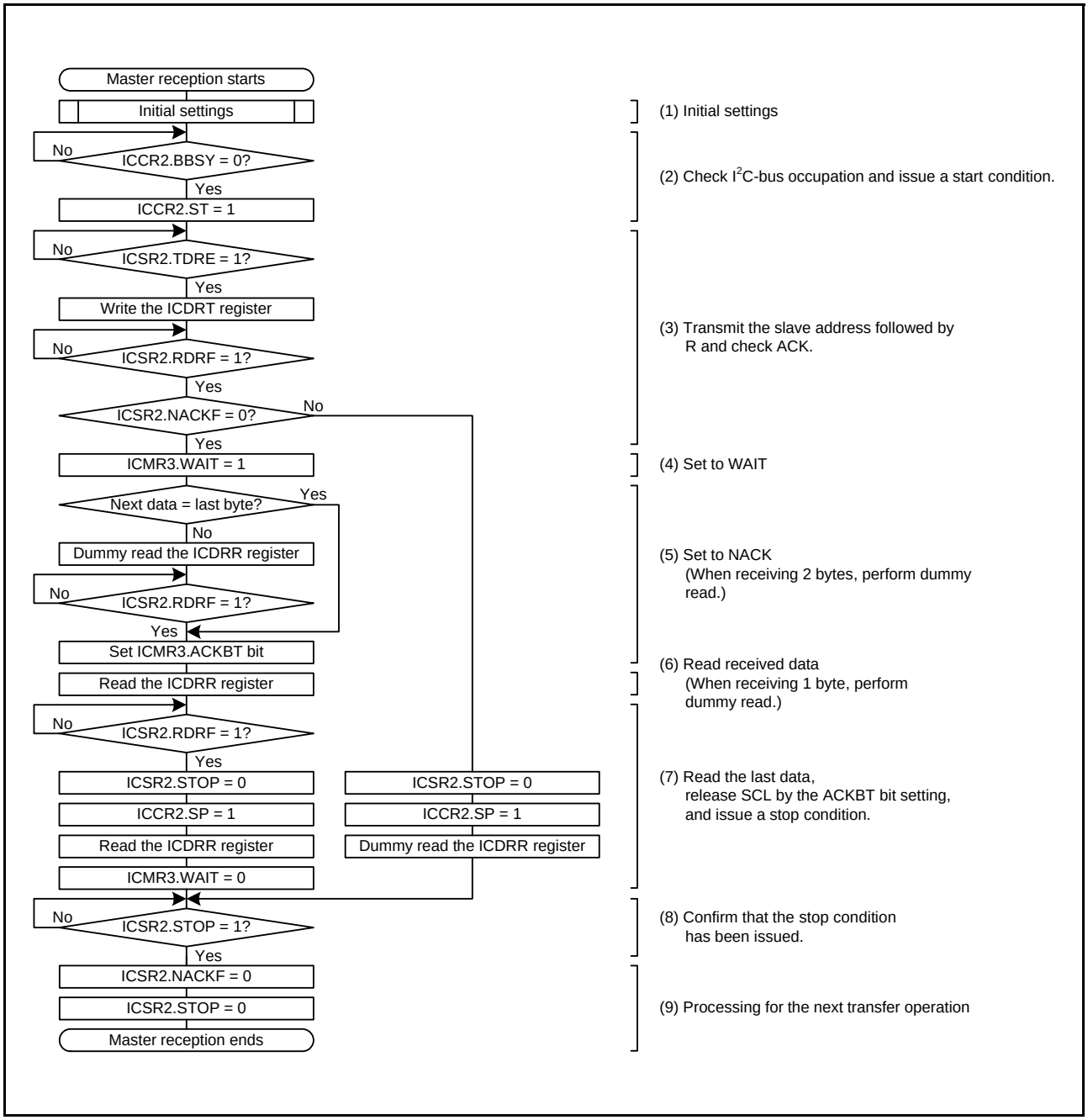


Figure 42.10 Example of Master Reception (7-Bit Address Format, 1 or 2 bytes)

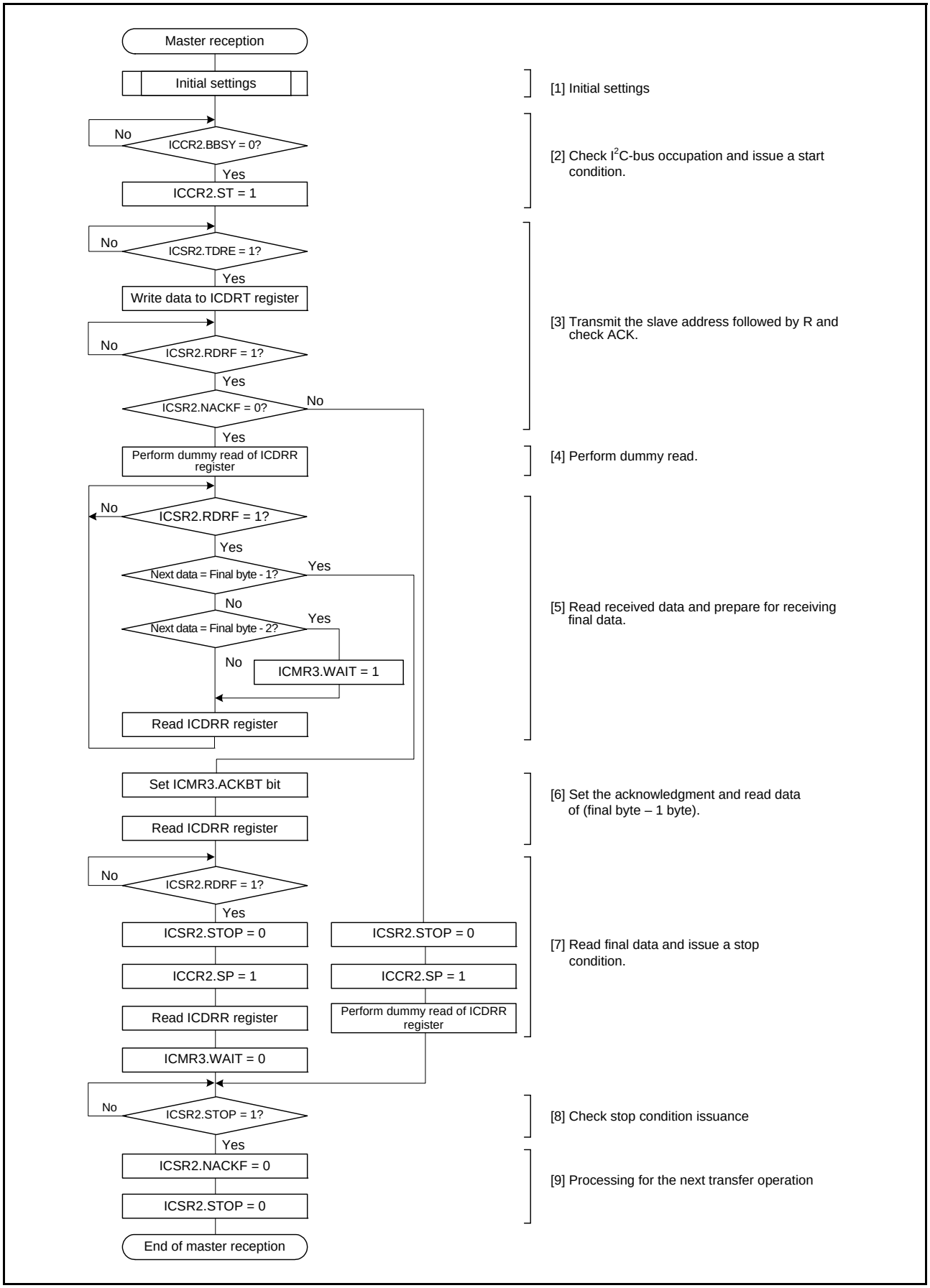


Figure 42.11 Example of Master Reception (7-Bit Address Format, 3 Bytes or More)

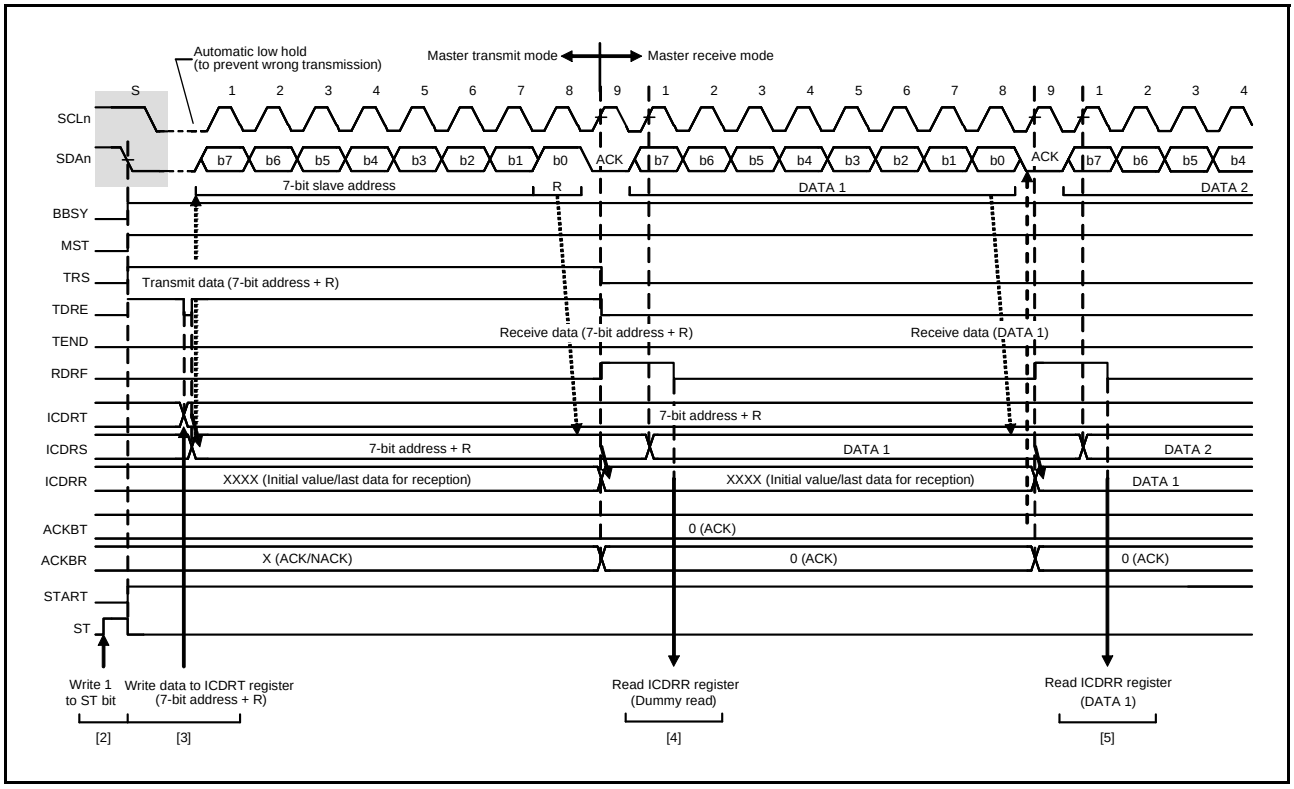


Figure 42.12 Master Receive Operation Timing (1) (7-Bit Address Format, When RDRFS bit is 0)

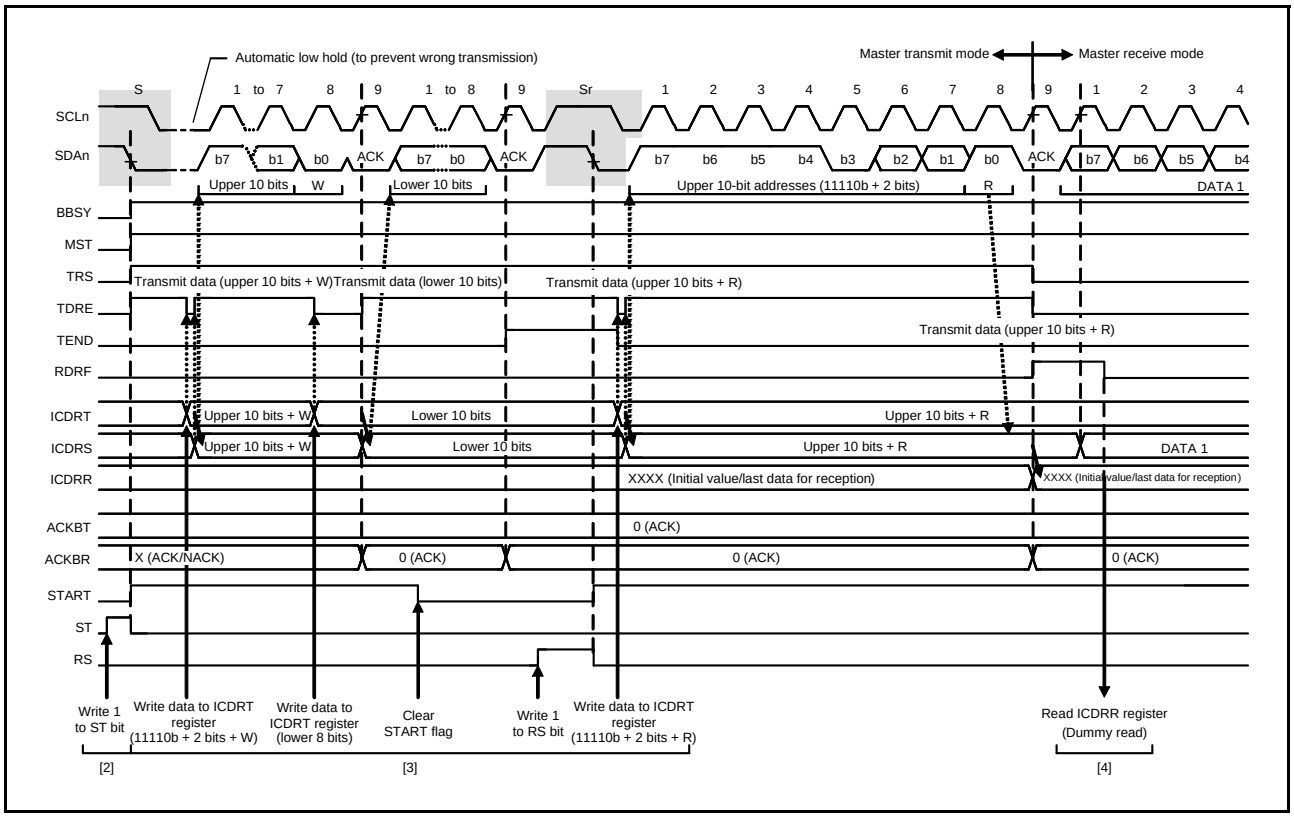


Figure 42.13 Master Receive Operation Timing (2) (10-Bit Address Format, When RDRFS bit is 0)

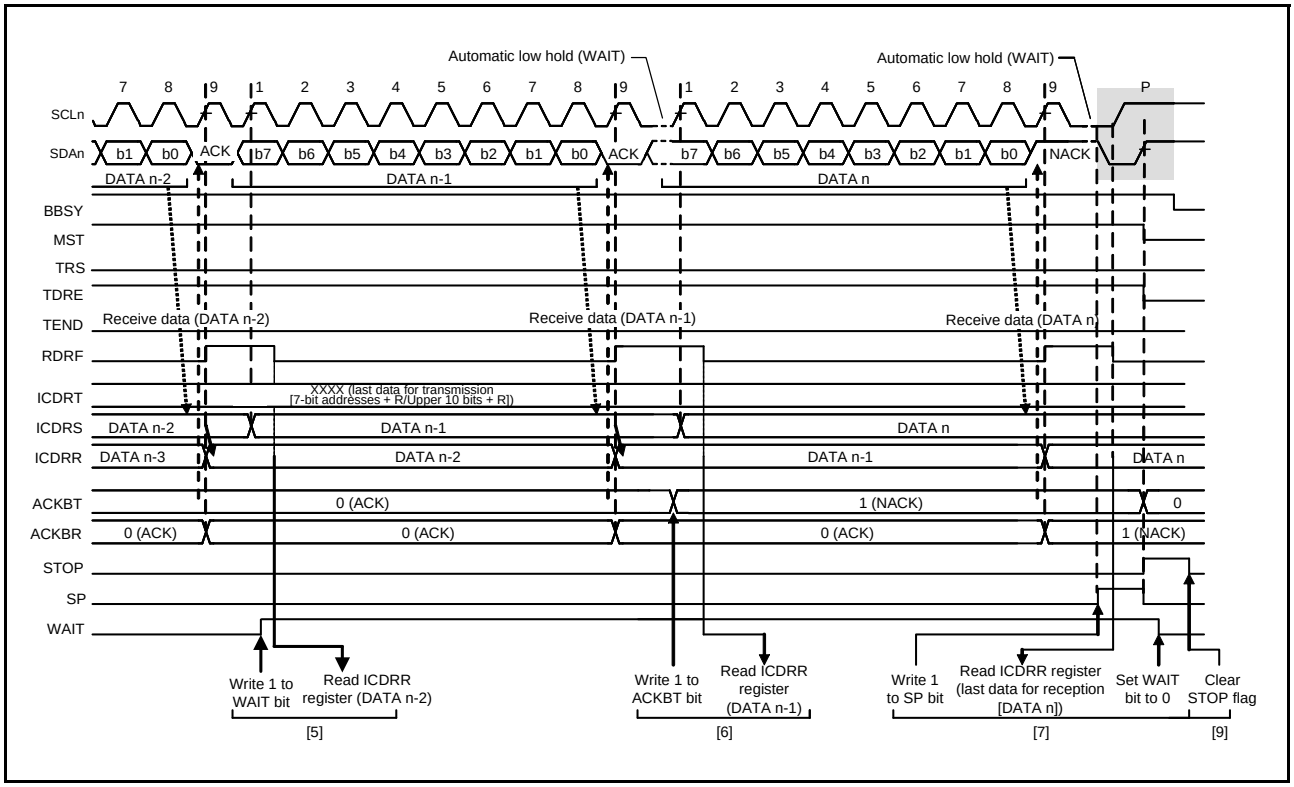


Figure 42.14 Master Receive Operation Timing (3) (When RDRFS bit is 0)

42.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL clock, the RIIC transmits data as a slave device, and the master device returns acknowledgments.

Figure 42.15 shows an example of usage of slave transmission and Figure 42.16 and Figure 42.17 show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

(1) Initial settings. For details, refer to section 42.3.2, Initial Settings.
After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 1, the RIIC automatically places itself in slave transmit mode by setting both the ICCR2.TRS bit and the ICSR2.TDRE flag to 1.

(3) After the ICSR2.TDRE flag is confirmed to be 1, write the data for transmission to the ICDRT register. At this time, if the RIIC receives no acknowledge from the master device (receives an NACK signal) while the ICFER.NACKE bit is 1, the RIIC suspends transfer of the next data.

(4) Wait until the ICSR2.TEND flag is set to 1 while the ICSR2.TDRE flag is 1, after the ICSR2.NACKF flag is set to 1 or the last byte for transmission is written to the ICDRT register. When the ICSR2.NACKF flag or the TEND flag is 1, the RIIC drives the SCLn line low on the ninth falling edge of SCL clock.

(5) When the ICSR2.NACKF flag or the ICSR2.TEND flag is 1, dummy read the ICDRR register to complete the processing. This releases the SCLn line.

(6) Upon detecting the stop condition, the RIIC automatically sets bits ICSR1.HOA, GCA, and AASy (y = 0 to 2), flags ICSR2.TDRE and TEND, and the ICCR2.TRS bit to 0, and enters slave receive mode.

(7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

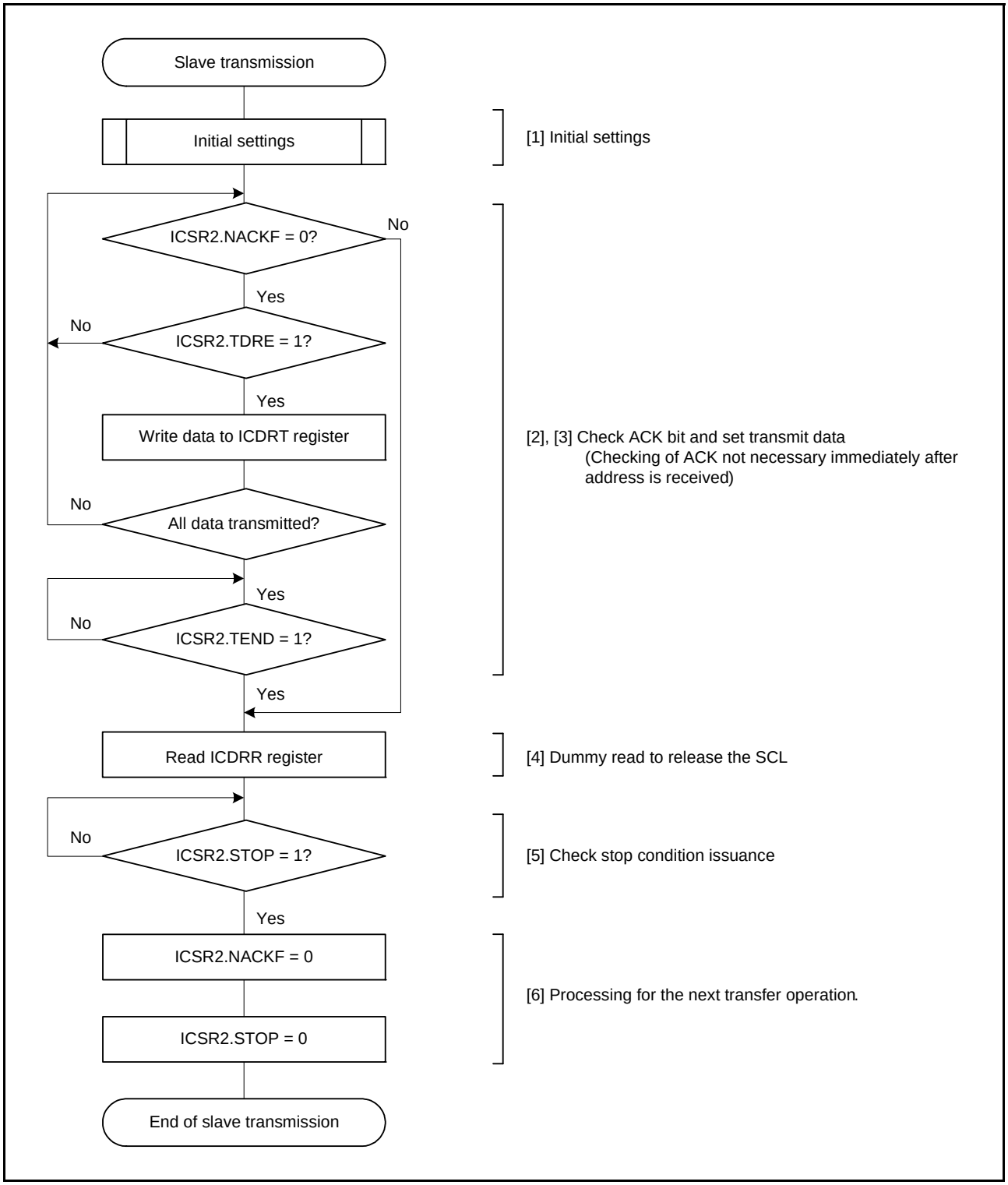


Figure 42.15 Example of Slave Transmission Flowchart

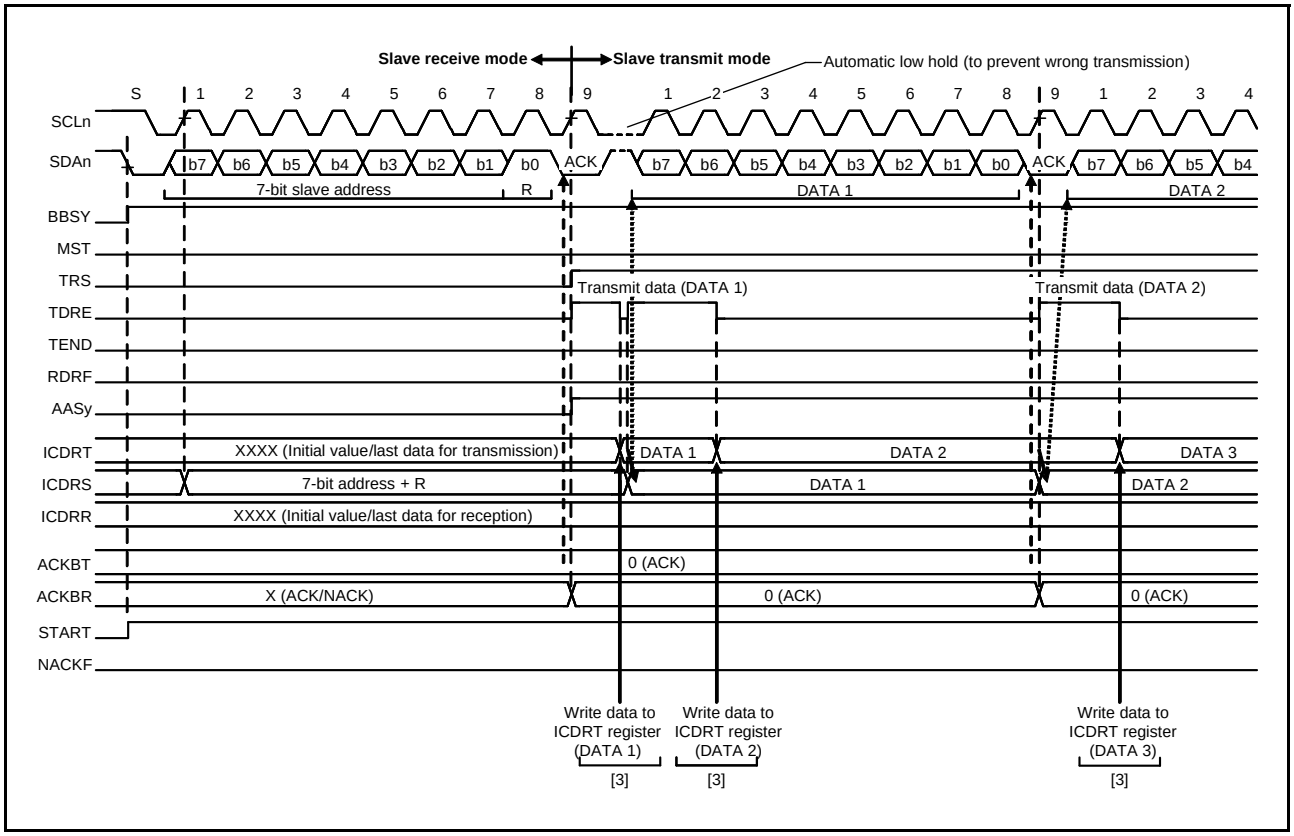


Figure 42.16 Slave Transmit Operation Timing (1) (7-Bit Address Format)

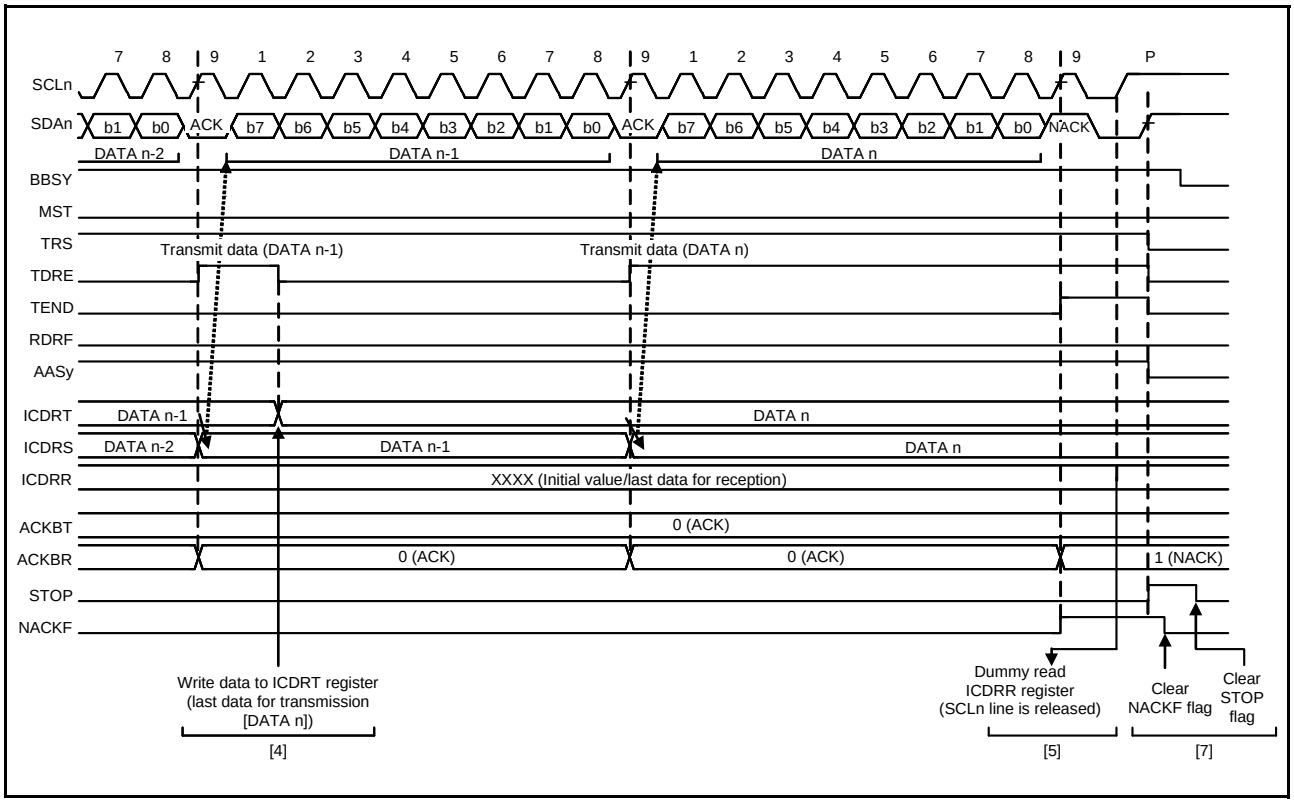


Figure 42.17 Slave Transmit Operation Timing (2)

42.3.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the RIIC returns acknowledgments as a slave device.

Figure 42.18 shows an example of usage of slave reception and Figure 42.19 and Figure 42.20 show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

(1) Initial settings. For details, refer to section 42.3.2, Initial Settings.
After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 0, the RIIC continues to place itself in slave receive mode and sets the ICSR2.RDRF flag to 1.

(3) After the ICSR2.STOP flag is confirmed to be 0 and the ICSR2.RDRF flag to be 1, dummy read the ICDRR register (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8 bits when the 10-bit address format is selected).

(4) When the ICDRR register is read, the RIIC automatically sets the ICSR2.RDRF flag to 0. If reading of the ICDRR register is delayed and a next byte is received while the RDRF flag is still set to 1, the RIIC holds the SCLn line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading the ICDRR register releases the SCLn line from being held at the low level.
When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1, read the ICDRR register until all the data is completely received.

(5) Upon detecting the stop condition, the RIIC automatically clears bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 0.

(6) After checking that the ICSR2.STOP flag is 1, set the ICSR2.STOP flag to 0 for the next transfer operation.

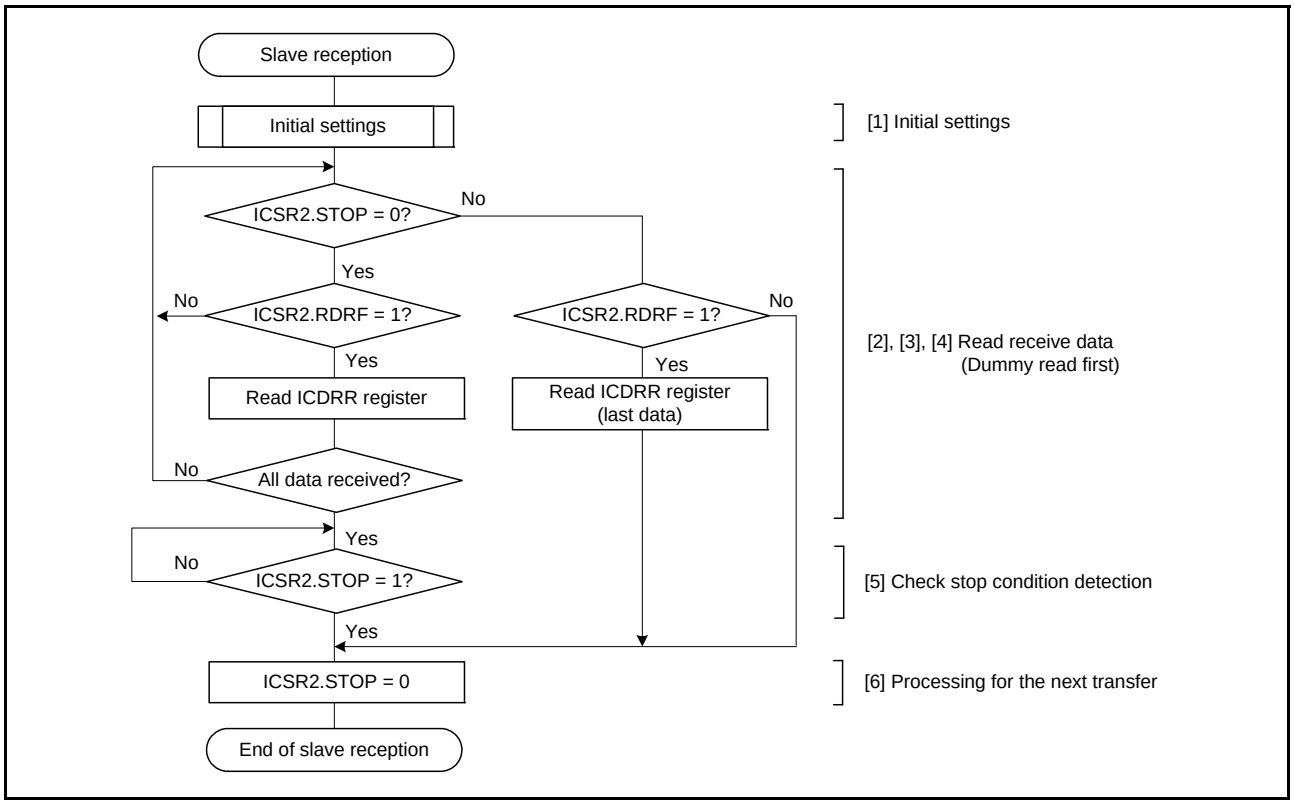


Figure 42.18 Example of Slave Reception Flowchart

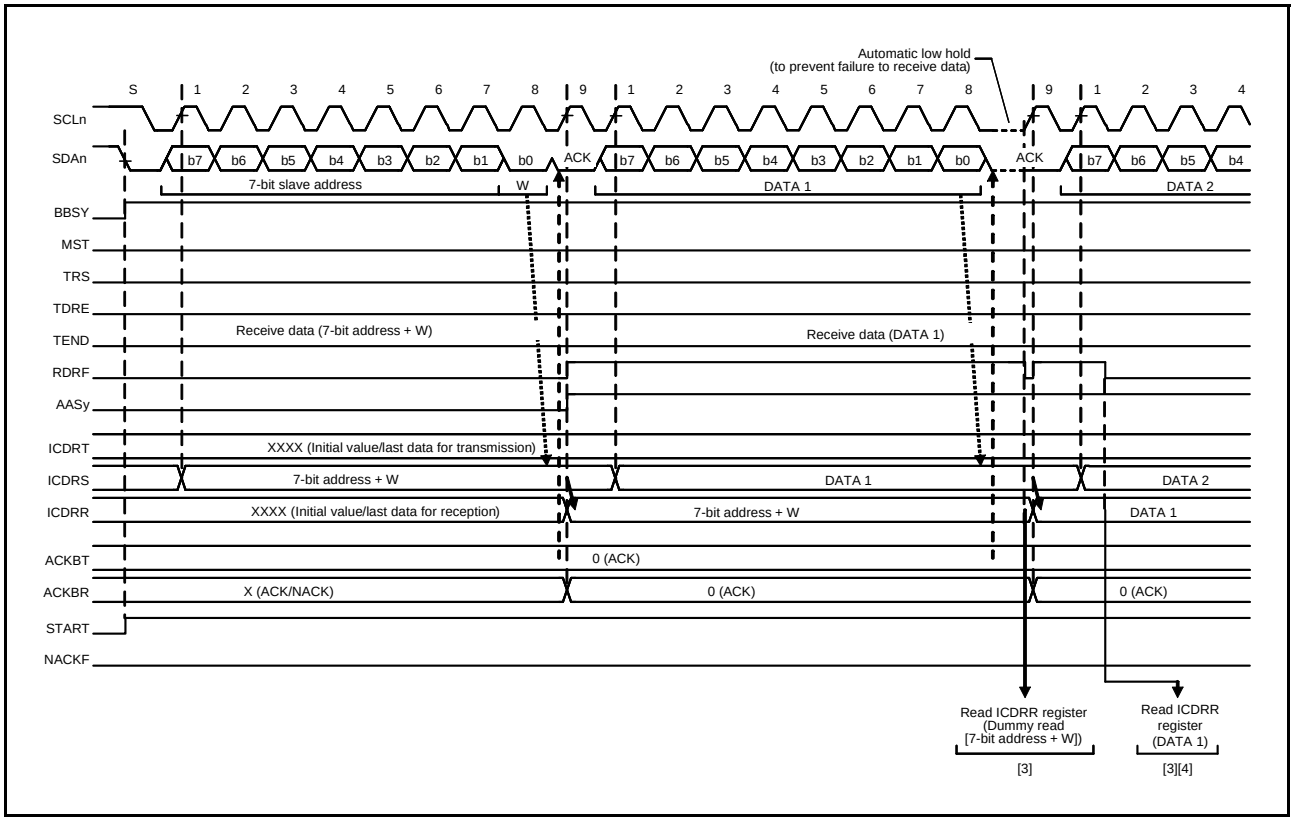


Figure 42.19 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS bit is 0)

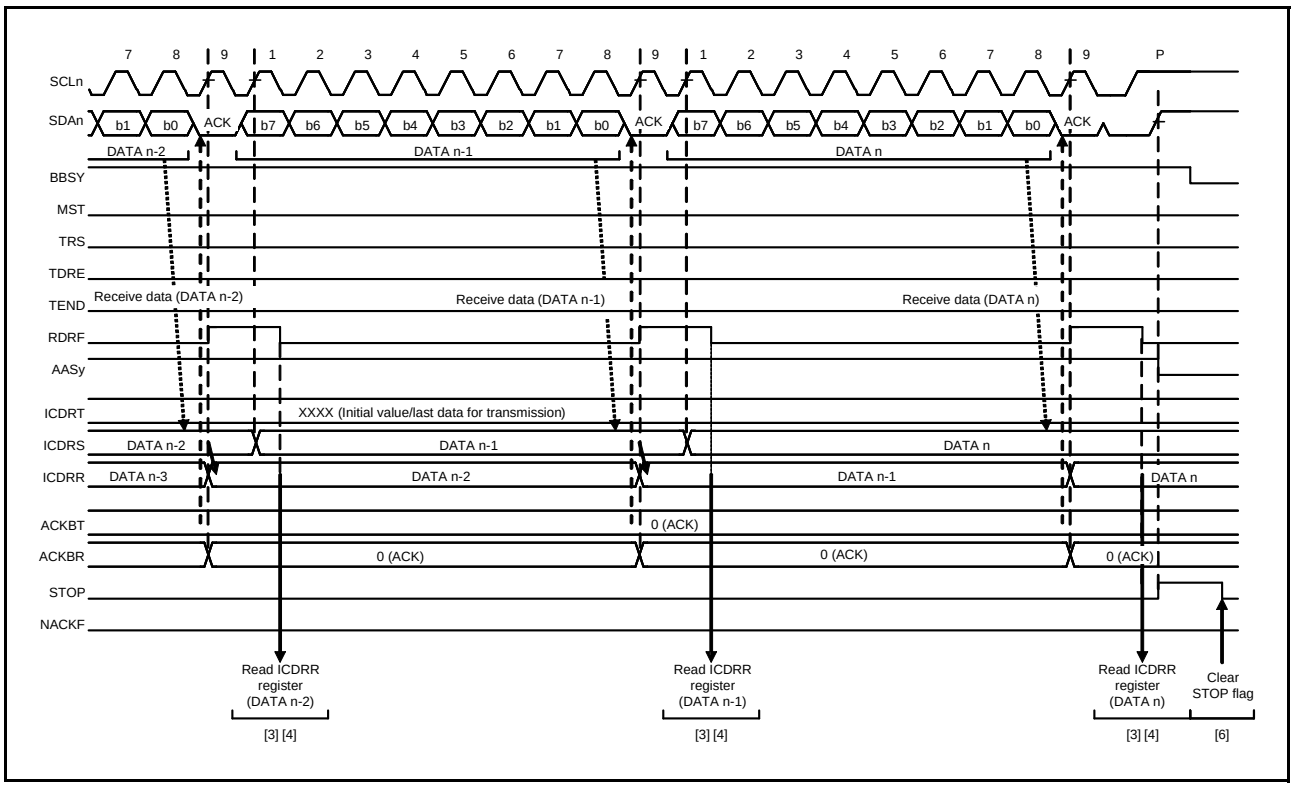


Figure 42.20 Slave Receive Operation Timing (2) (when RDRFS bit is 0)

42.4 SCL Synchronization Circuit

In generation of the SCL clock, the RIIC starts counting out the value for width at high level specified in the ICBRH register when it detects a rising edge on the SCLn line and drives the SCLn line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCLn line, it starts counting out the width at low level period specified in the ICBRL register, and then stops driving the SCLn line (releases the line) once counting of the width at low level is complete. The SCL clock is thus generated.

If multiple master devices are connected to the I2C-bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Because this synchronization of SCL signals must be bit by bit, the RIIC is equipped with a facility (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCLn line while in master mode.

When the RIIC has detected a rising edge on the SCLn line and thus started counting out the width at high level specified in the ICBRH register, and the level on the SCLn line falls because an SCL signal is being generated by another master device, the RIIC stops counting when it detects the falling edge, drives the level on the SCLn line low, and starts counting out the width at low level specified in the ICBRL register. When the RIIC finishes counting out the width at low level, it stops driving the SCLn line to the low level (i.e. releases the line). At this time, if the width at low level of the SCL clock signal from the other master device is longer than the width at low level set in the RIIC, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the SCLn line has been released. When the RIIC finishes outputting the low-level period of the SCL clock, the SCLn line is released and the SCL clock rises. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the ICFER.SCLE bit is set to 1.

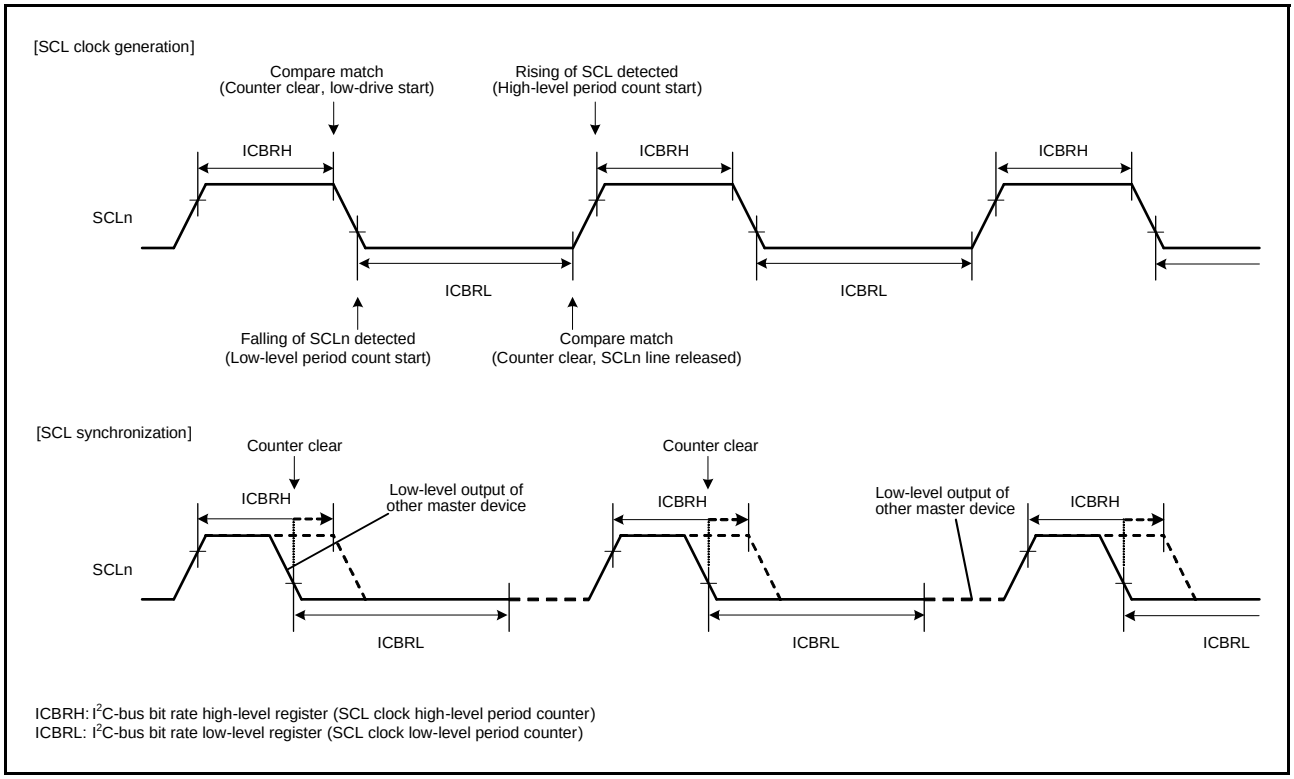


Figure 42.21 Generation and Synchronization of the SCL Signal from the RIIC

42.5 SDA Output Delay Function

The RIIC module incorporates a function for delaying output on the SDA line. The delay can be applied to all output (issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals) on the SDA line.

With the SDA output delay function, SDA output is delayed from detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval over which the SCL clock is at the low level. Doing this leads to usage with the aim of preventing erroneous operation of communications devices, with the aim of satisfying the 300-ns (min.) data-hold time requirement of the SMBus specification.

The output delay function is enabled by setting the ICMR2.SDDL[2:0] bits to any value other than 000b, and disabled by setting the same bits to 000b.

While the SDA output delay function is enabled (i.e. while the ICMR2.SDDL[2:0] bits are set to any value other than 000b), the ICMR2.DLCS bit selects the clock source for counting by the SDA output delay counter as the internal base clock (IICφ) for the RIIC module or as a clock signal derived by dividing the frequency of the internal base clock by two (IICφ/2). The counter counts the number of cycles set in the ICMR2.SDDL[2:0] bits. After counting of the set number of cycles of delay is completed, the RIIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.

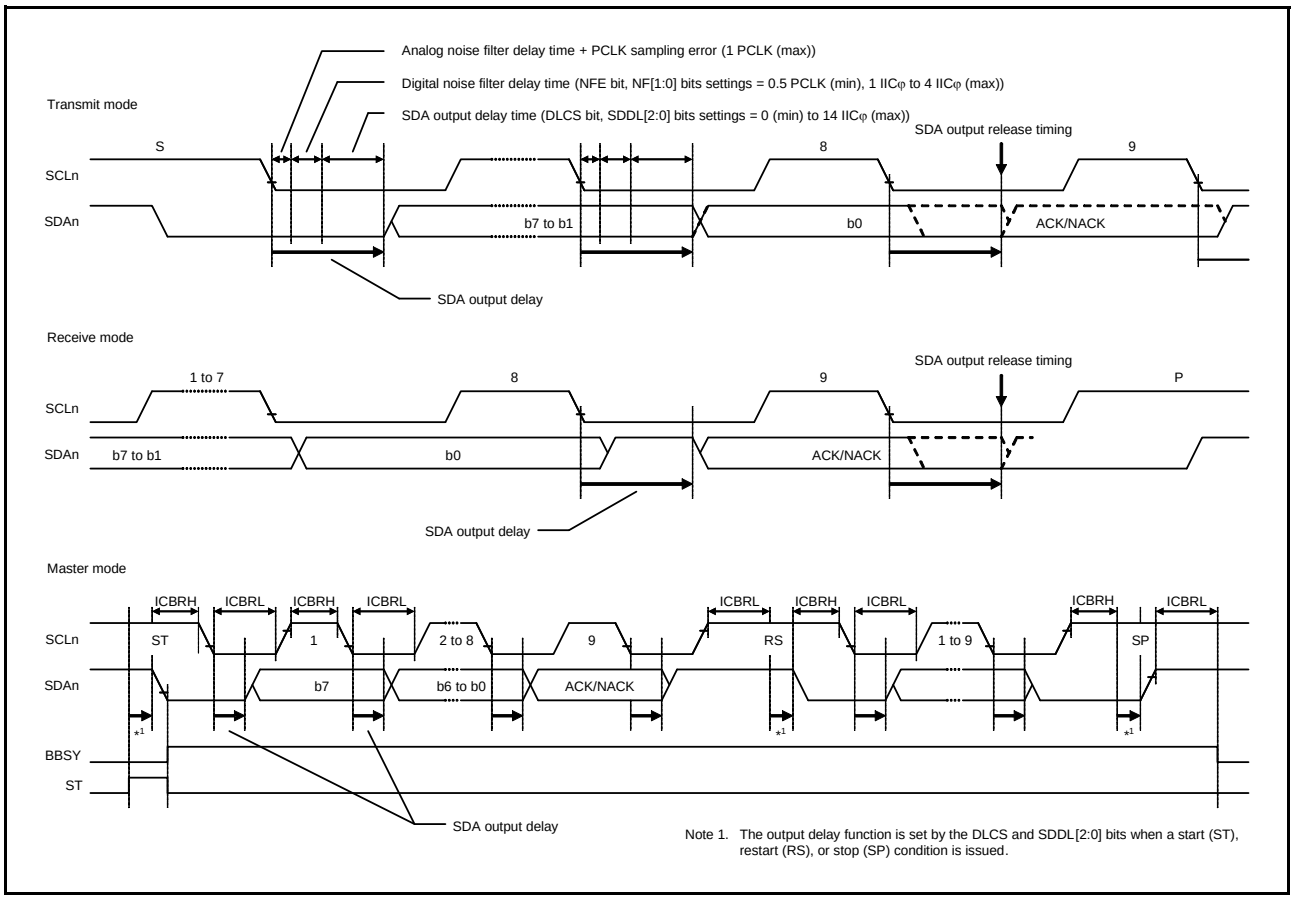


Figure 42.22 SDA Output Delay Function

42.6 Digital Noise Filter Circuit

The states of the SCLn and SDAn pins are conveyed to the internal circuitry through analog noise-filter and digital noise-filter circuits. Figure 42.23 is a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the RIIC consists of four flip-flop circuit stages connected in series and a match-detection circuit.

The number of effective stages in the digital noise filter is selected by the ICMR3.NF[1:0] bits. The selected number of effective stages determines the noise-filtering capability as a period from one to four IICφ cycles.

The input signal to the SCLn pin (or SDAn pin) is sampled on falling edges of the IICφ signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected by the ICMR3.NF[1:0] bits, the signal level is conveyed to the subsequent stage. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (PCLK) and the transfer rate is small (e.g. data transfer at 400 kbps with PCLK = 4 MHz), the characteristics of the digital noise filter may lead to the elimination of needed signals as noise. In such cases, it is possible to disable the digital noise-filter circuit (by setting the ICFER.NFE bit to 0) and use only the analog noise filter circuit.

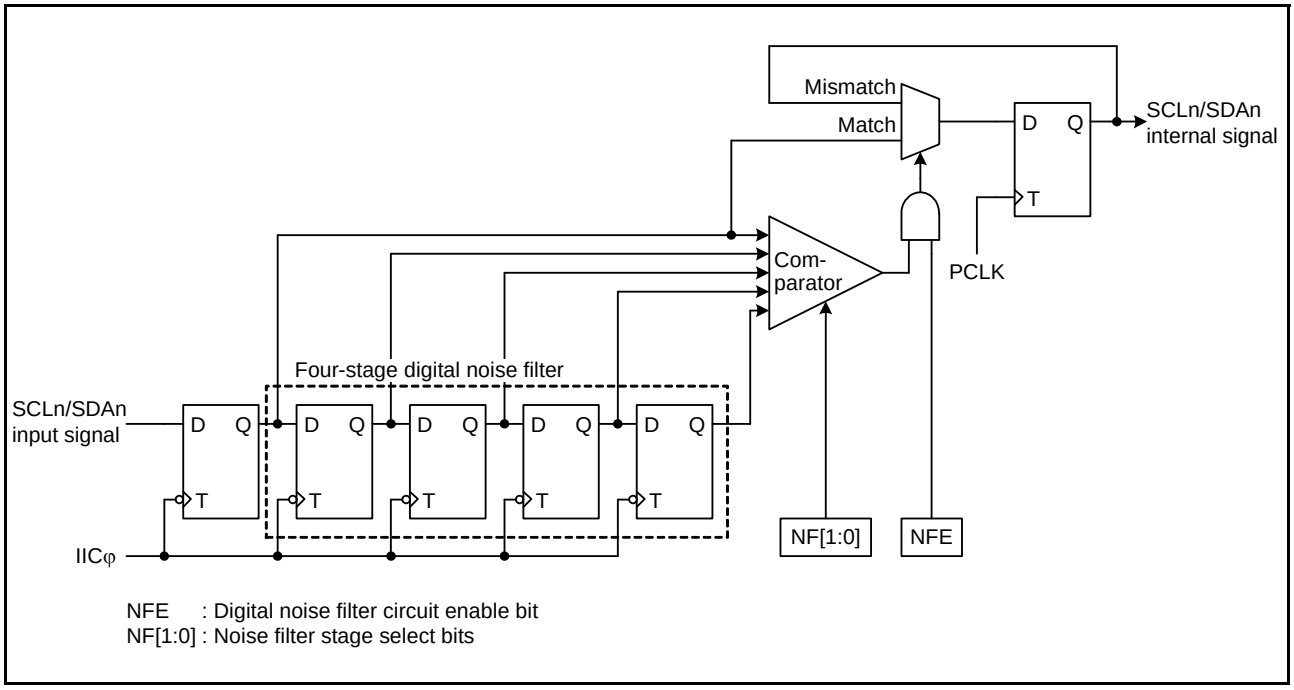


Figure 42.23 Block Diagram of Digital Noise Filter Circuit

42.7 Address Match Detection

The RIIC can set three unique slave addresses in addition to the general call address and host address, and also can set 7-bit or 10-bit slave addresses.

42.7.1 Slave-Address Match Detection

The RIIC can set three unique slave addresses, and has a slave address detection function for each unique slave address. When the ICSER.SARyE bit (y = 0 to 2) is set to 1, the slave addresses set in registers SARUy and SARLy (y = 0 to 2) can be detected.

When the RIIC detects a match of the set slave address, the corresponding ICSR1.AASy flag (y = 0 to 2) is set to 1 at the rising edge of the ninth SCL clock cycle, and the ICSR2.RDRF flag or the ICSR2.TDRE flag is set to 1 by the following R/W# bit. This causes a receive data full interrupt (RXI) or transmit data empty interrupt (TXI) to be generated. The AASy flag is used to identify which slave address has been specified.

Figure 42.24 to Figure 42.26 show the AASy flag set timing in three cases.

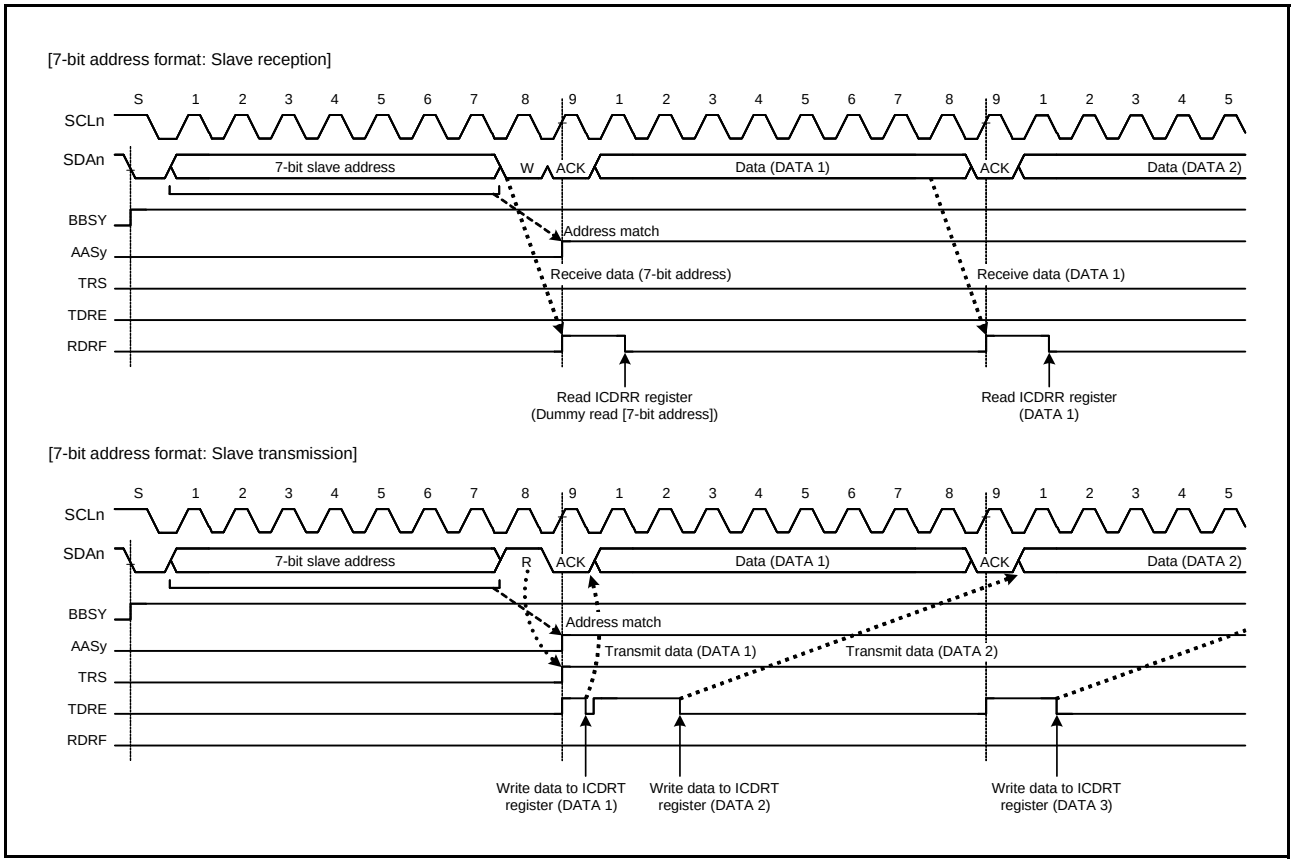


Figure 42.24 AASy Flag Set Timing with 7-Bit Address Format Selected

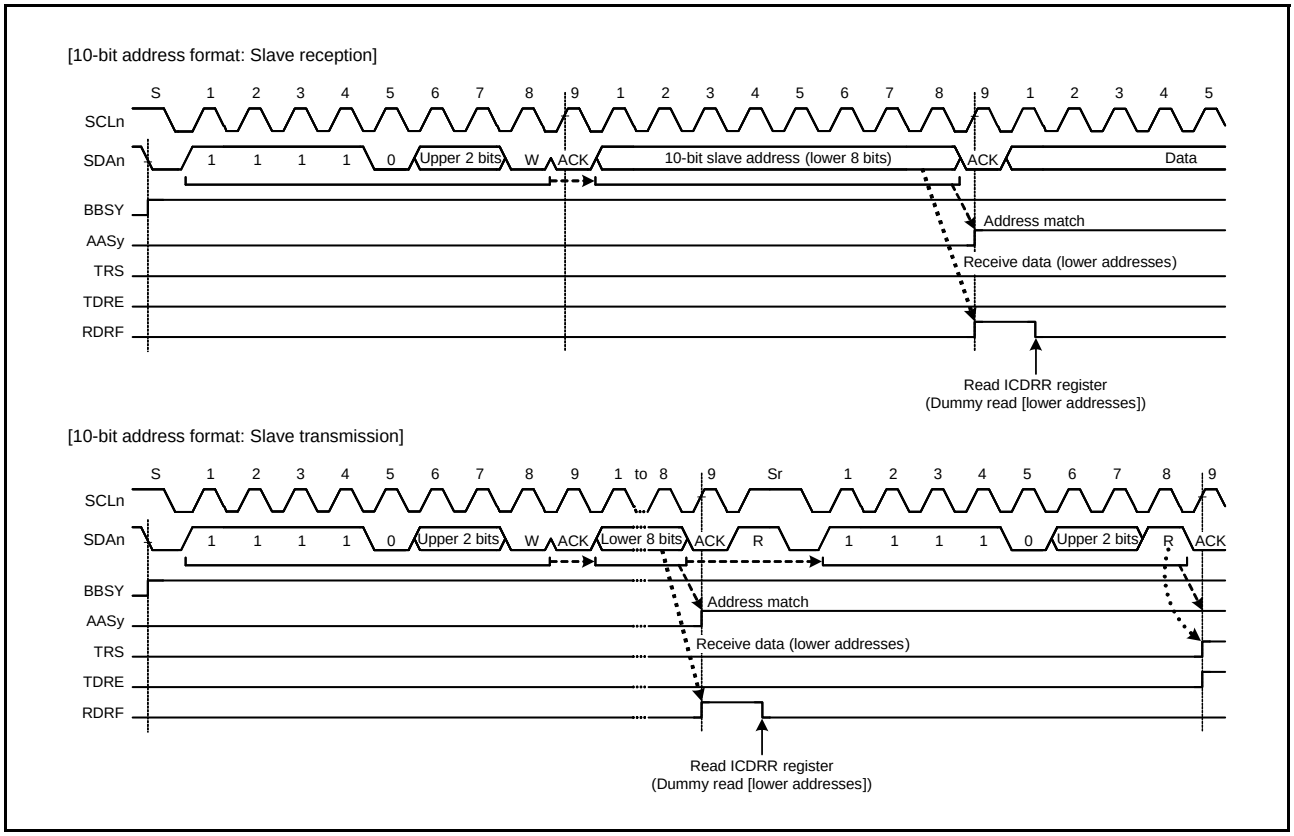


Figure 42.25 AASy Flag Set Timing with 10-Bit Address Format Selected

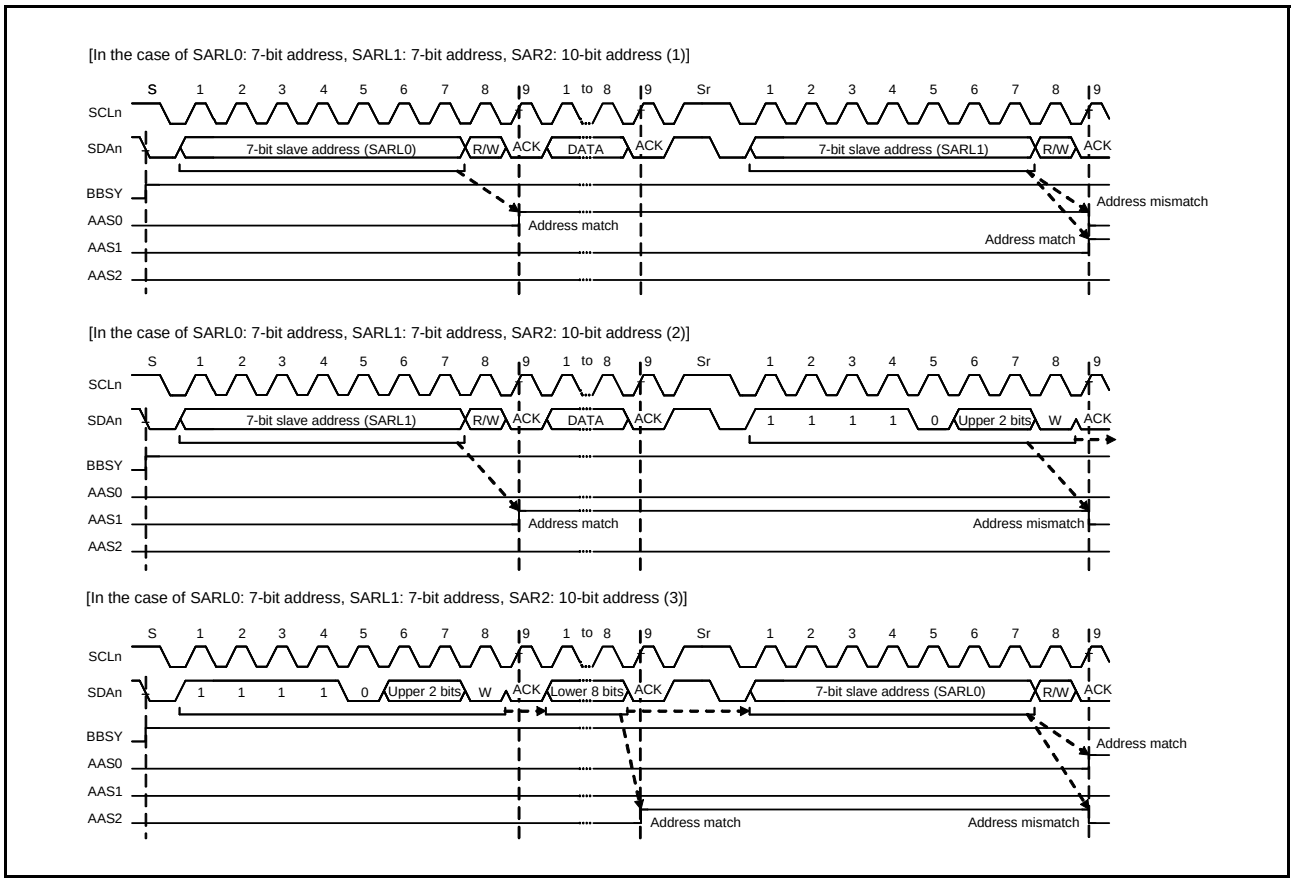


Figure 42.26 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

42.7.2 Detection of the General Call Address

The RIIC has a facility for detecting the general call address (0000 000b + 0 (write)). This is enabled by setting the ICSER.GCAE bit to 1.

If the address received after a start or restart condition is issued is 0000 000b + 1 (read) (start byte), the RIIC recognizes this as the address of a slave device with an “all-zero” address but not as the general call address.

When the RIIC detects the general call address, both the ICSR1.GCA flag and the ICSR2.RDRF flag are set to 1 on the rising edge of the ninth cycle of SCL clock. This leads to the generation of a receive data full interrupt (RXI). The value of the GCA flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

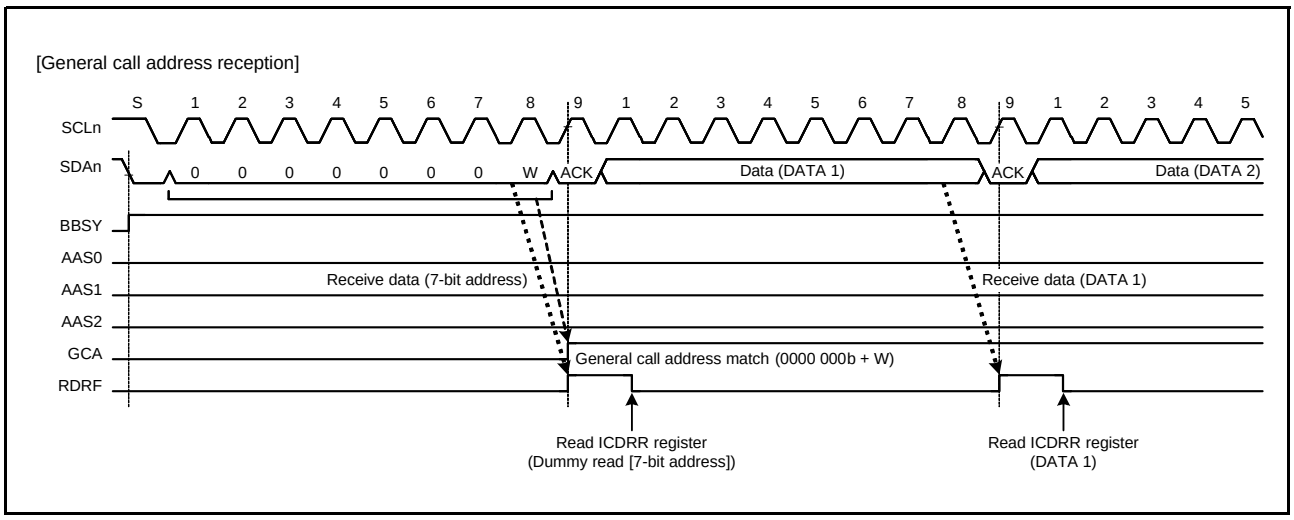


Figure 42.27 Timing of GCA Flag Setting during Reception of General Call Address

42.7.3 Device-ID Address Detection

The RIIC module has a facility for detecting device-ID addresses conformant with the I2C-bus specification (Rev. 03). When the RIIC receives 1111 100b as the first byte after a start condition or restart condition was issued with the ICSER.DIDE bit set to 1, the RIIC recognizes the address as a device ID, sets the ICSR1.DID flag to 1 on the rising edge of the eighth SCL clock cycle when the following R/W# bit is 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the RIIC sets the corresponding ICSR1.AASy flag (y = 0 to 2) to 1.

After that, when the first byte received after a start or restart condition is issued matches the device ID address (1111 100b) again and the following R/W# bit is 1, the RIIC does not compare the second and subsequent bytes and sets the ICSR2.TDRE flag to 1.

In the device-ID address detection function, the RIIC sets the DID flag to 0 if a match with the RIIC's own slave address is not obtained or a match with the device ID address is not obtained after a match with the RIIC's own slave address and the detection of a restart condition. If the first byte after detection of a start or restart condition matches the device ID address (1111 100b) and the R/W# bit is 0, the RIIC sets the DID flag to 1 and compares the second and subsequent bytes with the RIIC's slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RIIC does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DID flag after confirming that TDRE flag is 1.

Furthermore, prepare the device-ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details of the information that must be included in device-ID fields, contact NXP Semiconductors.

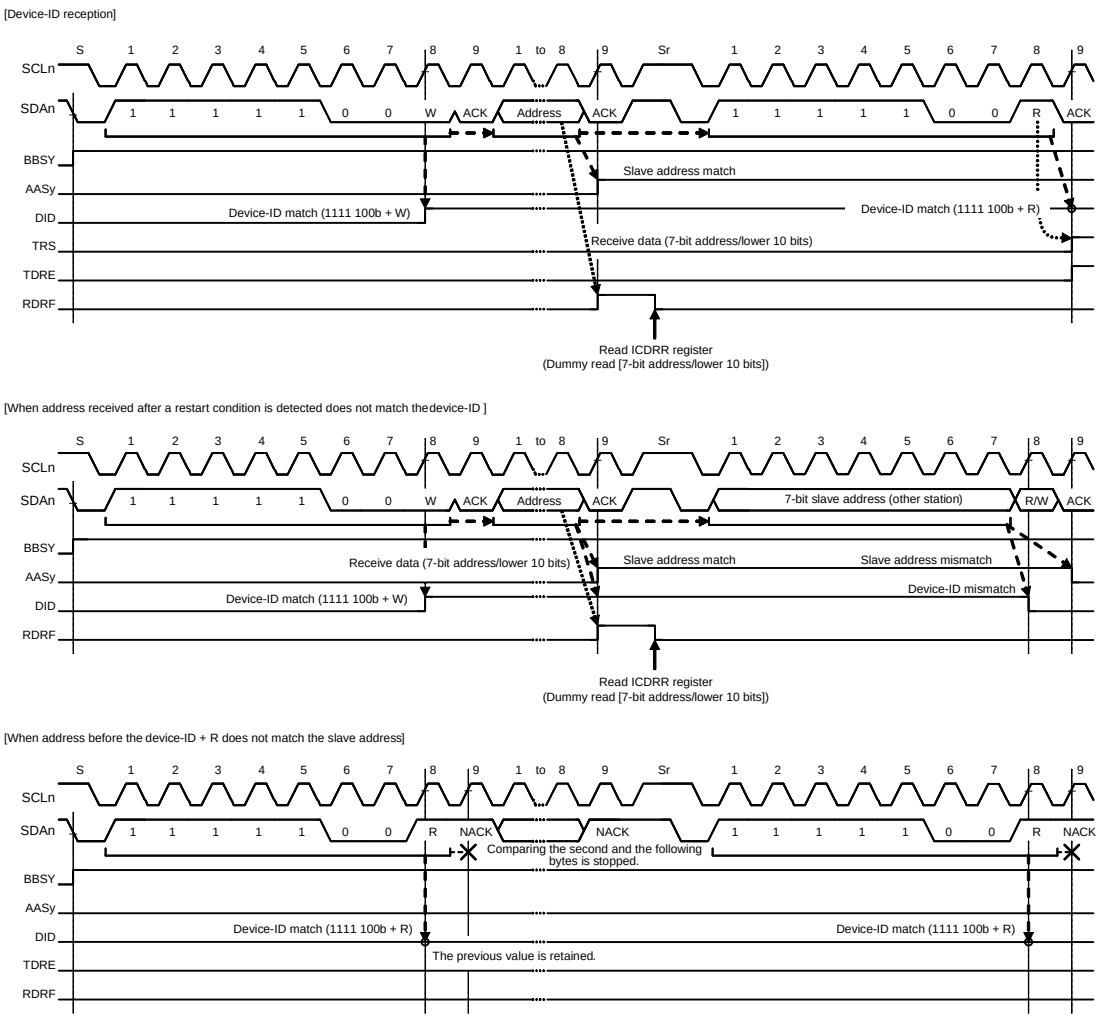


Figure 42.28 AASy/DID Flag Set/Clear Timing during Reception of Device-ID

42.7.4 Host Address Detection

The RIIC has a function to detect the host address while the SMBus is operating. When the ICSER.HOAE bit is set to 1 while the ICMR3.SMBS bit is 1, the RIIC can detect the host address (0001 000b) in slave receive mode (bits MST and TRS in the ICCR2 register are 00b).

When the RIIC detects the host address, the ICSR1.HOA flag is set to 1 at the rising edge of the ninth SCL clock cycle, and at the same time, the ICSR2.RDRF flag is set to 1 when the R/W# bit is 0 (Wr bit). This causes a receive data full interrupt (RXI) to be generated. The HOA flag is used to recognize that the host address was sent from the smart battery or other devices.

If the bit following the host address (0001 000b) is an Rd bit (R/W# bit is 1), the RIIC can also detect the host address. After the host address is detected, the RIIC operates in the same manner as normal slave operation.

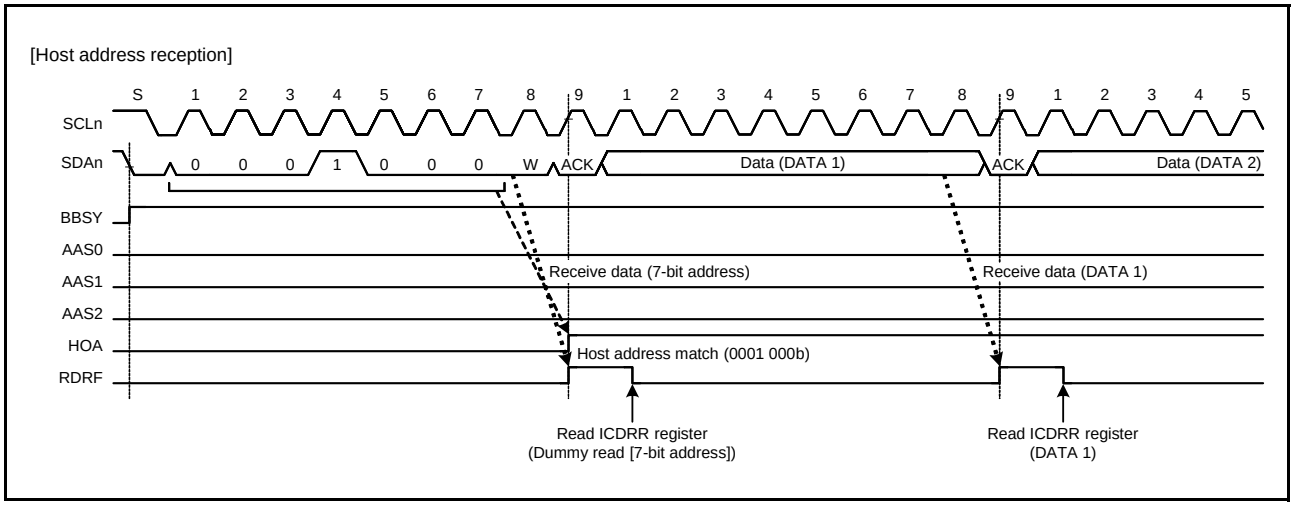


Figure 42.29 HOA Flag Set Timing during Reception of Host Address

42.8 Automatic Low-Hold Function for SCL

42.8.1 Function to Prevent Wrong Transmission of Transmit Data

If the shift register (ICDRS) is empty when data have not been written to the I2C-bus transmit data register (ICDRT) with the RIIC in transmission mode (ICCR2.TRS bit is 1), the SCLn line is automatically held at the low level over the intervals shown below. This low-hold period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

Master transmit mode

- Low-level interval after a start condition or restart condition is issued
- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

Slave transmit mode

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

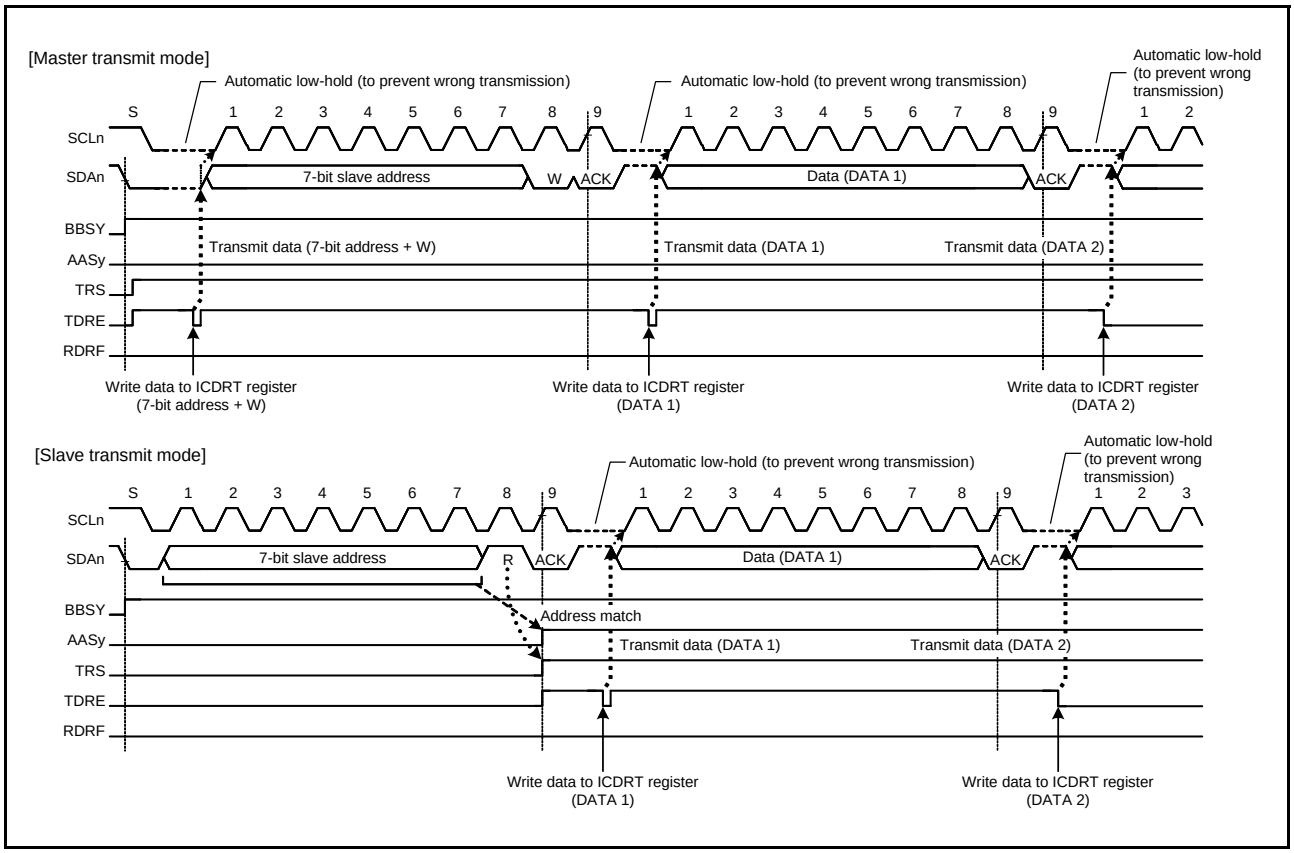


Figure 42.30 Automatic Low-Hold Operation in Transmit Mode

42.8.2 NACK Reception Transfer Suspension Function

The RIIC has a function to suspend transfer operation when NACK is received in transmit mode (ICCR2.TRS bit is 1). This function is enabled when the ICFER.NACKE bit is set to 1 (transfer suspension enabled). If the next transmit data has already been written (ICSR2.TDRE flag is 0) when NACK is received, next data transmission at the falling edge of the ninth SCL clock cycle is automatically suspended. This prevents the SDAn line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (ICSR2.NACKF flag is 1), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to set the NACKF flag to 0. In master transmit mode, after setting the NACKF flag to 0, issue a restart condition, or issue a stop condition and then issue a start condition again.

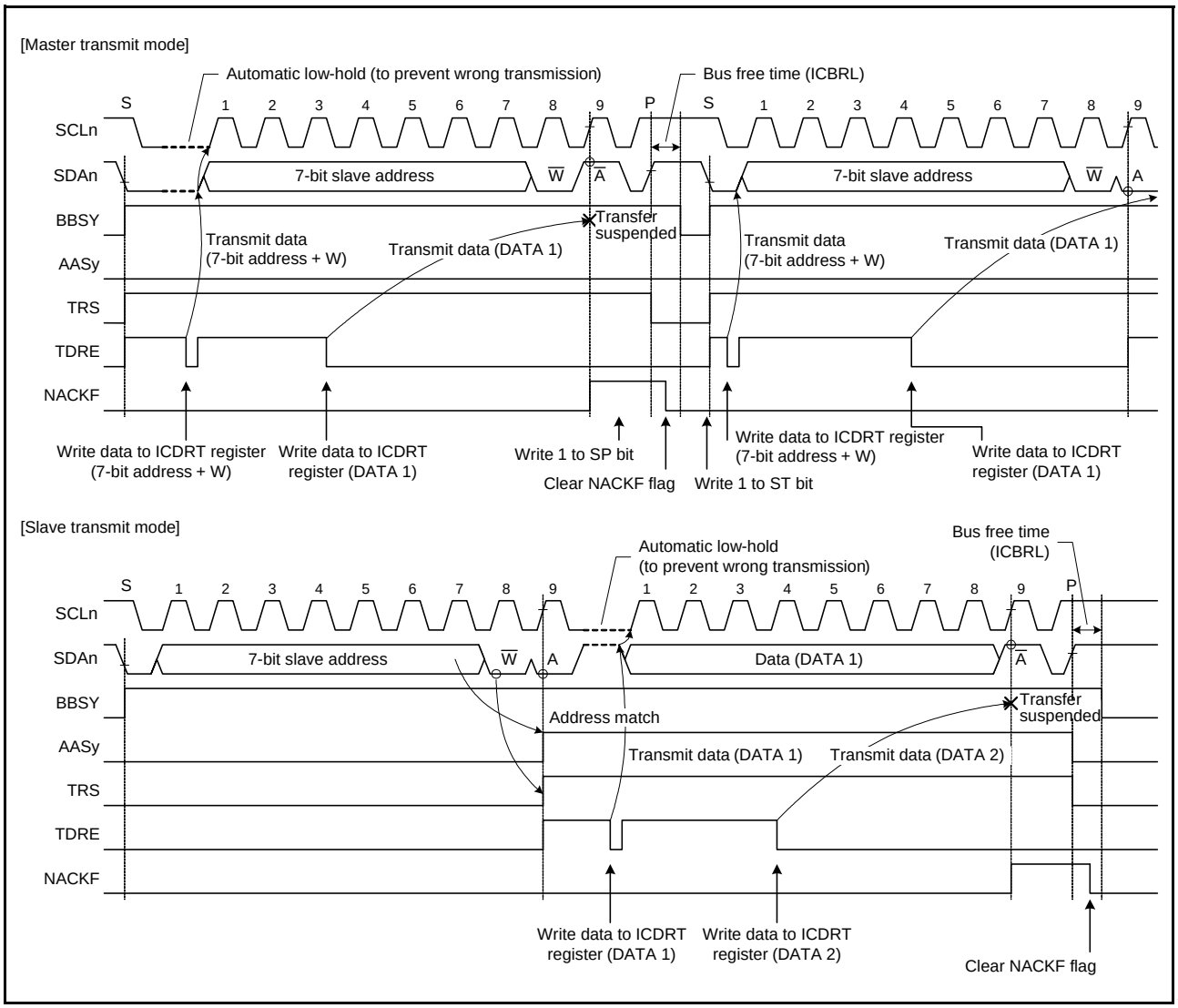


Figure 42.31 Suspension of Data Transfer When NACK is Received (NACKE = 1)

42.8.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDRR) read is delayed for a period of one transfer byte or more with receive data full (ICSR2.RDRF flag is 1) in receive mode (ICCR2.TRS bit is 0), the RIIC holds the SCLn line low automatically immediately before the next data is received to prevent failure to receive data.

This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the RIIC's own slave address is designated after a

stop condition is issued. This function does not disturb other communication because the RIIC does not hold the SCLn line low when a mismatch with its own slave address occurs after a stop condition is issued.
Sections in which the SCLn line is held low can be selected with a combination of the WAIT and RDRFS bits in the ICMR3 register.

(1) 1-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the ICMR3.WAIT bit is set to 1, the RIIC performs 1-byte receive operation using the WAIT bit function. Furthermore, when the ICMR3.RDRFS bit is 0, the RIIC automatically sends the ICMR3.ACKBT bit value for the acknowledge bit in the period from the falling edge of the eighth SCL clock cycle to the falling edge of the ninth SCL clock cycle, and automatically holds the SCLn line low at the falling edge of the ninth SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from the ICDRR register, which enables bytewise receive operation.

The WAIT bit function is enabled for receive bytes after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

(2) 1-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the RDRFS Bit

When the ICMR3.RDRFS bit is set to 1, the RIIC performs 1-byte receive operation using the RDRFS bit function. When the RDRFS bit is set to 1, the ICSR2.RDRF flag (receive data full) is set to 1 at the rising edge of the eighth SCL clock cycle, and the SCLn line is automatically held low at the falling edge of the eighth SCL clock cycle. This low-hold is released by writing a value to the ICMR3.ACKBT bit, but cannot be released by reading data from the ICDRR register, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.

The RDRFS bit function is enabled for receive bytes after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

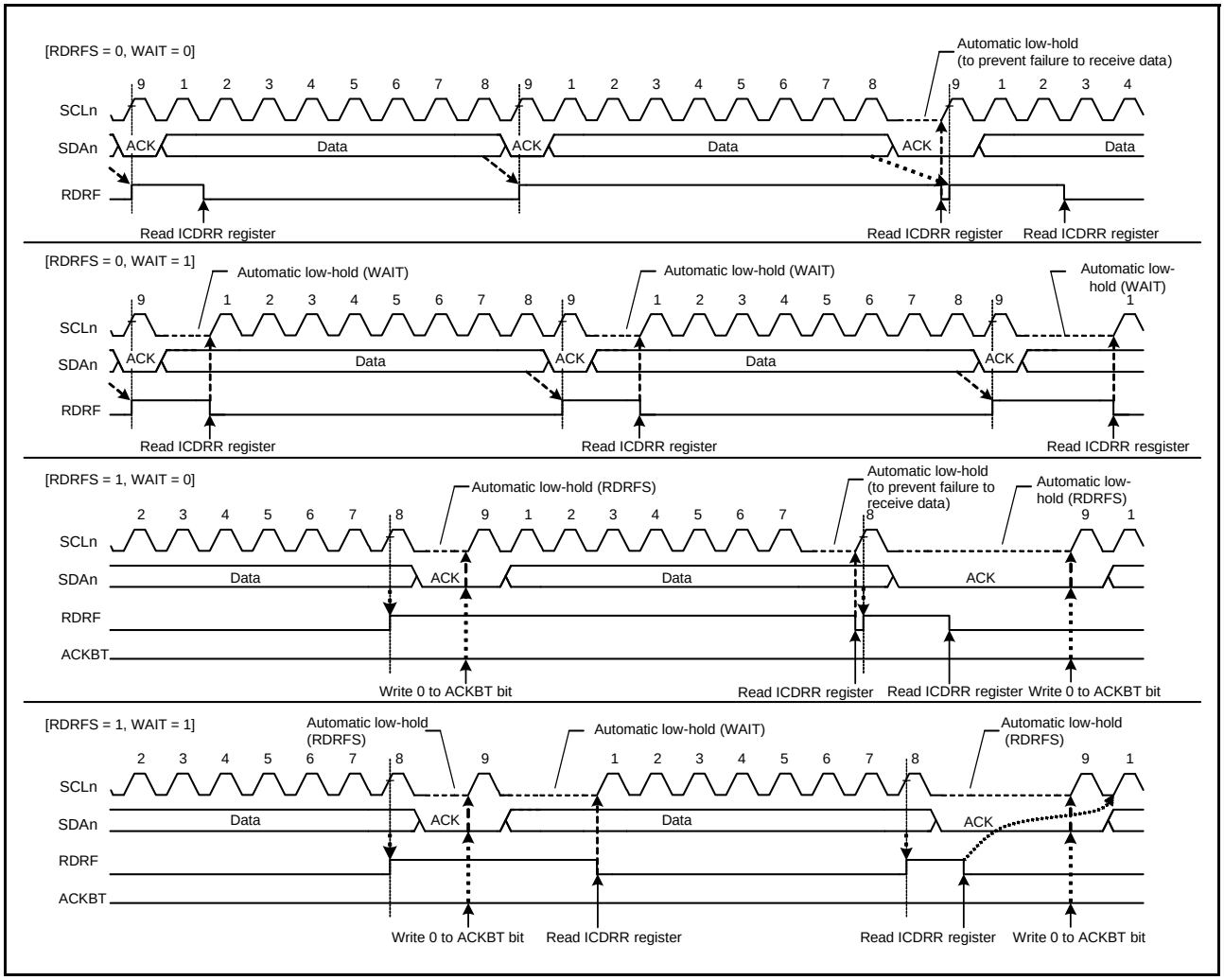


Figure 42.32 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

42.9 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I2C-bus specification, the RIIC has functions to prevent double-issue of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

42.9.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDAn line low to issue a start condition. However, if the SDAn line has already been driven low by another master device issuing a start condition, the RIIC causes arbitration to be lost, so priority is given to transfer by the other master device. Similarly, if the ICCR2.ST bit is set to 1 while the ICCR2.BBSY flag is 1 (bus busy state), arbitration is lost, so priority is given to transfer by the other master device. No start condition is issued in this case.
When a start condition is issued successfully, if the data for transmission including the address bits (i.e. the internal SDA output level) and the level on the SDAn line do not match (the high output as the internal SDA output; i.e. the SDAn pin is in the high-impedance state) and the low level is detected on the SDAn line, the RIIC loses in arbitration.
After a loss in arbitration of mastership, the RIIC immediately enters slave receive mode. If a slave address (including the general call address) matches its own address at this time, the RIIC continues in slave operation.
A loss in arbitration of mastership is detected when the following conditions are met while the ICFER.MALE bit is 1 (master arbitration-lost detection enabled).

Conditions for master arbitration-lost

- Non-matching of the internal level for output on SDA and the level on the SDAn line after a start condition was issued by setting the ICCR2.ST bit to 1 while the ICCR2.BBSY flag was set to 0 (erroneous issuing of a start condition)
- Setting of the ICCR2.ST bit to 1 (start condition double-issue error) while the BBSY flag is set to 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDAn line in master transmit mode (bits MST and TRS in the ICCR2 register = 11b)

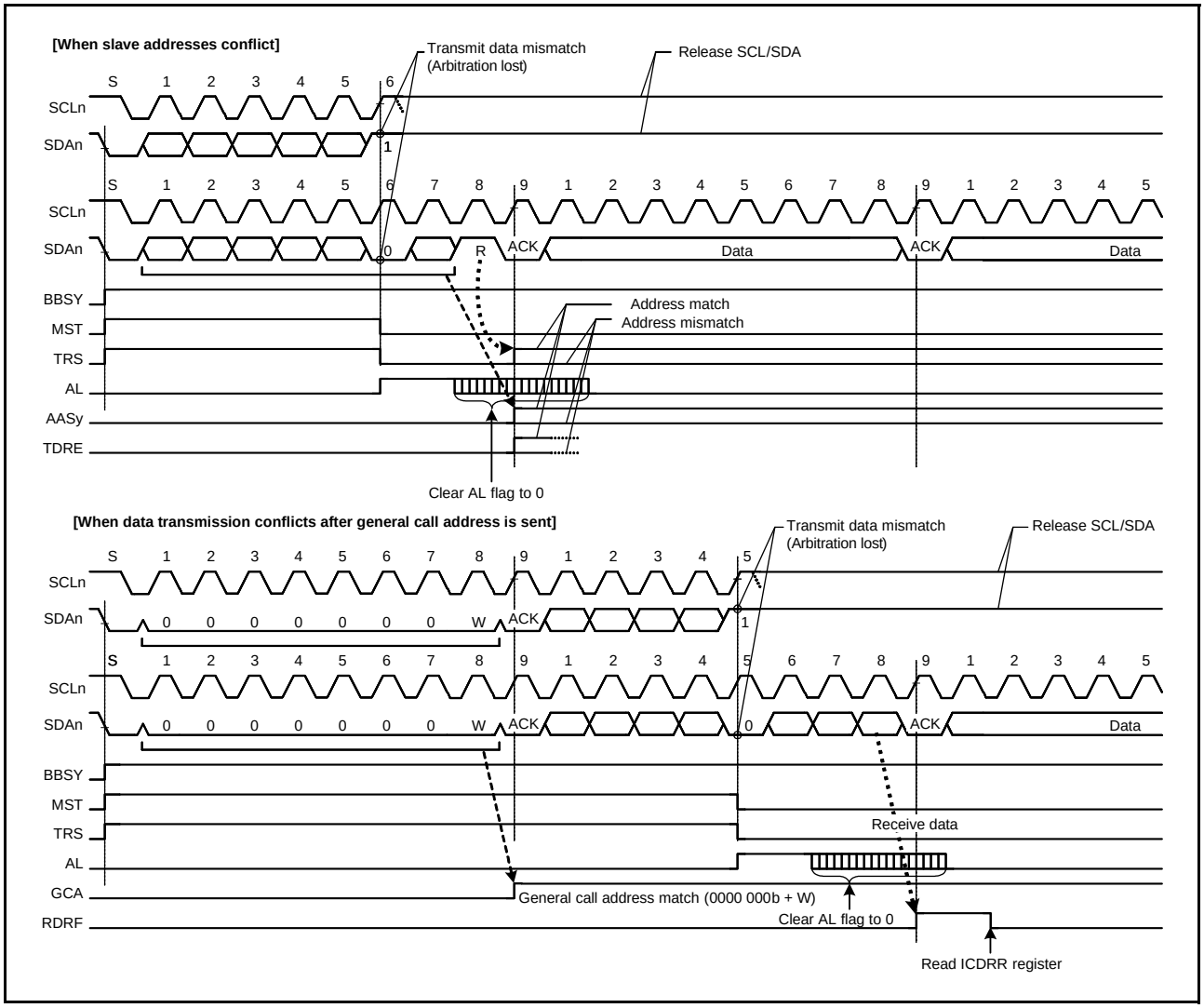


Figure 42.33 Examples of Master Arbitration-Lost Detection (MALE = 1)

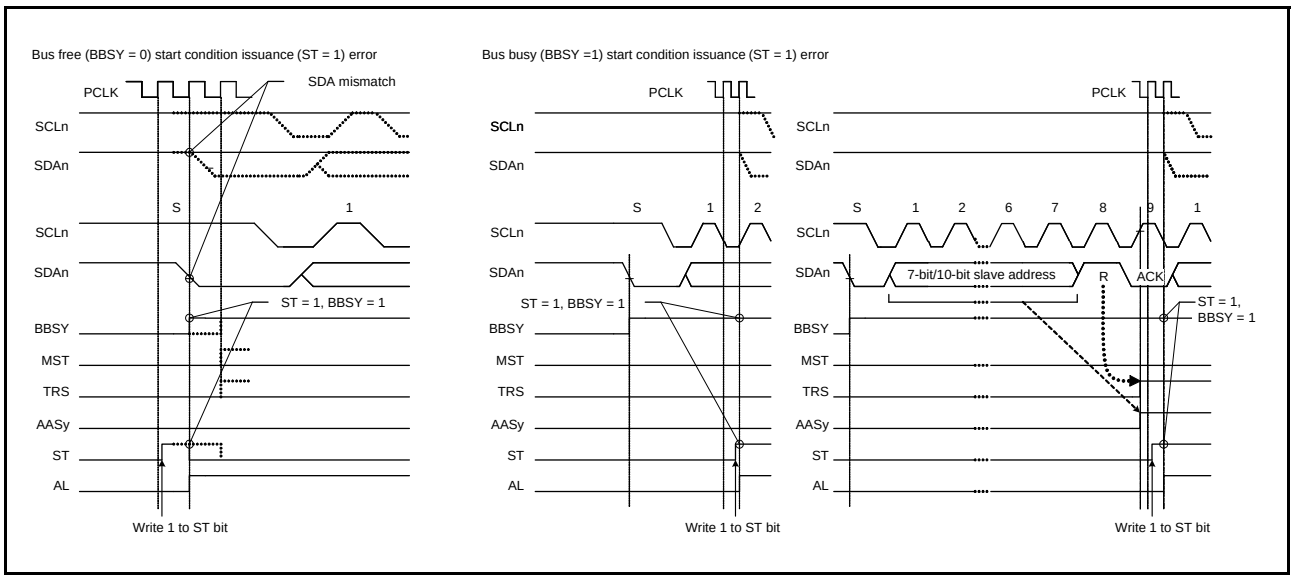


Figure 42.34 Arbitration-Lost When a Start Condition is Issued (MALE = 1)

42.9.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)

The RIIC has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the SDAn line (the high output as the internal SDA output; i.e. the SDAn pin is in the high-impedance state) and the low level is detected on the SDAn line during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device. Figure 42.35 shows an example of arbitration-lost detection during transmission of NACK.

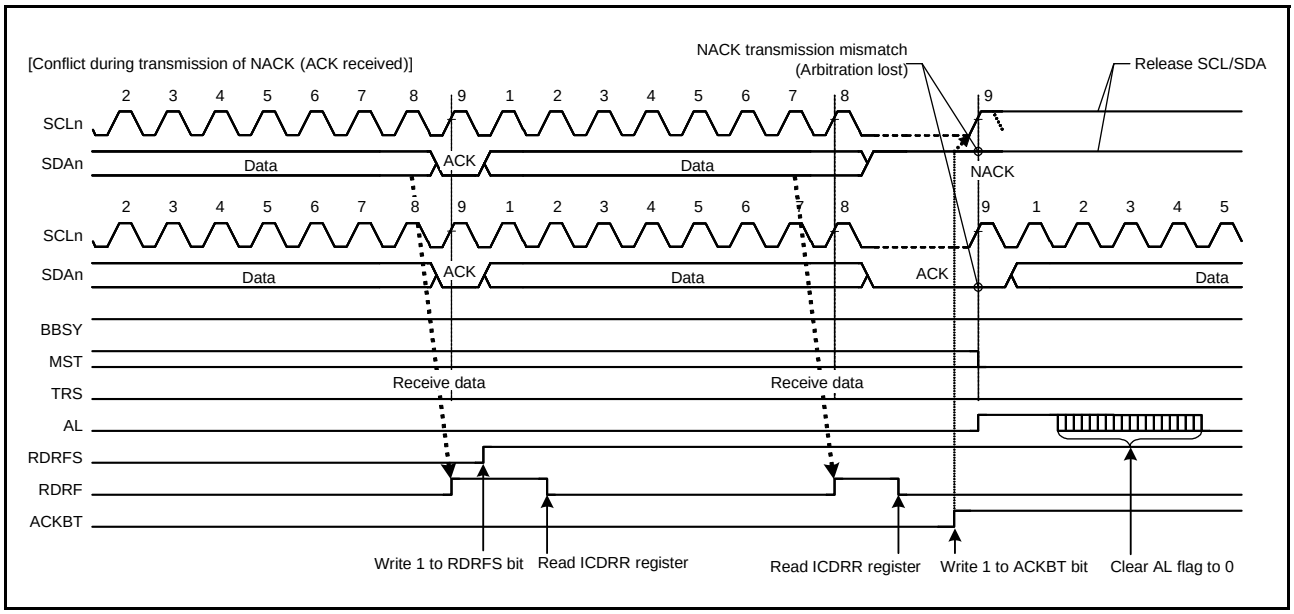


Figure 42.35 Example of Arbitration-Lost Detection during Transmission of NACK (NALE = 1)

The following explains arbitration-lost detection using an example where two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives 2 bytes of data from the slave device, and master B receives 4 bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in both master A and master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Here, master A sends NACK when it has received 2 final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary 4 bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect ACK transmitted by master B and issues a stop condition. Therefore, the issuance of the stop condition conflicts with the SCL clock output of master B, which disturbs communication.

When the RIIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost.

If arbitration is lost during transmission of NACK, the RIIC immediately cancels the slave match condition and enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing (such as FFh transmission processing) necessary if the UDID (Unique Device Identifier) of assign address does not match in the Get UDID (general) processing after the Assign address command.

The RIIC detects arbitration-lost during transmission of NACK when the following condition is met with the ICFER.NALE bit set to 1 (arbitration-lost detection during NACK transmission enabled).

Condition for arbitration-lost during NACK transmission

- When the internal SDA output level does not match the SDAn line (ACK is received) during transmission of NACK (ICMR3.ACKBT bit = 1)

42.9.3 Slave Arbitration-Lost Detection (SALE Bit)

The RIIC has a function to cause arbitration to be lost if the data for transmission (i.e. the internal SDA output level) and the level on the SDAn line do not match (the high output as the internal SDA output; i.e. the SDAn pin is in the high-impedance state and the low level is detected on the SDAn line in slave transmit mode). This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.
When it loses slave arbitration, the RIIC is immediately released from the slave-matched state and enters slave receive mode. This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminate subsequent redundant processing (processing for the transmission of FFh).
The RIIC detects slave arbitration-lost when the following condition is met with the ICFER.SALE bit set to 1 (slave arbitration-lost detection enabled).

Condition for slave arbitration-lost

- When transmit data excluding acknowledge (internal SDA output level) does not match the SDAn line in slave transmit mode (bits MST and TRS in the ICCR2 register are 01b)

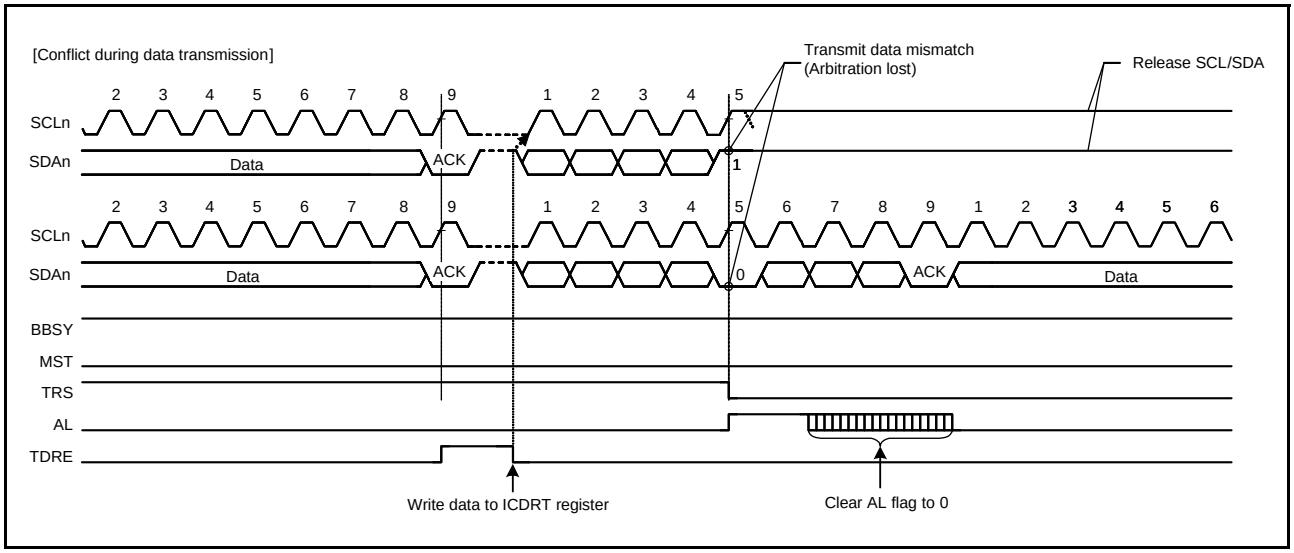


Figure 42.36 Example of Slave Arbitration-Lost Detection (SALE = 1)

42.10 Start Condition/Restart Condition/Stop Condition Issuing Function

42.10.1 Issuing a Start Condition

The RIIC issues a start condition when the ICCR2.ST bit is set to 1.
When the ST bit is set to 1, a start condition issuance request is made and the RIIC issues a start condition when the ICCR2.BBSY flag is 0 (bus free state). When a start condition is issued normally, the RIIC automatically shifts to the master transmit mode.
A start condition is issued in the following sequence.

Start condition issuance

(1) Drive the SDAn line low (high level to low level).
(2) Ensure the time set in the ICBRH register and the start condition hold time.
(3) Drive the SCLn line low (high level to low level).
(4) Detect low level of the SCLn line and ensure the low-level period of SCLn line set in the ICBRL register.

42.10.2 Issuing a Restart Condition

The RIIC issues a restart condition when the ICCR2.RS bit is set to 1.
When the RS bit is set to 1, a restart condition issuance request is made and the RIIC issues a restart condition when the ICCR2.BBSY flag is 1 (bus busy state) and the ICCR2.MST bit is 1 (master mode).
A restart condition is issued in the following sequence.

Restart condition issuance

(1) Release the SDAn line.
(2) Ensure the low-level period of SCLn line set in the ICBRL register.
(3) Release the SCLn line (low level to high level).
(4) Detect a high level of the SCLn line and ensure the time set in the ICBRL register and the restart condition setup time.
(5) Drive the SDAn line low (high level to low level).
(6) Ensure the time set in the ICBRH register and the restart condition hold time.
(7) Drive the SCLn line low (high level to low level).
(8) Detect a low level of the SCLn line and ensure the low-level period of SCLn line set in the ICBRL register.

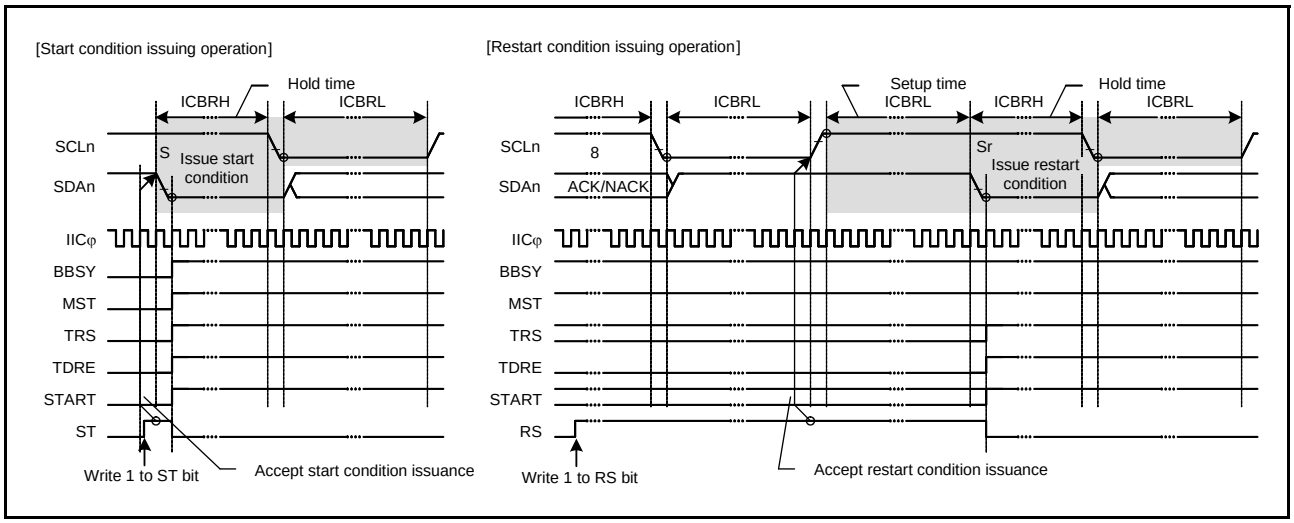


Figure 42.37 Start Condition/Restart Condition Issue Timing (ST and RS Bits)

42.10.3 Issuing a Stop Condition

The RIIC issues a stop condition when the ICCR2.SP bit is set to 1.
When the SP bit is set to 1, a stop condition issuance request is made and the RIIC issues a stop condition when the ICCR2.BBSY flag is 1 (bus busy state) and the ICCR2.MST bit is 1 (master mode).
A stop condition is issued in the following sequence.

Stop condition issuance

- Drive the SDAn line low (high level to low level).
- Ensure the low-level period of SCLn line set in the ICBRL register.
- Release the SCLn line (low level to high level).
- Detect a high level of the SCLn line and ensure the time set in the ICBRH register and the stop condition setup time.
- Release the SDAn line (low level to high level).
- Ensure the time set in the ICBRL register and the bus free time.
- Set the BBSY flag to 0 (to release the bus mastership).

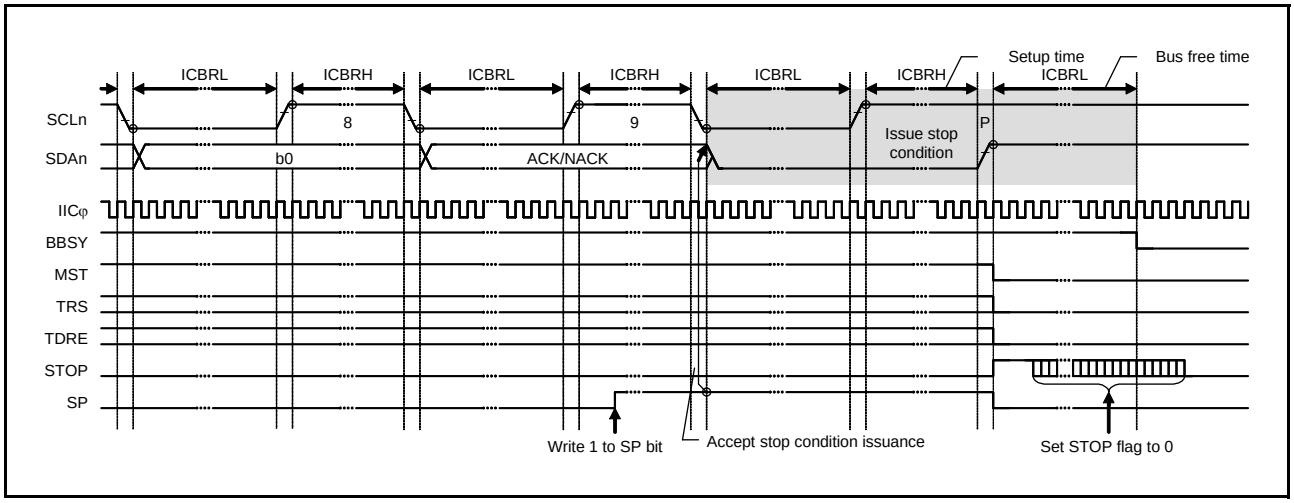


Figure 42.38 Stop Condition Issue Timing (SP Bit)

42.11 Bus Hanging

If the clock signals from the master and slave devices go out of synchronization due to noise or other factors, the I2C-bus might hang with a fixed level on the SCLn line and/or SDAn line.
As measures against the bus hanging, the RIIC has a timeout function to detect hanging by monitoring the SCLn line, a function for the output of an extra SCL clock cycle to release the bus from a hung state due to clock signals being out of synchronization, the RIIC reset function, and internal reset function.
By checking bits SCLO, SDAO, SCLI, and SDAI in the ICCR1 register, it is possible to see whether the RIIC or its partner in communications is placing the low level on the SCLn or SDAn lines.

42.11.1 Timeout Function

The RIIC includes a timeout function for detecting when the SCLn line has been stuck longer than the predetermined time. The RIIC can detect an abnormal bus state by monitoring that the SCLn line is stuck low or high for a predetermined time.
The timeout function monitors the SCLn line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the SCLn line changes (rising or falling), but continues to count unless the SCLn line changes. If the internal counter overflows due to no SCLn line change, the RIIC can detect the timeout and report the bus hung state.
This timeout function is enabled when the ICFER.TMOE bit is 1. It detects a hung state that the SCLn line is stuck low or high during the following conditions:

- The bus is busy (ICCR2.BBSY flag is 1) in master mode (ICCR2.MST bit is 1).
- The RIIC's own slave address is detected (ICSR1 register is not 00h) and the bus is busy (ICCR2.BBSY flag is 1) in slave mode (ICCR2.MST bit is 0).
- The bus is free (ICCR2.BBSY flag is 0) while generation of a start condition is requested (ICCR2.ST bit is 1).

The internal counter of the timeout function works using the internal reference clock (IICφ) set by the ICMR1.CKS[2:0] bits as a count source. It functions as a 16-bit counter when long mode is selected (ICMR2.TMOS bit is 0) or a 14-bit counter when short mode is selected (TMOS bit is 1).
The SCLn line level (low/high or both levels) during which this counter is activated can be selected by the setting of bits TMOH and TMOL in the ICMR2 register. If both bits TMOL and TMOH are set to 0, the internal counter does not work.

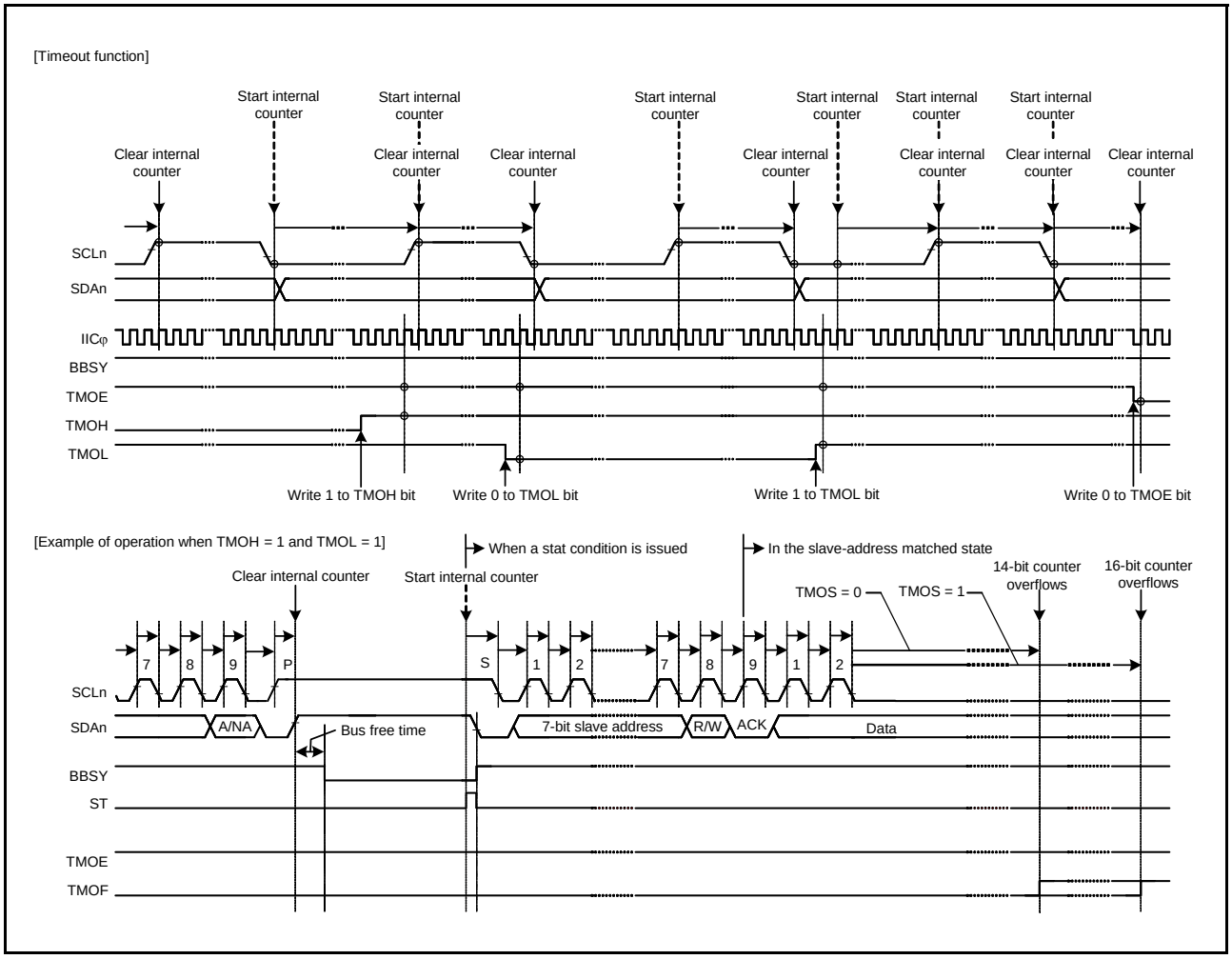


Figure 42.39 Timeout Function

42.11.2 Extra SCL Clock Cycle Output Function

In master mode, the RIIC module has a facility for the output of extra SCL clock cycles to release the SDAn line of the slave device from being held at the low level due to the master being out of synchronization with the slave device. This function is mainly used in master mode to release the SDAn line of the slave device from the state of being fixed to the low level by including extra cycles of SCL output from the RIIC with single cycles of the SCL clock as the unit if the RIIC cannot issue a stop condition because the slave device is holding the SDAn line at the low level. Do not use this facility in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.

When the ICCR1.CLO bit is set to 1 in master mode, a single cycle of the SCL clock at the frequency corresponding to the transfer rate settings (settings of the ICMR1.CKS[2:0] bits, and of registers ICBRH and ICBRL) is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically set to 0. Therefore, further extra clock cycles can be output consecutively by writing 1 to the CLO bit after confirming the CLO bit to be 0.

When the RIIC module is in master mode and the slave device is holding the SDAn line at the low level because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a stop condition is not possible. The facility for output of an extra cycle of the SCL clock can be used to output extra cycles of SCL one by one to make the slave device release the SDAn line from being held at the low level, thus recovering the bus from an unusable state. Release of the SDAn line by the slave device can be monitored by reading the ICCR1.SDAI bit. After confirming release of the SDAn line by the slave device, complete communications by reissuing the stop condition.

Use this facility with the ICFER.MALE bit (master arbitration-lost detection disabled) set to 0. If the MALE bit is set to 1 (master arbitration-lost detection enabled), arbitration is lost when the value of the ICCR1.SDAO bit does not match the state of the SDAn line, so take care on this point.

Output conditions for using the ICCR1.CLO bit

- When the bus is free (ICCR2.BBSY flag is 0) or in master mode (ICCR2.MST bit is 1 and ICCR2.BBSY flag is 1)
- When the communication device does not hold the SCLn line low

Figure 42.40 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

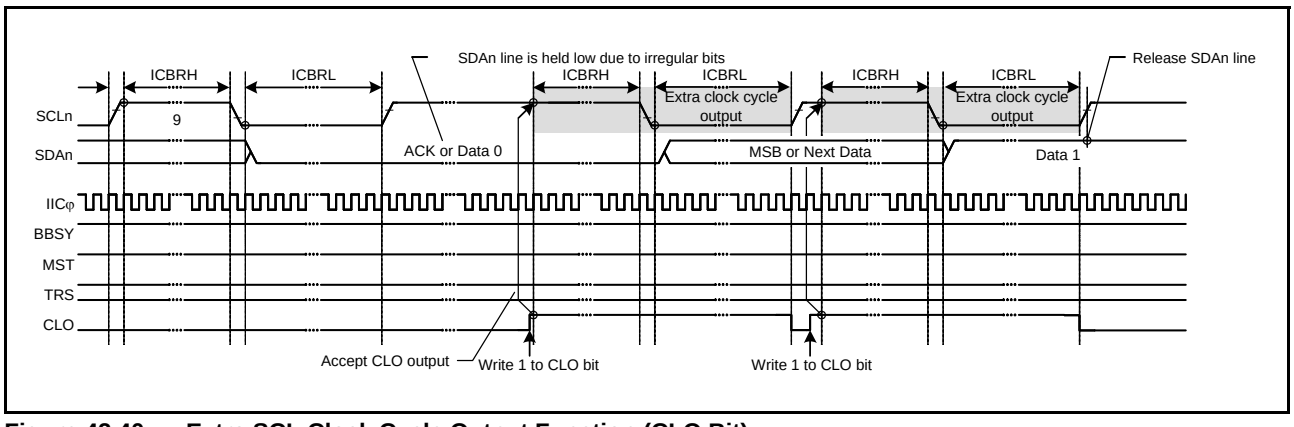


Figure 42.40 Extra SCL Clock Cycle Output Function (CLO Bit)

42.11.3 RIIC Reset and Internal Reset

The RIIC module incorporates a function for resetting itself. There are two types of reset. One is referred to as an RIIC reset; this initializes all registers including the ICCR2.BBSY flag. The other is referred to as an internal reset; this releases the RIIC from the slave-address matched state and initializes the internal counter while retaining other settings. After issuing a reset, be sure to set the ICCR1.IICRST bit to 0.

Both types of reset are effective for release from bus-hung states because both restore the output state of the SCLn and SDAn pins to the high-impedance state.

Issuing a reset during slave operation may lead to a loss of synchronization between the master device clock and the slave device clock, so avoided this where possible. Note that monitoring of the bus state, such as for the presence of a start condition, is not possible during an RIIC reset (bits ICE and IICRST in the ICCR1 register are 01b).

For a detailed description of the RIIC and internal resets, refer to section 42.14, Resets and Register and Function States When Issuing Each Condition.

42.12 SMBus Operation

The RIIC is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the ICMR3.SMBS bit to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus specification, set the ICMR1.CKS[2:0] bits, the ICBRH register, and the ICBRL register. In addition, determine the values of the ICMR2.DLCS bit and the ICMR2.SDDL[2:0] bits to meet the data hold time specification of 300 ns or more. If the RIIC is used only as a slave device, the transfer rate setting is not necessary, whereas the ICBRL register needs to be set to a value longer than the data setup time (250 ns).
For the SMBus device default address (1100 001b), use one of the slave address registers L0 to L2 (registers SARL0, SARL1, and SARL2), and set the corresponding SARUy.FS bit (7-bit/10-bit address format select) (y = 0 to 2) to 0 (7-bit address format).
When transmitting the UDID (Unique Device Identifier), set the ICFER.SALE bit to 1 to enable the slave arbitration-lost detection function.

42.12.1 SMBus Timeout Measurement

(1) Measuring timeout of slave device

The following period (timeout interval: $T_{LOW:SEXT}$) must be measured for slave devices in SMBus communication.

- From start condition to stop condition

To measure timeout for slave devices, measure the period from start condition detection to stop condition detection with the MTU or TMR timer using a start condition detection interrupt (STI) and stop condition detection interrupt (SPI) of the RIIC. The measured timeout period must be within the total clock low-level period [slave device] $T_{LOW:SEXT}$: 25 ms (max.) of the SMBus specification.
If the time measured with the MTU or TMR exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (min.) of the SMBus specification, the slave device must release the bus by writing 1 to the ICCR1.IICRST bit to issue an internal reset of the RIIC. When an internal reset is issued, the RIIC stops driving the bus for the SCLn pin and SDAn pin and make the SCLn/SDAn pin outputs high-impedance, which releases the bus.

(2) Measuring timeout of master device

The following periods (timeout interval: $T_{LOW:MEXT}$) must be measured for master devices in SMBus communication.

- From start condition to acknowledge bit
- Between acknowledge bits
- From acknowledge bit to stop condition

To measure timeout for master devices, measure these periods with the MTU or TMR timer using a start condition detection interrupt (STI), stop condition detection interrupt (SPI), and transmit end interrupt (TEI) or receive data full interrupt (RXI) of the RIIC. The measured timeout period must be within the total clock low-level extended period (master device) $T_{LOW:MEXT}$: 10 ms (max.) of the SMBus specification, and the total of all $T_{LOW:MEXT}$ from start condition to stop condition must be within $T_{LOW:SEXT}$: 25 ms (max.).
For the ACK receive timing (rising edge of the ninth SCL clock cycle), monitor the ICSR2.TEND flag in master transmit mode (master transmitter) and the ICSR2.RDRF flag in master receive mode (master receiver). For this reason, perform bytewise transmit operation in master transmit mode, and hold the ICMR3.RDRFS bit 0 until the byte just before reception of the final byte in master receive mode. While the RDRFS bit is 0, the RDRF flag is set to 1 at the rising edge of the ninth SCL clock cycle.
If the period measured with the MTU or TMR exceeds the total clock low-level extended period (master device) $T_{LOW:MEXT}$: 10 ms (max.) of the SMBus specification or the total of measured periods exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (min.) of the SMBus specification, the master device must stop the transaction by issuing a stop condition. In master transmit mode, immediately stop the transmit operation (writing data to the ICDRT register).

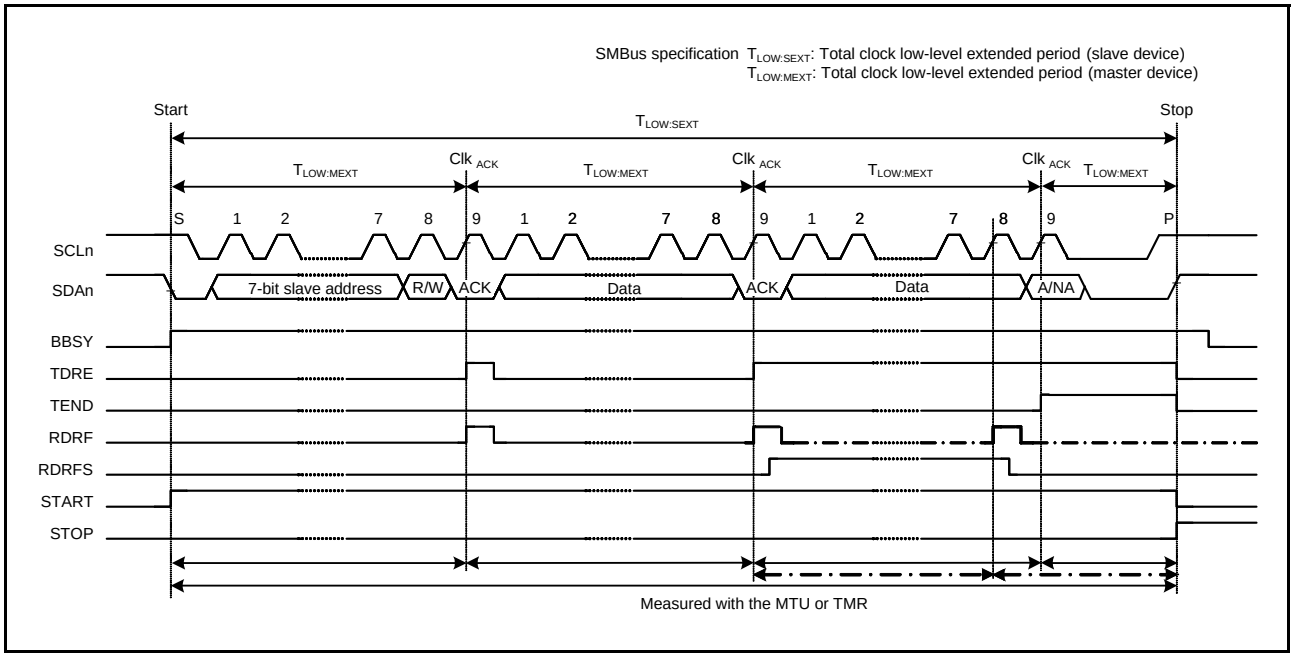


Figure 42.41 SMBus Timeout Measurement

42.12.2 Packet Error Code (PEC)

This MCU incorporates a CRC calculator. The CRC calculator enables transmission of a packet error code (PEC) or checking the received data of the SMBus in data communication of the RIIC. For the CRC generating polynomials of the CRC calculator, refer to section 46, CRC Calculator (CRC).

The PEC data in master transmit mode can be generated by writing all transmit data to the CRC data input register (CRCDIR) in the CRC calculator.

The PEC data in master receive mode can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC data output register (CRCDOR) with the received PEC data.

To send ACK or NACK according to the match or mismatch result when the final byte is received as a result of the PEC code check, set the ICMR3.RDRFS bit to 1 before the rising edge of the eighth SCL clock cycle during reception of the final byte, and hold the SCLn line low at the falling edge of the eighth clock cycle.

42.12.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of its own slave address or to request its own slave address from the SMBus host.

For a product of this MCU to operate as an SMBus host (or ARP master), the host address (0001 000b) sent from the slave device must be detected as a slave address, so the RIIC has a function for detecting the host address. To detect the host address as a slave address, set the ICMR3.SMBS bit and the ICSER.HOAE bit to 1. Operation after the host address has been detected is the same as normal slave operation.

42.13 Interrupt Sources

The RIIC issues four types of interrupt request: transfer error or event generation (arbitration-lost, NACK detection, timeout detection, start condition detection, and stop condition detection), receive data full, transmit data empty, and transmit end.

Table 42.6 lists details of the several interrupt requests. The receive data full and transmit data empty are both capable of activating data transfer by the DTC or DMAC.

Table 42.6 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation	Priority	Interrupt Condition
EEI	Transfer error/ event generation	AL	Not possible	Not possible	High	AL = 1 • ALIE = 1
		NACKF				NACKF = 1 • NAKIE = 1
		TMOF				TMOF = 1 • TMOIE = 1
		START				START = 1 • STIE = 1
		STOP				STOP = 1 • SPIE = 1
RXI*2	Receive data full	RDRF	Possible	Possible		RDRF = 1 • RIE = 1
TXI*1	Transmit data empty	TDRE	Possible	Possible		TDRE = 1 • TIE = 1
TEI*3	Transmit end	TEND	Not possible	Not possible	Low	TEND = 1 • TEIE = 1

Note: There is a delay time between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or an interrupt request has been masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt handling. Returning from interrupt handling without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.

Note 1. Because TXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICSR2.TDRE flag (a condition for TXI) is automatically set to 0 when data for transmission are written to the ICDRT register or a stop condition is detected (ICSR2.STOP flag is 1).

Note 2. Because RXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICSR2.RDRF flag (a condition for RXI) is automatically set to 0 when data are read from the ICDRR register.

Note 3. When using the TEI interrupt, clear the ICSR2.TEND flag in the TEI interrupt handling.
Note that the ICSR2.TEND flag is automatically set to 0 when data for transmission are written to the ICDRT register or a stop condition is detected (ICSR2.STOP flag is 1).

Clear the each flag or mask the interrupt request during interrupt handling.

42.13.1 Buffer Operation for TXI and RXI Interrupts

If the conditions for generating a TXI and RXI interrupt are satisfied while the corresponding IR flag is 1, the interrupt request is not output for the ICU but retained internally (the capacity for internal retention is one request per source). An interrupt request that was being retained internally is output to the ICU when the value of the ICU.IRn.IR flag becomes 0. Internally retained interrupt requests are automatically cleared under normal conditions of usage. Internally retained interrupt requests can also be cleared by writing 0 to the corresponding interrupt enable bit in the ICIER register.

42.14 Resets and Register and Function States When Issuing Each Condition

The RIIC can be reset by MCU reset, RIIC reset, and internal reset functions. Table 42.7 lists the register and function states when issuing each reset or condition.

Table 42.7 Register and Function States When Issuing Each Reset or Condition

		MCU Reset	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start Condition/ Restart Condition Detection	Stop Condition Detection
ICCR1	ICE, IICRST	To be reset	Retained	Retained	Retained	Retained
	SCLO, SDAO		To be reset	To be reset		
	Others			Retained		
ICCR2	BBSY	To be reset	To be reset	Retained	Retained	Retained
	ST			To be reset	To be reset	Retained
	Others					To be reset
ICMR1	BC[2:0]	To be reset	To be reset	To be reset	To be reset	Retained
	Others			Retained	Retained	
ICMR2		To be reset	To be reset	Retained	Retained	Retained
ICMR3		To be reset	To be reset	Retained	Retained	Retained
ICFER		To be reset	To be reset	Retained	Retained	Retained
ICSER		To be reset	To be reset	Retained	Retained	Retained
ICIER		To be reset	To be reset	Retained	Retained	Retained
ICSR1		To be reset	To be reset	To be reset	Retained	To be reset
ICSR2	TDRE, TEND	To be reset	To be reset	To be reset	Retained	To be reset
	START				Retained	
	STOP				Retained	Retained
	Others				Retained	Retained
SARL0, SARL1, SARL2, SARU0, SARU1, SARU2		To be reset	To be reset	Retained	Retained	Retained
ICBRH, ICBRL		To be reset	To be reset	Retained	Retained	Retained
ICDRT		To be reset	To be reset	Retained	Retained	Retained
ICDRR		To be reset	To be reset	Retained	Retained	Retained
ICDRS		To be reset	To be reset	To be reset	Retained	Retained
Timeout function		To be reset	To be reset	Operation	Operation	Operation
Bus free time measurement		To be reset	To be reset	Operation	Operation	Operation

42.15 Event Link Function (Output)

The RIIC0 handles event output for the event link controller (ELC) corresponding to the following sources.

- Communication error/ communication event
- Receive data full
- Transmit data empty
- Transmit end

42.15.1 Interrupt Handling and Event Linking

The RIIC module produces four kinds of interrupt: transfer error (arbitration-lost detection, detection of NACK, detection of timeout, or detection of a stop condition) event, receive data full, transmit data empty, and transmit end interrupts detection of a start condition. Each of these has an enable bit to control enabling and disabling of the interrupt signal. An interrupt request signal is output for the CPU when an interrupt source condition is satisfied while the setting of the corresponding enable bit is enabled.
The corresponding event signals are sent to other modules via the ELC when the interrupt source conditions are satisfied, regardless of the settings of the interrupt enable bits.
For details on interrupt sources, see Table 42.6.

42.16 Usage Notes

42.16.1 Setting Module Stop Function

Module stop state can be entered or released using module stop control register B (MSTPCRB) or module stop control register C (MSTPCRC). The initial setting is for operation of the RIIC to be stopped. RIIC register access is enabled by releasing the module stop state.
For details on module stop control registers B and C, refer to section 11, Low Power Consumption.

42.16.2 Notes on Starting Transfer

If the IR flag corresponding to the RIIC interrupt is 1 when transfer is started (ICCR1.ICE bit is 1), follow the procedure below to clear interrupts before enabling operations. Starting transfer with the IR flag set to 1 while the ICCR1.ICE bit is 1 leads to an interrupt request being internally retained after transfer starts, and this can lead to unanticipated behavior of the IR flag.

1. Confirm that the ICCR1.ICE bit is 0.
2. Set the relevant interrupt enable bits (ICIER.TIE, etc.) on the peripheral function side to 0.
3. Read the relevant interrupt enable bits (ICIER.TIE, etc.) on the peripheral function side and confirm that its value is 0.
4. Set the IR flag to 0.

43. CAN Module (CAN)

43.1 Overview

This MCU implements three channels of the CAN (Controller Area Network) module that complies with the ISO 11898-1 Standards. The CAN module transmits and receives both formats of messages, namely the standard identifier (11 bits) (identifier is hereafter referred to as ID) and extended ID (29 bits).

Table 43.1 lists the specifications of the CAN module, and Figure 43.1 shows a block diagram of the CAN module (i = 0 to 2).

Connect the CAN bus transceiver externally.

Table 43.1 Specifications of CAN Module

Item	Description
Protocol	• ISO 11898-1 compliant (standard and extended frames)
Bit rate	• Programmable bit rate up to 1 Mbps (fCAN ≥ 8 MHz) fCAN: CAN clock source
Message box	• 32 mailboxes: Two selectable mailbox modes Normal mailbox mode: 32 mailboxes can be configured for either transmission or reception. FIFO mailbox mode: 24 mailboxes can be configured for either transmission or reception. Of the other mailboxes, four FIFO stages can be configured for transmission and four FIFO stages for reception.
Reception	• Data frame and remote frame can be received. • Selectable receiving ID format (only standard ID, only extended ID, or both IDs) • Programmable one-shot reception function • Selectable from overwrite mode (message overwritten) and overrun mode (message discarded) • The reception complete interrupt can be individually enabled or disabled for each mailbox.
Acceptance filter	• Eight acceptance masks (one mask for every four mailboxes) • The mask can be individually enabled or disabled for each mailbox.
Transmission	• Data frame and remote frame can be transmitted. • Selectable transmitting ID format (only standard ID, only extended ID, or both IDs) • Programmable one-shot transmission function • Selectable from ID priority mode and mailbox number priority mode • Transmission request can be aborted (the completion of abort can be confirmed with a flag) • The transmission complete interrupt can be individually enabled or disabled for each mailbox.
Mode transition for bus-off recovery	• Mode transition for the recovery from the bus-off state can be selected: ISO 11898-1 Standards compliant Automatic entry to CAN halt mode at bus-off entry Automatic entry to CAN halt mode at bus-off end Entry to CAN halt mode by a program Transition into error-active state by a program
Error status monitoring	• CAN bus errors (stuff error, form error, ACK error, CRC error, bit error, and ACK delimiter error) can be monitored. • Transition to error states can be detected (error-warning, error-passive, bus-off entry, and bus-off recovery). • The error counters can be read.
Time stamp function	• Time stamp function using a 16-bit counter • The reference clock can be selected from 1-, 2-, 4- and 8-bit time periods.
Interrupt function	• Five types of interrupt sources (reception complete, transmission complete, receive FIFO, transmit FIFO, and error interrupts)
CAN sleep mode	• Current consumption can be reduced by stopping the CAN clock.
Software support unit	• Three software support units: Acceptance filter support Mailbox search support (receive mailbox search, transmit mailbox search, and message lost search) Channel search support
CAN clock source	Peripheral module clock (PCLKB) or CANMCLK
Test mode	• Three test modes available for user evaluation Listen-only mode Self-test mode 0 (external loopback) Self-test mode 1 (internal loopback)
Power consumption reducing function	Module-stop state can be set.

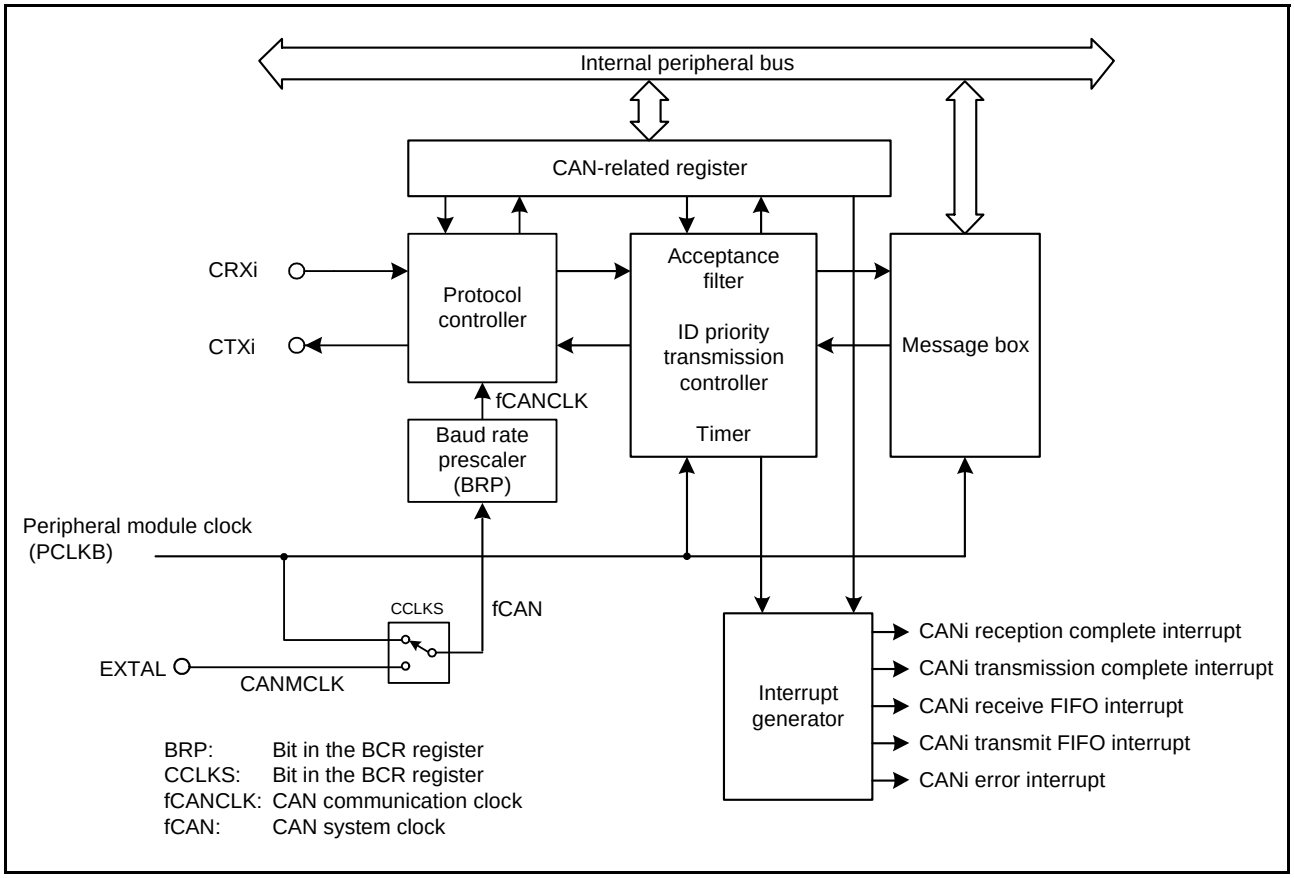


Figure 43.1 Block Diagram of CAN Module (i = 0 to 2)

- CRXi and CTXi (i = 0 to 2)
  CAN input and output pins
- Protocol controller
  Handles CAN protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, and error handling, etc.
- Message box
  Consists of 32 mailboxes which can be configured as either transmit or receive mailboxes. Each mailbox has an individual ID, data length code, a data field (8 bytes), and a time stamp.
- Acceptance filter
  Performs filtering of received messages. MKR0 to MKR7 are used for the filtering process.
- Timer
  Used for the time stamp function. The timer value when a message is stored into the mailbox is written as the time stamp value.
- Interrupt generator:
  Generates the following five types of interrupts:
  CANi reception complete interrupt
  CANi transmission complete interrupt
  CANi receive FIFO interrupt
  CANi transmit FIFO interrupt
  CANi error interrupt

Table 43.2 lists the CAN module pins.
The CAN functions should be selected for the pins multiplexed with other signals. For details, see section 22, I/O Ports.

Table 43.2 Pin Configuration

Pin Name	I/O	Function
CRX0	Input	Pin for receiving data
CTX0	Output	Pin for transmitting data
CRX1	Input	Pin for receiving data
CTX1	Output	Pin for transmitting data
CRX2	Input	Pin for receiving data
CTX2	Output	Pin for transmitting data

43.2 Register Descriptions

43.2.1 Control Register (CTLR)

Address(es): CAN0.CTLR 0009 0840h, CAN1.CTLR 0009 1840h, CAN2.CTLR 0009 2840h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RBOC	BOM[1:0]		SLPM	CANM[1:0]		TSPS[1:0]		TSRC	TPM	MLM	IDFM[1:0]		MBM
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MBM	CAN Mailbox Mode Select *1	0: Normal mailbox mode 1: FIFO mailbox mode	R/W
b2, b1	IDFM[1:0]	ID Format Mode Select*1	b2 b1 0 0: Standard ID mode All mailboxes (including FIFO mailboxes) handle only standard Ids. 0 1: Extended ID mode All mailboxes (including FIFO mailboxes) handle only extended IDs. 1 0: Mixed ID mode All mailboxes (including FIFO mailboxes) handle both standard IDs and extended IDs. Standard IDs or extended IDs are specified by using the IDE bit in the corresponding mailbox in normal mailbox mode. In FIFO mailbox mode, the IDE bit in the corresponding mailbox is used for mailboxes [0] to [23], the IDE bits in FIDCR0 and FIDCR1 are used for the receive FIFO, and the IDE bit in mailbox [24] is used for the transmit FIFO. 1 1: Do not use this combination	R/W
b3	MLM	Message Lost Mode Select*2	0: Overwrite mode 1: Overrun mode	R/W
b4	TPM	Transmission Priority Mode Select*2	0: ID priority transmit mode 1: Mailbox number priority transmit mode	R/W
b5	TSRC	Time Stamp Counter Reset Command*4	0: Nothing occurred 1: Reset*3	R/W
b7, b6	TSPS[1:0]	Time Stamp Prescaler Select*1	b7 b6 0 0: Every bit time 0 1: Every 2-bit time 1 0: Every 4-bit time 1 1: Every 8-bit time	R/W
b9, b8	CANM[1:0]	CAN Operating Mode Select*5	b9 b8 0 0: CAN operation mode 0 1: CAN reset mode 1 0: CAN halt mode 1 1: CAN reset mode (forcible transition)	R/W
b10	SLPM	CAN Sleep Mode*5,*6	0: Other than CAN sleep mode 1: CAN sleep mode	R/W
b12, b11	BOM[1:0]	Bus-Off Recovery Mode *1	b12 b11 0 0: Normal mode (ISO 11898-1 compliant) 0 1: Entry to CAN halt mode automatically at bus-off entry 1 0: Entry to CAN halt mode automatically at bus-off end 1 1: Entry to CAN halt mode (during bus-off recovery period) by a program request	R/W
b13	RBOC	Forcible Return From Bus-Off*2	0: Nothing occurred 1: Forcible return from bus-off*3	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Write to the BOM[1:0], TSPS[1:0], TPM, MLM, IDFM[1:0], and MBM bits in CAN reset mode.
Note 2. Set the RBOC bit to 1 in the bus-off state.
Note 3. This bit is automatically set back to 0 after being set to 1. It should be read as 0.
Note 4. Set the TSRC bit to 1 in CAN operation mode.
Note 5. When the CANM[1:0] and SLPM bits are changed, check STR to ensure that the mode has been switched. Do not change the CANM[1:0] bits or SLPM bit until the mode has been switched.

Note 6. Write to the SLPM bit in CAN reset mode or CAN halt mode. When changing the SLPM bit, write 0 or 1 to only the SLPM bit.

MBM Bit (CAN Mailbox Mode Select)

When the MBM bit is 0 (normal mailbox mode), mailboxes [0] to [31] are configured as transmit or receive mailboxes.
When the MBM bit is 1 (FIFO mailbox mode), mailboxes [0] to [23] are configured as transmit or receive mailboxes. Mailboxes [24] to [27] are configured as a transmit FIFO and mailboxes [28] to [31] as a receive FIFO.
Transmit data is written into mailbox [24] (mailbox [24] is a window mailbox for the transmit FIFO). Receive data is read from mailbox [28] (mailbox [28] is a window mailbox for the receive FIFO).
Table 43.3 lists the mailbox configuration.

IDFM[1:0] Bits (ID Format Mode Select)

The IDFM[1:0] bits specify the ID format.

MLM Bit (Message Lost Mode Select)

The MLM bit specifies the operation when a new message is captured in the unread mailbox. Overwrite mode or overrun mode can be selected. All mailboxes (including the receive FIFO) are set to either overwrite mode or overrun mode.
When the MLM bit is 0, all mailboxes are set to overwrite mode and the new message is overwriting the old message.
When the MLM bit is 1, all mailboxes are set to overrun mode and the new message is discarded.

TPM Bit (Transmission Priority Mode Select)

The TPM bit specifies the priority of modes when transmitting messages.
ID priority transmit mode or mailbox number transmit mode can be selected. All mailboxes are set for either ID priority transmission or mailbox number priority transmission.
When the TPM bit is 0, ID priority transmit mode is selected and transmission priority complies with the CAN bus arbitration rule, as defined in the ISO 11898-1 Standards. In ID priority transmit mode, mailboxes [0] to [31] (in normal mailbox mode), and mailboxes [0] to [23] (in FIFO mailbox mode), and the transmit FIFO are compared for the IDs of mailboxes configured for transmission. If two or more mailbox IDs are the same, the mailbox with the smaller number has higher priority.
Only the next message to be transmitted from the transmit FIFO is included in the transmission arbitration. If a transmit FIFO message is being transmitted, the next pending message within the transmit FIFO is included in the transmission arbitration.
When the TPM bit is 1, mailbox number transmit mode is selected and the transmit mailbox with the smallest mailbox number has the highest priority. In FIFO mailbox mode, the transmit FIFO has lower priority than normal mailboxes (mailboxes [0] to [23]).

TSRC Bit (Time Stamp Counter Reset Command)

The TSRC bit is used to reset the time stamp counter. When the TSRC bit is set to 1, TSR is set to 0000h. This bit is automatically set to 0.

TSPS[1:0] Bits (Time Stamp Prescaler Select)

The TSPS[1:0] bits select the prescaler for the time stamp. The reference clock for the time stamp can be selected to be either 1-, 2-, 4- or 8-bit time periods.

CANM[1:0] Bits (CAN Operating Mode Select)

The CANM[1:0] bits select one of the following modes for the CAN module: CAN operation mode, CAN reset mode, or CAN halt mode. CAN sleep mode is set by the SLPM bit. For details, refer to section 43.3, Operating Mode.
When the CAN module enters CAN halt mode according to the setting of the BOM[1:0] bits, the CANM[1:0] bits are automatically set to 10b.

SLPM Bit (CAN Sleep Mode)

When the SLPM bit is set to 1, the CAN module enters CAN sleep mode. When the SLPM bit is set to 0, the CAN module exits CAN sleep mode. For details, refer to section 43.3, Operating Mode.

BOM[1:0] Bits (Bus-Off Recovery Mode)

The BOM[1:0] bits are used to select bus-off recovery mode for the CAN module.
When the BOM[1:0] bits are 00b, the recovery from bus-off is compliant with the ISO 11898-1 Standards, i.e. the CAN module reenters CAN communication (error-active state) after detecting 11 consecutive recessive bits 128 times. A bus-off recovery interrupt request is generated when recovering from bus-off.
When the BOM[1:0] bits are 01b and the CAN module reaches the bus-off state, the CANM[1:0] bits in CTLR are set to 10b (CAN halt mode) to enter CAN halt mode. No bus-off recovery interrupt request is generated when recovering from bus-off, and TECR and RECR are set to 00h.
When the BOM[1:0] bits are 10b, the CANM[1:0] bits are set to 10b as soon as the CAN module reaches the bus-off state. The CAN module enters CAN halt mode after the recovery from the bus-off state, i.e. after detecting 11 consecutive recessive bits 128 times. A bus-off recovery interrupt request is generated when recovering from bus-off, and TECR and RECR are set to 00h.
When the BOM[1:0] bits are 11b, the CAN module enters CAN halt mode by setting the CANM[1:0] bits to 10b while the CAN module is still in the bus-off state. No bus-off recovery interrupt request is generated when recovering from bus-off and TECR and RECR are set to 00h. However, if the CAN module recovers from bus-off after detecting 11 consecutive recessive bits 128 times before the CANM[1:0] bits are set to 10b, a bus-off recovery interrupt request is generated.
If the CPU requests an entry to CAN reset mode at the same time as the CAN module attempts to enter CAN halt mode (at bus-off entry when the BOM[1:0] bits are 01b, or at bus-off end when the BOM[1:0] bits are 10b), then the CPU request to enter CAN reset mode has higher priority.

RBOC Bit (Forcible Return From Bus-Off)

When the RBOC bit is set to 1 (force return from bus-off) in the bus-off state, the CAN module forcibly returns from the bus-off state. This bit is automatically set to 0. The error state changes from bus-off to error-active. When the RBOC bit is set to 1, RECR and TECR are set to 00h and the BOST bit in STR is set to 0 (the CAN module is not in bus-off state). The other registers remain unchanged even when the RBOC bit is set to 1. No bus-off recovery interrupt request is generated by this recovery from the bus-off state. Use the RBOC bit only when the BOM[1:0] bits are 00b (normal mode).

Table 43.3 Mailbox Configuration

Mailbox	MBM Bit = 0 (Normal Mailbox Mode)	MBM Bit = 1 (FIFO Mailbox Mode)
Mailboxes [0] to [23]	Normal mailbox	Normal mailbox
Mailboxes [24] to [27]		Transmit FIFO
Mailboxes [28] to [31]		Receive FIFO

Points 1 to 5 below should be considered when the CTLR.MBM bit is set to 1.
Note 1. Transmit FIFO is controlled by TFCR. MCTLj of mailboxes [24] to [27] is disabled.
MCTL24 to MCTL27 cannot be used by the transmit FIFO.
Note 2. Receive FIFO is controlled by RFCR. MCTLj of mailboxes [28] to [31] is disabled.
MCTL28 to MCTL31 cannot be used by the receive FIFO.
Note 3. Refer to MIER about the FIFO interrupts.
Note 4. The corresponding bits in MKIVLR for mailboxes [24] to [31] are disabled. Set 0 to these bits.
Note 5. Transmit/receive FIFOs can be used for both data frames and remote frames.

43.2.2 Bit Configuration Register (BCR)

Address(es): CAN0.BCR 0009 0844h, CAN1.BCR 0009 1844h, CAN2.BCR 0009 2844h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSEG1[3:0]				—	—	BRP[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	SJW[1:0]		—	TSEG2[2:0]			—	—	—	—	—	—	—	CCLKS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CCLKS	CAN Clock Source Selection	0: PCLKB (generated by the PLL clock) 1: CANMCLK (generated by the main clock)	R/W
b7 to b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	TSEG2[2:0]	Time Segment 2 Control	b10 b8 0 0 0: Setting prohibited 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13, b12	SJW[1:0]	Resynchronization Jump Width Control	b13 b12 0 0: 1 Tq 0 1: 2 Tq 1 0: 3 Tq 1 1: 4 Tq	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25 to b16	BRP[9:0]	Prescaler Division Ratio Select*1	These bits set the frequency of the CAN communication clock (fCANCLK).	R/W
b26	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b31 to b28	TSEG1[3:0]	Time Segment 1 Control	b31 b28 0 0 0 0: Setting prohibited 0 0 0 1: Setting prohibited 0 0 1 0: Setting prohibited 0 0 1 1: 4 Tq 0 1 0 0: 5 Tq 0 1 0 1: 6 Tq 0 1 1 0: 7 Tq 0 1 1 1: 8 Tq 1 0 0 0: 9 Tq 1 0 0 1: 10 Tq 1 0 1 0: 11 Tq 1 0 1 1: 12 Tq 1 1 0 0: 13 Tq 1 1 0 1: 14 Tq 1 1 1 0: 15 Tq 1 1 1 1: 16 Tq	R/W

Tq: Time Quantum

Note 1. Do not select the value less than 1 while the SCKCR3.CKSEL[2:0] bits are 010b (selecting the main clock oscillator).

For bit timing setting, refer to section 43.4, CAN Communication Speed Setting.
Set BCR before entering CAN halt mode or CAN operation mode from CAN reset mode. After the setting is made once, this register can be written to in CAN reset mode or CAN halt mode.
BCR consists of 24 bits. A 32-bit read/write access should be performed carefully not to rewrite bits b0 to b7.

CCLKS Bit (CAN Clock Source Selection)

When the CCLKS bit is 0, the peripheral module clock (PCLKB) produced by the PLL frequency synthesizer is used as the CAN clock source (fCAN).
When the CCLKS bit is 1, CANMCLK produced externally by the EXTAL pins is used as the CAN clock source (fCAN).

TSEG2[2:0] Bits (Time Segment 2 Control)

The TSEG2[2:0] bits are used to specify the length of the phase buffer segment 2 (PHASE_SEG2) with a Tq value. A value from 2 to 8 Tq can be set. Set a value smaller than that of the TSEG1[3:0] bits.

SJW[1:0] Bits (Resynchronization Jump Width Control)

The SJW[1:0] bits are used to specify the resynchronization jump width with a Tq value. A value from 1 to 4 Tq can be set. Set a value smaller than or equal to that of the TSEG2[2:0] bits.

BRP[9:0] Bits (Prescaler Division Ratio Select)

The BRP[9:0] bits are used to set the frequency of the CAN communication clock (fCANCLK). The fCANCLK cycle is 1 Tq. If the setting is P (0 to 1023), the baud rate prescaler divides fCAN by P + 1.

TSEG1[3:0] Bits (Time Segment 1 Control)

The TSEG1[3:0] bits are used to specify the total length of the propagation time segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1) with a time quantum (Tq) value.
A value from 4 to 16 Tq can be set.

43.2.3 Mask Register k (MKRk) (k = 0 to 7)

Address(es): CAN0.MKR0 0009 0400h, CAN0.MKR1 0009 0404h, CAN0.MKR2 0009 0408h, CAN0.MKR3 0009 040Ch, CAN0.MKR4 0009 0410h, CAN0.MKR5 0009 0414h, CAN0.MKR6 0009 0418h, CAN0.MKR7 0009 041Ch, CAN1.MKR0 0009 1400h, CAN1.MKR1 0009 1404h, CAN1.MKR2 0009 1408h, CAN1.MKR3 0009 140Ch, CAN1.MKR4 0009 1410h, CAN1.MKR5 0009 1414h, CAN1.MKR6 0009 1418h, CAN1.MKR7 0009 141Ch, CAN2.MKR0 0009 2400h, CAN2.MKR1 0009 2404h, CAN2.MKR2 0009 2408h, CAN2.MKR3 0009 240Ch, CAN2.MKR4 0009 2410h, CAN2.MKR5 0009 2414h, CAN2.MKR6 0009 2418h, CAN2.MKR7 0009 241Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	SID[10:0]											EID[17:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EID[17:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b17 to b0	EID[17:0]	Extended ID	0: Corresponding EID[17:0] bit is not compared 1: Corresponding EID[17:0] bit is compared	R/W
b28 to b18	SID[10:0]	Standard ID	0: Corresponding SID[10:0] bit is not compared 1: Corresponding SID[10:0] bit is compared	R/W
b31 to b29	—	Reserved	The read value is undefined. The write value should be 0.	R/W

For the mask function in FIFO mailbox mode, refer to section 43.6, Acceptance Filtering and Masking Functions.
Write to MKR0 to MKR7 in CAN reset mode or CAN halt mode.

EID[17:0] Bits (Extended ID)

The EID[17:0] bits are the filter mask bits for the CAN extended ID bits.
These bits are used to receive extended ID messages.
When the EID[17:0] bit is set to 0, the received ID is not compared with the mailbox ID for the corresponding EID[17:0] bit.
When the EID[17:0] bit is set to 1, the received ID is compared with the mailbox ID for the corresponding EID[17:0] bit.

SID[10:0] Bits (Standard ID)

The SID[10:0] bits are the filter mask bits corresponding to the CAN standard ID bits.
These bits are used to receive both standard ID and extended ID messages.
When the SID[10:0] bit is set to 0, the received ID is not compared with the mailbox ID for the corresponding SID[10:0] bit.
When the SID[10:0] bit is set to 1, the received ID is compared with the mailbox ID for the corresponding SID[10:0] bit.

43.2.4 FIFO Received ID Compare Registers 0 and 1 (FIDCR0 and FIDCR1)

Address(es): CAN0.FIDCR0 0009 0420h, CAN0.FIDCR1 0009 0424h, CAN1.FIDCR0 0009 1420h, CAN1.FIDCR1 0009 1424h, CAN2.FIDCR0 0009 2420h, CAN2.FIDCR1 0009 2424h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	—	SID[10:0]											EID[17:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EID[17:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b17 to b0	EID[17:0]	Extended ID	0: Corresponding EID[17:0] bits are 0 1: Corresponding EID[17:0] bits are 1	R/W
b28 to b18	SID[10:0]	Standard ID	0: Corresponding SID[10:0] bits are 0 1: Corresponding SID[10:0] bits are 1	R/W
b29	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R/W
b31	IDE	ID Extension*1	0: Standard ID 1: Extended ID	R/W

Note 1. When the IDFM[1:0] bits are not 10b, the IDE bit should be written with 0 and read as 0.

FIDCR0 and FIDCR1 are enabled when the MBM bit in CTLR is set to 1 (FIFO mailbox mode). Bits EID[17:0], SID[10:0], RTR, and IDE in MB28 to MB31 are disabled.
For the usage of FIDCR0 and FIDCR1, refer to section 43.6, Acceptance Filtering and Masking Functions.
Write to FIDCR0 and FIDCR1 in CAN reset mode or CAN halt mode.

EID[17:0] Bits (Extended ID)

The EID[17:0] bits set the extended ID of data frames and remote frames.
These bits are used to receive extended ID messages.

SID[10:0] Bits (Standard ID)

The SID[10:0] bits set the standard ID of data frames and remote frames.
These bits are used to receive both standard ID and extended ID messages.

RTR Bit (Remote Transmission Request)

The RTR bit sets the specified frame format of data frames or remote frames.

- When both RTR bits in FIDCR0 and FIDCR1 are set to 0, only data frames can be received.
- When both RTR bits in FIDCR0 and FIDCR1 are set to 1, only remote frames can be received.
- When the RTR bits in FIDCR0 and FIDCR1 are set to 0 or 1 individually, both data frames and remote frames can be received.

IDE Bit (ID Extension)

The IDE bit sets the ID format to standard ID or extended ID. The IDE bit is enabled when the IDFM[1:0] bits in CTLR are 10b (mixed ID mode).

- When both IDE bits in FIDCR0 and FIDCR1 are set to 0, only standard ID frames can be received.
- When both IDE bits in FIDCR0 and FIDCR1 are set to 1, only extended ID frames can be received.
- When the IDE bits in FIDCR0 and FIDCR1 are set to 0 or 1 individually, both standard ID and extended ID frames can be received.

43.2.5 Mask Invalid Register (MKIVLR)

Address(es): CAN0.MKIVLR 0009 0428h, CAN1.MKIVLR 0009 1428h, CAN2.MKIVLR 0009 2428h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MB31	MB30	MB29	MB28	MB27	MB26	MB25	MB24	MB23	MB22	MB21	MB20	MB19	MB18	MB17	MB16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	MB31 to MB0	Mask Invalid	0: Mask valid 1: Mask invalid	R/W

Each bit in MKIVLR corresponds to a mailbox.
The correspondence between the bits and mailboxes is shown below.
Bit 0 in MKIVLR corresponds to mailbox 0 (MB0) and bit 31 corresponds to mailbox 31 (MB31).*1
When a bit is set to 1, the relevant acceptance mask register becomes invalid for the corresponding mailbox. When a mask invalid bit is set to 1, a message is received by the corresponding mailbox only if the receive message ID matches the mailbox ID exactly.
Write to MKIVLR in CAN reset mode or CAN halt mode.

Note 1. Set bits 31 to 24 to 0 in FIFO mailbox mode.

43.2.6 Mailbox Register j (MBj) (j = 0 to 31)

Table 43.4 lists the CANi mailbox memory mapping, and Table 43.5 lists the CAN data frame configuration.
The value after reset of the CANi mailbox is undefined.
Write to MBj only when the related MCTLj (j = 0 to 31) is 00h and the corresponding mailbox is not processing an abort request.
See Table 43.4 for detailed register addresses.

Table 43.4 CANi Mailbox Memory Mapping

Address			Message Content
CAN0	**CAN1**	**CAN2**	**Memory Mapping**
0009 0200h + 16 × j + 0	0009 1200h + 16 × j + 0	0009 2200h + 16 × j + 0	IDE, RTR, SID10 to SID6
0009 0200h + 16 × j + 1	0009 1200h + 16 × j + 1	0009 2200h + 16 × j + 1	SID5 to SID0, EID17, EID16
0009 0200h + 16 × j + 2	0009 1200h + 16 × j + 2	0009 2200h + 16 × j + 2	EID15 to EID8
0009 0200h + 16 × j + 3	0009 1200h + 16 × j + 3	0009 2200h + 16 × j + 3	EID7 to EID0
0009 0200h + 16 × j + 4	0009 1200h + 16 × j + 4	0009 2200h + 16 × j + 4	—
0009 0200h + 16 × j + 5	0009 1200h + 16 × j + 5	0009 2200h + 16 × j + 5	Data length code (DLC[3:0])
0009 0200h + 16 × j + 6	0009 1200h + 16 × j + 6	0009 2200h + 16 × j + 6	Data byte 0
0009 0200h + 16 × j + 7	0009 1200h + 16 × j + 7	0009 2200h + 16 × j + 7	Data byte 1
0009 0200h + 16 × j + 8	0009 1200h + 16 × j + 8	0009 2200h + 16 × j + 8	Data byte 2
0009 0200h + 16 × j + 9	0009 1200h + 16 × j + 9	0009 2200h + 16 × j + 9	Data byte 3
0009 0200h + 16 × j + 10	0009 1200h + 16 × j + 10	0009 2200h + 16 × j + 10	Data byte 4
0009 0200h + 16 × j + 11	0009 1200h + 16 × j + 11	0009 2200h + 16 × j + 11	Data byte 5
0009 0200h + 16 × j + 12	0009 1200h + 16 × j + 12	0009 2200h + 16 × j + 12	Data byte 6
0009 0200h + 16 × j + 13	0009 1200h + 16 × j + 13	0009 2200h + 16 × j + 13	Data byte 7
0009 0200h + 16 × j + 14	0009 1200h + 16 × j + 14	0009 2200h + 16 × j + 14	Time stamp upper byte
0009 0200h + 16 × j + 15	0009 1200h + 16 × j + 15	0009 2200h + 16 × j + 15	Time stamp lower byte

Note: When accessing to the content of the MBj register, access to an address of a multiple of 4 (the suffix of the address is 0h, 4h, 8h, or Ch) in 32-bit units and to an even address in 16-bit units.

Table 43.5 CAN Data Frame Configuration

SID10 to SID6	SID5 to SID0	EID17 to EID16	EID15 to EID8	EID7 to EID0	DLC3 to DLC1	DATA0	DATA1	…	DATA7

The previous value of each mailbox is retained unless a new message is received.

Address(es): CAN0.MB0 to CAN0.MB31 0009 0200h to 0009 03FFh, CAN1.MB0 to CAN1.MB31 0009 1200h to 0009 13FFh, CAN2.MB0 to CAN2.MB31 0009 2200h to 0009 23FFh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	—	SID[10:0]											EID[17:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EID[17:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b17 to b0	EID[17:0]	Extended ID*1	0: Corresponding EID[17:0] bits are 0 1: Corresponding EID[17:0] bits are 1	R/W
b28 to b18	SID[10:0]	Standard ID	0: Corresponding SID[10:0] bits are 0 1: Corresponding SID[10:0] bits are 1	R/W
b29	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b30	RTR	Remote Frame Request	0: Data frame 1: Remote frame	R/W
b31	IDE	ID Extension*2	0: Standard ID 1: Extended ID	R/W

Note 1. If the mailbox has received a standard ID message, the EID bits in the mailbox are undefined.
Note 2. The IDE bit is enabled when the IDFM[1:0] bits in CTLR are 10b (mixed ID mode). When the IDFM[1:0] bits are not 10b, it should be written with 0 and read as 0.

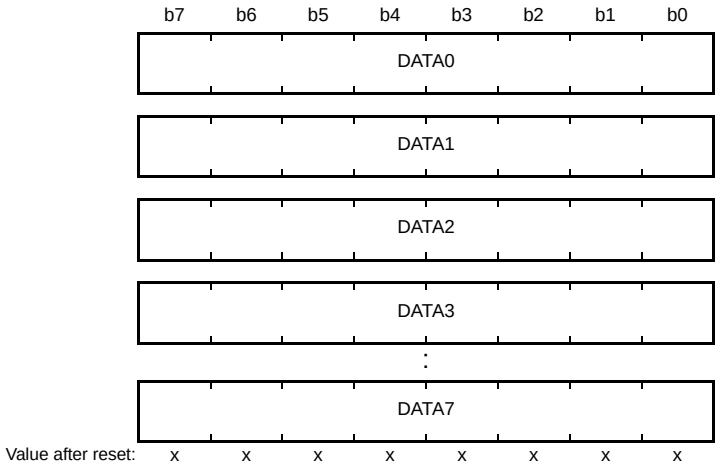
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	DLC[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DLC[3:0]	Data Length Code*1	b3 b0 0 0 0 0: Data length = 0 byte 0 0 0 1: Data length = 1 byte 0 0 1 0: Data length = 2 bytes 0 0 1 1: Data length = 3 bytes 0 1 0 0: Data length = 4 bytes 0 1 0 1: Data length = 5 bytes 0 1 1 0: Data length = 6 bytes 0 1 1 1: Data length = 7 bytes 1 x x x: Data length = 8 bytes	R/W
b15 to b4	—	Reserved	The read value is undefined. The write value should be 0.	R/W

x: Don't care

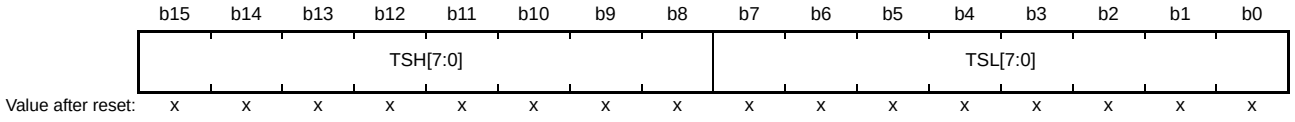
Note 1. If the mailbox has received a message whose data length set by the DLC[3:0] bits is less than 8 bytes, the values of DATA larger than the data length set by the DLC[3:0] bits in the mailbox are undefined.



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DATA0 to DATA7	Data Bytes 0 to 7*1, *2, *3	DATA0 to DATA7 store the transmitted or received CAN message data. Transmission or reception starts from DATA0. The bit order on the CAN bus is MSB first, and transmission or reception starts from bit 7.	R/W

Note 1. If the mailbox has received a message with n bytes less than 8 bytes, the values of DATAn to DATA7 in the mailbox are undefined.
Note 2. If the mailbox has received a remote frame, the previous values of DATA0 to DATA7 in the mailbox are retained.
Note 3. DATA0 to DATA3 or DATA4 to DATA7 cannot be accessed in 32-bit units at one time. The access must be divided into 3 times: DATA0 to DATA1, DATA2 to DATA5, and DATA6 to DATA7, or 4 times: DATA0 to DATA1, DATA2 to DATA3, DATA4 to DATA5, and DATA6 to DATA7.



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TSL[7:0]	Time Stamp Lower Byte	Bits TSH[7:0] and TSL[7:0] store the counter value of the time stamp when received messages are stored in the mailbox.	R/W
b15 to b8	TSH[7:0]	Time Stamp Higher Byte		R/W

EID[17:0] Bits (Extended ID)

The EID[17:0] bits set the extended ID of data frames and remote frames.
These bits are used to transmit or receive extended ID messages.

SID[10:0] Bits (Standard ID)

The SID[10:0] bits set the standard ID of data frames and remote frames.
These bits are used to transmit or receive both standard ID and extended ID messages.

RTR Bit (Remote Frame Request)

The RTR bit sets the frame format of data frames or remote frames.

- Receive mailbox receives only frames with the format specified by the RTR bit.
- Transmit mailbox transmits according to the frame format specified by the RTR bit.
- Receive FIFO mailbox receives the data frame, remote frame, or both frames specified by the RTR bit in FIDCR0 and FIDCR1.
- Transmit FIFO mailbox transmits the data frame or remote frame specified by the RTR bit in the relevant transmit message.

IDE Bit (ID Extension)

The IDE bit sets the ID format of standard IDs or extended IDs. The IDE bit is enabled when the IDFM[1:0] bits in CTLR is 10b (mixed ID mode).

- Receive mailbox receives only the ID format specified by the IDE bit.
- Transmit mailbox transmits with the ID format specified by the IDE bit.
- Receive FIFO mailbox receives messages with the standard ID and extended ID specified by the IDE bit in FIDCR0 and FIDCR1.
- Transmit FIFO mailbox transmits messages with the standard ID or extended ID specified by the IDE bit in the relevant transmit message.

DLC[3:0] Bits (Data Length Code)

The DLC[3:0] bits specify the number of bytes of data to be transmitted in data frames. When a remote frame is used to request data, this field specifies the requested number of bytes of data.
When a data frame is received, the number of bytes received is stored in this field. When a remote frame is received, this field is used to store the number requested by the frame.

43.2.7 Mailbox Interrupt Enable Register (MIER)

Address(es): CAN0.MIER 0009 042Ch, CAN1.MIER 0009 142Ch, CAN2.MIER 0009 242Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MB31	MB30	MB29	MB28	MB27	MB26	MB25	MB24	MB23	MB22	MB21	MB20	MB19	MB18	MB17	MB16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

- Normal mailbox mode

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	MB31 to MB0	Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled Bit 31 corresponds to mailbox 31 (MB31), and bit 0 corresponds to mailbox 0 (MB0).	R/W

- FIFO mailbox mode

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	MB23 to MB0	Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled Bit 23 corresponds to mailbox 23 (MB23), and bit 0 corresponds to mailbox 0 (MB0).	R/W
b24	MB24	Transmit FIFO Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/W
b25	MB25	Transmit FIFO Interrupt Generation Timing Control	0: Every time transmission is completed 1: When the transmit FIFO becomes empty due to completion of transmission	R/W
b27, b26	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b28	MB28	Receive FIFO Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/W
b29	MB29	Receive FIFO Interrupt Generation Timing Control*1	0: Every time reception is completed 1: When the receive FIFO becomes buffer warning by completion of reception	R/W
b31, b30	—	Reserved	The read value is undefined. The write value should be 0.	R/W

Note 1. No interrupt request is generated when the receive FIFO becomes buffer warning from full. “Buffer warning” indicates a state in which the third message is stored in the receive FIFO.

MIER can individually enable interrupts for each mailbox.
In normal mailbox mode (all bits) and in FIFO mailbox mode (bits 24 to 0 in MIER), each bit corresponds to the mailbox with the related number. These bits enable or disable transmission/reception complete interrupts for the corresponding mailboxes.

- Bit 0 in MIER corresponds to mailbox 0 (MB0).
- Bit 31 in MIER corresponds to mailbox 31 (MB31).

In FIFO mailbox mode, bits 29, 28, 25, and 24 of MIER specify whether transmit/receive FIFO interrupts are enabled/disabled and the timing when interrupt requests are generated.
Write to MIER only when the related MCTLj (j = 0 to 31) is 00h and the corresponding mailbox is not processing a transmission or reception abort request. In FIFO mailbox mode, change the bits in MIER for the related FIFO only when the TFE bit in TFCR is 0 and the TFEST flag is 1, and the RFE bit in RFCR is 0 and the RFEST flag in RFCR is 1.

43.2.8 Message Control Register j (MCTLj) (j = 0 to 31)

Address(es): CAN0.MCTL0 to CAN0.MCTL31 0009 0820h to 0009 083Fh, CAN1.MCTL0 to CAN1.MCTL31 0009 1820h to 0009 183Fh, CAN2.MCTL0 to CAN2.MCTL31 0009 2820h to 0009 283Fh

- Transmit mode (when the TRMREQ bit is 1 and the RECREQ bit is 0)

	b7	b6	b5	b4	b3	b2	b1	b0
	TRMREQ	RECREQ	—	ONESHOT	—	TRMABT	TRMACTIVE	SENTDATA
Value after reset:	0	0	0	0	0	0	0	0

- Receive mode (when the TRMREQ bit is 0 and the RECREQ bit is 1)

	b7	b6	b5	b4	b3	b2	b1	b0
	TRMREQ	RECREQ	—	ONESHOT	—	MSGLOST	INVALDATA	NEWDATA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SENTDATA	Transmission Complete Flag *1,*2	0: Transmission is not completed 1: Transmission is completed	R/W
	NEWDATA	Reception Complete Flag *1,*2	0: No data has been received or 0 is written to the NEWDATA flag 1: A new message is being stored or has been stored to the mailbox	R/W
b1	TRMACTIVE	Transmission-in-Progress Status Flag	(Transmit mailbox setting enabled) 0: Transmission is pending or transmission is not requested 1: From acceptance of transmission request to completion of transmission, or error/arbitration-lost	R
	INVALDATA	Reception-in-Progress Status Flag	(Receive mailbox setting enabled) 0: Message valid 1: Message being updated	R
b2	TRMABT	Transmission Abort Complete Flag*1,*2	(Transmit mailbox setting enabled) 0: Transmission has started, transmission abort failed because transmission is completed, or transmission abort is not requested 1: Transmission abort is completed	R/W
	MSGLOST	Message Lost Flag*1,*2	(Receive mailbox setting enabled) 0: Message is not overwritten or overrun 1: Message is overwritten or overrun	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	ONESHOT	One-Shot Enable*3	0: One-shot reception or one-shot transmission disabled 1: One-shot reception or one-shot transmission enabled	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	RECREQ	Receive Mailbox Request *2,*3,*4,*5	0: Not configured for reception 1: Configured for reception	R/W
b7	TRMREQ	Transmit Mailbox Request *2,*4	0: Not configured for transmission 1: Configured for transmission	R/W

Note 1. Write 0 only. Writing 1 has no effect.
Note 2. When writing 0 to bits NEWDATA, SENTDATA, MSGLOST, TRMABT, RECREQ, and TRMREQ by a program, do not use the logic operation instruction (AND). Write 0 to only the specified bit and write 1 to the other bits before using the transfer (MOV) instruction.
Note 3. To enter one-shot receive mode, write 1 to the ONESHOT bit at the same time as setting the RECREQ bit to 1.
To exit one-shot receive mode, write 0 to the ONESHOT bit after writing 0 to the RECREQ bit and confirming that it has been set to 0.
To enter one-shot transmit mode, write 1 to the ONESHOT bit at the same time as setting the TRMREQ bit to 1.
To exit one-shot transmit mode, write 0 to the ONESHOT bit after the message has been transmitted or aborted.
Note 4. Do not set both the RECREQ and TRMREQ bits to 1.
Note 5. When setting the RECREQ bit to 0, set bits MSGLOST, NEWDATA, and RECREQ to 0 simultaneously.

Write to the MCTLj in CAN operation mode or CAN halt mode.
Do not use MCTL24 to MCTL31 in FIFO mailbox mode.

SENTDATA Flag (Transmission Complete Flag)

The SENTDATA flag is set to 1 when data transmission from the corresponding mailbox is completed. The SENTDATA flag is set to 0 by writing 0 by a program.
To set the SENTDATA flag to 0, first set the TRMREQ bit to 0. Bits SENTDATA and TRMREQ cannot be set to 0 simultaneously. To transmit a new message from the corresponding mailbox, set the SENTDATA flag to 0.

NEWDATA Flag (Reception Complete Flag)

The NEWDATA flag is set to 1 when a new message is being stored or has been stored to the mailbox. The timing for setting this bit to 1 is simultaneous with the INVALDATA flag. The NEWDATA flag is set to 0 by writing 0 by a program. The NEWDATA flag cannot be set to 0 by writing 0 by a program while the related INVALDATA flag is 1.

TRMACTIVE Flag (Transmission-in-Progress Status Flag)

The TRMACTIVE flag is set to 1 when the corresponding mailbox of the CAN module begins transmitting a message. The TRMACTIVE flag is set to 0 when the CAN module has lost CAN bus arbitration, a CAN bus error occurs, or data transmission is completed.

INVALDATA Flag (Reception-in-Progress Status Flag)

After the completion of a message reception, the INVALDATA flag is set to 1 while the received message is being updated into the corresponding mailbox. The INVALDATA flag is set to 0 immediately after the message has been stored. If the mailbox is read while the INVALDATA flag is 1, the data is undefined.

TRMABT Flag (Transmission Abort Complete Flag)

The TRMABT flag is set to 1 in the following cases:

- Following a transmission abort request, when the transmission abort is completed before starting transmission.
- Following a transmission abort request, when the CAN module detects CAN bus arbitration-lost or a CAN bus error.
- In one-shot transmission mode (RECREQ bit = 0, TRMREQ bit = 1, and ONESHOT bit = 1), when the CAN module detects CAN bus arbitration-lost or a CAN bus error.

The TRMABT flag is not set to 1 when data transmission is completed. In this case, the SENTDATA flag is set to 1. The TRMABT flag is set to 0 by writing 0 by a program.

MSGLOST Flag (Message Lost Flag)

The MSGLOST flag is set to 1 when the mailbox is overwritten or overrun by a new received message while the NEWDATA flag is 1. The MSGLOST flag is set to 1 at the end of the 6th bit of EOF. The MSGLOST flag is set to 0 by writing 0 by a program.
In both overwrite and overrun modes, the MSGLOST flag cannot be set to 0 by writing 0 by a program during five peripheral module clock (PCLKB) cycles following the sixth bit of EOF.

ONESHOT Bit (One-Shot Enable)

The ONESHOT bit can be used in the following two ways, receive mode and transmit mode.

- One-shot receive mode
  When the ONESHOT bit is set to 1 in receive mode (RECREQ bit = 1 and TRMREQ bit = 0), the mailbox receives a message only one time. (The mailbox does not behave as a receive mailbox after having received a message one time.) The behavior of flags NEWDATA and INVALDATA is the same as in normal receive mode. In one-shot receive mode, the MSGLOST flag is not set to 1. To set the ONESHOT bit to 0, first write 0 to the RECREQ bit and ensure that it has been set to 0.
- One-shot transmit mode
  When the ONESHOT bit is set to 1 in transmit mode (RECREQ bit = 0 and TRMREQ bit = 1), the CAN module transmits a message only one time. (The CAN module does not transmit the message again if a CAN bus error or CAN bus arbitration-lost occurs.) When transmission is completed, the SENTDATA flag is set to 1. If transmission is not completed due to a CAN bus error or CAN bus arbitration-lost, the TRMABT flag is set to 1. Set the ONESHOT bit to 0 after the SENTDATA or TRMABT flag is set to 1.

RECREQ Bit (Receive Mailbox Request)

The RECREQ bit selects receive modes listed in Table 43.10.
When the RECREQ bit is set to 1, the corresponding mailbox is configured for reception of a data frame or a remote frame.
When the RECREQ bit is set to 0, the corresponding mailbox is not configured for reception of a data frame or a remote frame.
Due to hardware protection, the RECREQ bit cannot be set to 0 by writing 0 by a program during the following period.

- Hardware protection is started
  From the acceptance filter processing (the beginning of CRC field)

- Hardware protection is released
  – For the mailbox that is specified to receive the incoming message, after the received data is stored into the mailbox or a CAN bus error occurs (i.e. maximum period of hardware protection is from the beginning of CRC field to the end of the 7th bit of EOF)
  – For the other mailboxes, after the acceptance filter processing
  – If no mailbox is specified to receive the message, after the acceptance filter processing
  When setting the RECREQ bit to 1, do not set the TRMREQ bit to 1. To change the configuration of a mailbox from transmission to reception, first abort the transmission and then set flags SENTDATA and TRMABT to 0 before changing to reception.

TRMREQ Bit (Transmit Mailbox Request)

The TRMREQ bit selects transmit modes listed in Table 43.10.
When the TRMREQ bit is set to 1, the corresponding mailbox is configured for transmission of a data frame or a remote frame.
When the TRMREQ bit is set to 0, the corresponding mailbox is not configured for transmission of a data frame or a remote frame.
If the TRMREQ bit is changed from 1 to 0 to cancel the corresponding transmission request, either the TRMABT or SENTDATA flag is set to 1. When setting the TRMREQ bit to 1, do not set the RECREQ bit to 1. To change the configuration of a mailbox from reception to transmission, first abort the reception and then set flags NEWDATA and MSGLOST to 0 before changing to transmission.

43.2.9 Receive FIFO Control Register (RFCR)

Address(es): CAN0.RFCR 0009 0848h, CAN1.RFCR 0009 1848h, CAN2.RFCR 0009 2848h

	b7	b6	b5	b4	b3	b2	b1	b0
	RFEST	RFWST	RFFST	RFMLF	RFUST[2:0]			RFE
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFE	Receive FIFO Enable	0: Receive FIFO disabled 1: Receive FIFO enabled	R/W
b3 to b1	RFUST[2:0]	Receive FIFO Unread Message Number Status Flag	b3 b1 0 0 0: No unread message 0 0 1: 1 unread message 0 1 0: 2 unread messages 0 1 1: 3 unread messages 1 0 0: 4 unread messages 1 0 1: Reserved 1 1 0: Reserved 1 1 1: Reserved	R
b4	RFMLF	Receive FIFO Message Lost Flag	0: No receive FIFO message lost has occurred 1: Receive FIFO message lost has occurred	R/W
b5	RFFST	Receive FIFO Full Status Flag	0: Receive FIFO is not full 1: Receive FIFO is full (4 unread messages)	R
b6	RFWST	Receive FIFO Buffer Warning Status Flag	0: Receive FIFO is not buffer warning 1: Receive FIFO is buffer warning (3 unread messages)	R
b7	RFEST	Receive FIFO Empty Status Flag	0: Unread message in receive FIFO 1: No unread message in receive FIFO	R

Write to RFCR in CAN operation mode or CAN halt mode.

RFE Bit (Receive FIFO Enable)

When the RFE bit is set to 1, the receive FIFO is enabled.

When the RFE bit is set to 0, the receive FIFO is disabled for reception and becomes empty (RFEST flag = 1). Write 0 to the RFE bit simultaneously with setting the RFMLF flag.

Do not set this bit to 1 in normal mailbox mode (MBM bit in CTLR = 0). Due to hardware protection, the RFE bit is not set to 0 by writing 0 by a program during the following period.

- Hardware protection is started
  - From the acceptance filter processing (the beginning of CRC field)
- Hardware protection is released
  - If the receive FIFO is specified to receive the incoming message, after the received data is stored into the receive FIFO or a CAN bus error occurs (i.e. maximum period of hardware protection is from the beginning of CRC field to the end of 7th bit of EOF)
  - If the receive FIFO is not specified to receive the message, after the acceptance filter processing

RFUST[2:0] Flags (Receive FIFO Unread Message Number Status Flag)

The RFUST[2:0] flags indicate the number of unread messages in the receive FIFO.

The value of the RFUST[2:0] flags is initialized to 000b when the RFE bit is set to 0.

RFMLF Flag (Receive FIFO Message Lost Flag)

The RFMLF flag is set to 1 (receive FIFO message lost has occurred) when the receive FIFO receives a new message and the receive FIFO is full. The timing for setting this bit to 1 is at the end of the 6th bit of EOF.
The RFMLF flag is set to 0 by writing 0 by a program (writing 1 has no effect). In both overwrite and overrun modes, the RFMLF flag cannot be set to 0 (receive FIFO message lost has not occurred) by writing 0 by a program due to hardware protection during five peripheral module clock (PCLKB) cycles following the sixth bit of EOF, if the receive FIFO is full and determined to receive a message.

RFFST Flag (Receive FIFO Full Status Flag)

The RFFST flag is set to 1 (receive FIFO is full) when the number of unread messages in the receive FIFO is 4. The RFFST flag is 0 (receive FIFO is not full) when the number of unread messages in the receive FIFO is less than 4. The RFFST flag is set to 0 when the RFE bit is 0.

RFWST Flag (Receive FIFO Buffer Warning Status Flag)

The RFWST flag is set to 1 (receive FIFO is buffer warning) when the number of unread messages in the receive FIFO is 3. The RFWST flag is 0 (receive FIFO is not buffer warning) when the number of unread messages in the receive FIFO is less than 3 or equal to 4. The RFWST flag is set to 0 when the RFE bit is 0.

RFEST Flag (Receive FIFO Empty Status Flag)

The RFEST flag is set to 1 (no unread message in receive FIFO) when the number of unread messages in the receive FIFO is 0. The RFEST flag is set to 1 when the RFE bit is set to 0. The RFEST flag is set to 0 (unread message in receive FIFO) when the number of unread messages in the receive FIFO is one or more.

Figure 43.2 shows the receive FIFO mailbox operation.

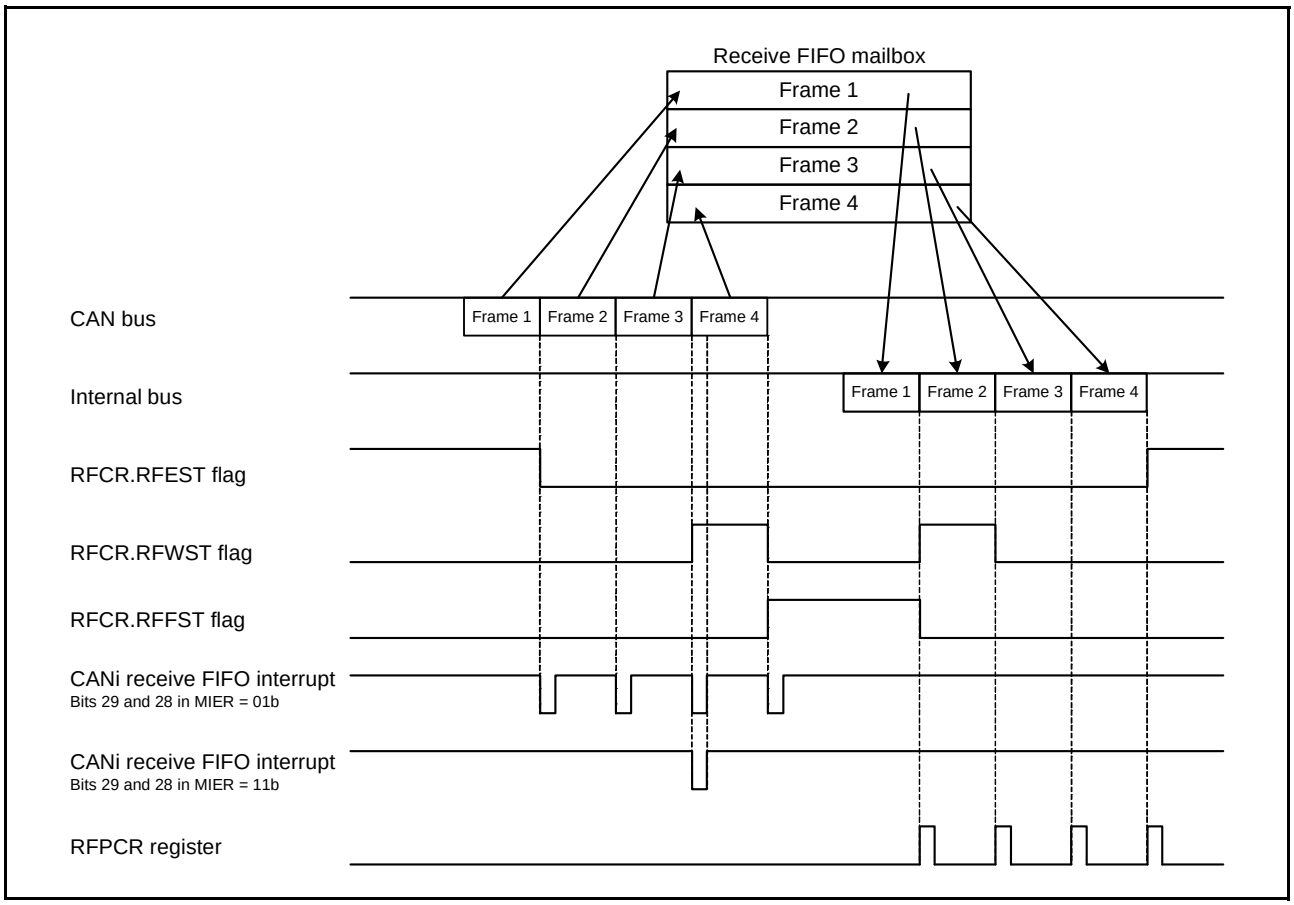


Figure 43.2 Receive FIFO Mailbox Operation (Bits 29 and 28 in MIER = 01b or 11b)

43.2.10 Receive FIFO Pointer Control Register (RFPCR)

Address(es): CAN0.RFPCR 0009 0849h, CAN1.RFPCR 0009 1849h, CAN2.RFPCR 0009 2849h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Description	R/W
b7 to b0	The CPU-side pointer for the receive FIFO is incremented by writing FFh to RFPCR.	W

When the receive FIFO is not empty, write FFh to RFPCR by a program to increment the CPU-side pointer for the receive FIFO to the next mailbox location.
Do not write to RFPCR when the RFE bit in RFCR is 0 (receive FIFO disabled).
Both the CAN-side pointer and the CPU-side pointer are incremented when a new message is received and the RFFST flag is 1 (receive FIFO is full) in overwrite mode. When the RFMLF flag is 1 in this condition, the CPU-side pointer cannot be incremented by writing to RFPCR by a program.

43.2.11 Transmit FIFO Control Register (TFCR)

Address(es): CAN0.TFCR 0009 084Ah, CAN1.TFCR 0009 184Ah, CAN2.TFCR 0009 284Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	TFEST	TFFST	—	—	TFUST[2:0]			TFE
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TFE	Transmit FIFO Enable	0: Transmit FIFO disabled 1: Transmit FIFO enabled	R/W
b3 to b1	TFUST[2:0]	Transmit FIFO Unsent Message Number Status Flag	b3 b2 b1 0 0 0: No unsent message 0 0 1: 1 unsent message 0 1 0: 2 unsent messages 0 1 1: 3 unsent messages 1 0 0: 4 unsent messages 1 0 1: Reserved 1 1 0: Reserved 1 1 1: Reserved	R
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	TFFST	Transmit FIFO Full Status Flag	0: Transmit FIFO is not full 1: Transmit FIFO is full (4 unsent messages)	R
b7	TFEST	Transmit FIFO Empty Status Flag	0: Unsent message in transmit FIFO 1: No unsent message in transmit FIFO	R

Write to TFCR in CAN operation mode or CAN halt mode.

TFE Bit (Transmit FIFO Enable)

When the TFE bit is set to 1, the transmit FIFO is enabled.
When the TFE bit is set to 0, the transmit FIFO becomes empty (TFEST flag = 1) and then unsent messages from the transmit FIFO are lost as described below:

- Immediately if a message from the transmit FIFO is not scheduled for the next transmission or during transmission
- Following the completion of transmission, a CAN bus error, CAN bus arbitration-lost, or entry to CAN halt mode if a message from the transmit FIFO is scheduled for the next transmission or already during transmission

Before setting the TFE bit to 1 again, ensure that the TFEST flag has been set to 1. After setting the TFE bit to 1, write transmit data into MB24.
Do not set the TFE bit to 1 in normal mailbox mode (MBM bit in CTLR = 0).

TFUST[2:0] Flags (Transmit FIFO Unsent Message Number Status Flag)

The TFUST[2:0] flags indicate the number of unsent messages in the transmit FIFO.
The TFUST[2:0] flags are set to 000b after TFE bit is cleared to 0 and transmission is aborted or completed.

TFFST Flag (Transmit FIFO Full Status Flag)

The TFFST flag is set to 1 (transmit FIFO is full) when the number of unsent messages in the transmit FIFO is 4. The TFFST flag is set to 0 (transmit FIFO is not full) when the number of unsent messages in the transmit FIFO is less than 4. The TFFST flag is set to 0 when transmission from the transmit FIFO has been aborted.

TFEST Flag (Transmit FIFO Empty Status Flag)

The TFEST flag is set to 1 (no message in transmit FIFO) when the number of unsent messages in the transmit FIFO is 0. The TFEST flag is set to 1 when transmission from the transmit FIFO has been aborted. The TFEST flag is set to 0 (message in transmit FIFO) when the number of unsent messages in the transmit FIFO is not 0.

Figure 43.3 shows the transmit FIFO mailbox operation.

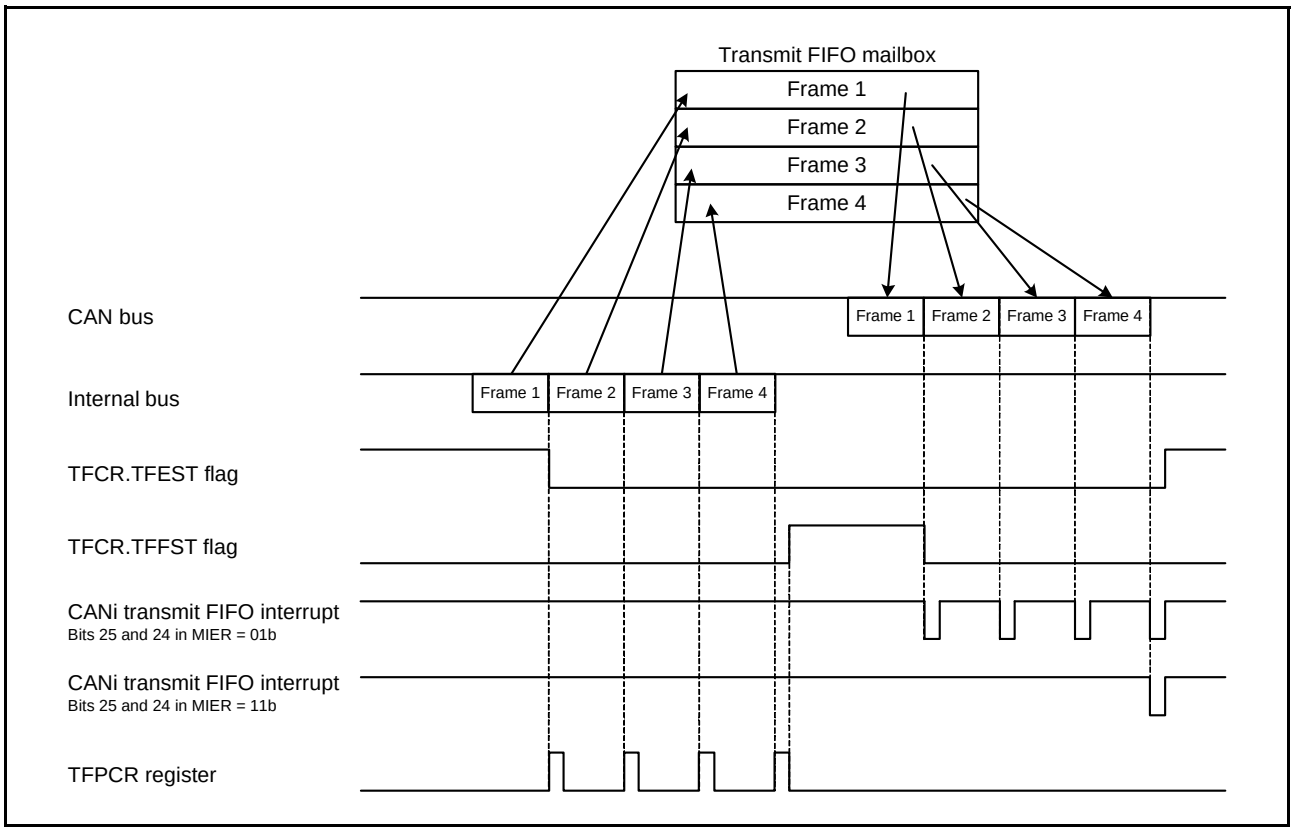


Figure 43.3 Transmit FIFO Mailbox Operation (Bits 25 and 24 in MIER = 01b or 11b)

43.2.12 Transmit FIFO Pointer Control Register (TFPCR)

Address(es): CAN0.TFPCR 0009 084Bh, CAN1.TFPCR 0009 184Bh, CAN2.TFPCR 0009 284Bh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Description	R/W
b7 to b0	The CPU-side pointer for the transmit FIFO is incremented by writing FFh to TFPCR.	W

When the transmit FIFO is not full, write FFh to TFPCR by a program to increment the CPU-side pointer for the transmit FIFO to the next mailbox location.
Do not write to TFPCR when the TFE bit in TFCR is 0 (transmit FIFO disabled).

43.2.13 Status Register (STR)

Address(es): CAN0.STR 0009 0842h, CAN1.STR 0009 1842h, CAN2.STR 0009 2842h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	RECST	TRMST	BOST	EPST	SLPST	HLTST	RSTST	EST	TABST	FMLST	NMLST	TFST	RFST	SDST	NDST
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	NDST	NEWDATA Status Flag	0: No mailbox with NEWDATA flag = 1 1: Mailbox(es) with NEWDATA flag = 1	R
b1	SDST	SENTDATA Status Flag	0: No mailbox with SENTDATA flag = 1 1: Mailbox(es) with SENTDATA flag = 1	R
b2	RFST	Receive FIFO Status Flag	0: No message in receive FIFO (empty) 1: Message in receive FIFO	R
b3	TFST	Transmit FIFO Status Flag	0: Transmit FIFO is full 1: Transmit FIFO is not full	R
b4	NMLST	Normal Mailbox Message Lost Status Flag	0: No mailbox with MSGLOST flag = 1 1: Mailbox(es) with MSGLOST flag = 1	R
b5	FMLST	FIFO Mailbox Message Lost Status Flag	0: RFMLF flag = 0 1: RFMLF flag = 1	R
b6	TABST	Transmission Abort Status Flag	0: No mailbox with TRMABT flag = 1 1: Mailbox(es) with TRMABT flag = 1	R
b7	EST	Error Status Flag	0: No error occurred 1: Error occurred	R
b8	RSTST	CAN Reset Status Flag	0: Not in CAN reset mode 1: In CAN reset mode	R
b9	HLTST	CAN Halt Status Flag	0: Not in CAN halt mode 1: In CAN halt mode	R
b10	SLPST	CAN Sleep Status Flag	0: Not in CAN sleep mode 1: In CAN sleep mode	R
b11	EPST	Error-Passive Status Flag	0: Not in error-passive state 1: In error-passive state	R
b12	BOST	Bus-Off Status Flag	0: Not in bus-off state 1: In bus-off state	R
b13	TRMST	Transmit Status Flag (transmitter)	0: Bus idle or reception in progress 1: Transmission in progress or in bus-off state	R
b14	RECST	Receive Status Flag (receiver)	0: Bus idle or transmission in progress 1: Reception in progress	R
b15	—	Reserved	The read value is 0.	R

NDST Flag (NEWDATA Status Flag)

The NDST flag is set to 1 when at least one NEWDATA flag in MCTLj (j = 0 to 31) is 1 regardless of the value of MIER. The NDST flag is set to 0 when all NEWDATA flags are 0.

SDST Flag (SENTDATA Status Flag)

The SDST flag is set to 1 when at least one SENTDATA flag in MCTLj (j = 0 to 31) is 1 regardless of the value of MIER. The SDST flag is set to 0 when all SENTDATA flags are 0.

RFST Flag (Receive FIFO Status Flag)

The RFST flag is set to 1 when the receive FIFO is not empty. The RFST flag is set to 0 when the receive FIFO is empty or normal mailbox mode is selected.

TFST Flag (Transmit FIFO Status Flag)

The TFST flag is set to 1 when the transmit FIFO is not full. The TFST flag is set to 0 when the transmit FIFO is full or normal mailbox mode is selected.

NMLST Flag (Normal Mailbox Message Lost Status Flag)

The NMLST flag is set to 1 when at least one MSGLOST flag in MCTLj (j = 0 to 31) is 1 regardless of the value of MIER. The NMLST flag is set to 0 when all MSGLOST flags are 0.

FMLST Flag (FIFO Mailbox Message Lost Status Flag)

The FMLST flag is set to 1 when the RFMLF flag in RFCR is 1 regardless of the value of MIER. The FMLST flag is set to 0 when the RFMLF flag is 0.

TABST Flag (Transmission Abort Status Flag)

The TABST flag is set to 1 when at least one TRMABT flag in MCTLj (j = 0 to 31) is 1 regardless of the value of MIER. The TABST flag is set to 0 when all TRMABT flags are 0.

EST Flag (Error Status Flag)

The EST flag is set to 1 when at least one error is detected by EIFR regardless of the value of EIER. The EST flag is set to 0 when no error is detected by EIFR.

RSTST Flag (CAN Reset Status Flag)

The RSTST flag is set to 1 when the CAN module is in CAN reset mode. The RSTST flag is 0 when the CAN module is not in CAN reset mode. Even when the state is changed from CAN reset mode to CAN sleep mode, the RSTST flag remains 1.

HLTST Flag (CAN Halt Status Flag)

The HLTST flag is set to 1 when the CAN module is in CAN halt mode. The HLTST flag is set to 0 when the CAN module is not in CAN halt mode. Even when the state is changed from CAN halt mode to CAN sleep mode, the HLTST flag remains 1.

SLPST Flag (CAN Sleep Status Flag)

The SLPST flag is set to 1 when the CAN module is in CAN sleep mode. The SLPST flag is set to 0 when the CAN module is not in CAN sleep mode.

EPST Flag (Error-Passive Status Flag)

The EPST flag is set to 1 when the value of TECR or RECR exceeds 127 and the CAN module is in the error-passive state ($128 \leq TEC < 256$ or $128 \leq REC < 256$). The EPST flag is set to 0 when the CAN module is not in the error-passive state.

BOST Flag (Bus-Off Status Flag)

The BOST flag is set to 1 when the value of TECR exceeds 255 and the CAN module is in the bus-off state ($TEC \geq 256$). The BOST flag is set to 0 when the CAN module is not in the bus-off state.

TRMST Flag (Transmit Status Flag (transmitter))

The TRMST flag is set to 1 when the CAN module performs as a transmitter node or is in the bus-off state. The TRMST flag is set to 0 when the CAN module performs as a receiver node or is in the bus-idle state.

RECST Flag (Receive Status Flag (receiver))

The RECST flag is set to 1 when the CAN module performs as a receiver node. The RECST flag is set to 0 when the CAN module performs as a transmitter node or is in the bus-idle state.

43.2.14 Mailbox Search Mode Register (MSMR)

Address(es): CAN0.MSMR 0009 0853h, CAN1.MSMR 0009 1853h, CAN2.MSMR 0009 2853h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MBSM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MBSM[1:0]	Mailbox Search Mode Select	b1 b0 0 0: Receive mailbox search mode 0 1: Transmit mailbox search mode 1 0: Message lost search mode 1 1: Channel search mode	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Write to MSMR in CAN operation mode or CAN halt mode.

MBSM[1:0] Bits (Mailbox Search Mode Select)

The MBSM[1:0] bits select the search mode for the mailbox search function.

When the MBSM[1:0] bits are 00b, receive mailbox search mode is selected. In this mode, the search targets are the NEWDATA flag in MCTLj (j = 0 to 31) for the normal mailbox and the RFEST flag in RFCR.

When the MBSM[1:0] bits are 01b, transmit mailbox search mode is selected. In this mode, the search target is the SENTDATA flag in MCTLj.

When the MBSM[1:0] bits are 10b, message lost search mode is selected. In this mode, the search targets are the MSGLOST flag in MCTLj for the normal mailbox and the RFMLF flag in RFCR.

When the MBSM[1:0] bits are 11b, channel search mode is selected. In this mode, the search target is CSSR. Refer to section 43.2.16, Channel Search Support Register (CSSR).

43.2.15 Mailbox Search Status Register (MSSR)

Address(es): CAN0.MSSR 0009 0852h, CAN1.MSSR 0009 1852h, CAN2.MSSR 0009 2852h

	b7	b6	b5	b4	b3	b2	b1	b0
	SEST	—	—	MBNST[4:0]				
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	MBNST[4:0]	Search Result Mailbox Number Status Flag	These bits output the smallest mailbox number that is searched in each mode of MSMR.	R
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SEST	Search Result Status Flag	0: Search result found 1: No search result	R

MBNST[4:0] Flags (Search Result Mailbox Number Status Flag)

The MBNST[4:0] flags output the smallest mailbox number that is searched in each mode of MSMR. In receive mailbox search mode, transmit mailbox search mode, and message lost search mode, the value of the mailbox i.e., the search result to be output, is updated as described below:

- When the NEWDATA, SENTDATA or MSGLOST flag for the output mailbox is set to 0
- When the NEWDATA, SENTDATA or MSGLOST flag for a higher-priority mailbox is set to 1

If the MBSM[1:0] bits are set to 00b (receive mailbox search mode) or 10b (message lost search mode), the receive FIFO (mailbox [28]) is output when the receive FIFO is not empty and there are no unread received messages or no lost messages in any of the normal mailboxes (mailboxes [0] to [23]). If the MBSM[1:0] bits are set to 01b (transmit mailbox search mode), the transmit FIFO (mailbox [24]) is not output. Table 43.6 lists the behavior of the MBNST[4:0] flags in FIFO mailbox mode.

In channel search mode, the MBNST[4:0] flags output the corresponding channel number. After MSSR is read by a program, the next target channel number is output.

SEST Flag (Search Result Status Flag)

The SEST flag is set to 1 (no search result) when no corresponding mailbox is found after searching all mailboxes. For example, in transmit mailbox search mode, the SEST flag is set to 1 when no SENTDATA flag for mailboxes is 1. The SEST flag is set to 0 when at least one SENTDATA flag is 1. When the SEST flag is 1, the value of the MBNST[4:0] flags is undefined.

Table 43.6 Behavior of MBNST[4:0] Flags in FIFO Mailbox Mode

MBSM[1:0] Bits	Mailbox [24] (Transmit FIFO)	Mailbox [28] (Receive FIFO)
00b	Mailbox [24] is not output.	Mailbox [28] is output when no MCTLj.NEWDATA flag for the normal mailboxes is set to 1 (new message is being stored or has been stored to the mailbox) and the receive FIFO is not empty.
01b		Mailbox [28] is not output.
10b		Mailbox [28] is output when no MCTLj.MSGLOST flag for the normal mailboxes is set to 1 (message is overwritten or overrun) and the RFCR.RFMLF bit is set to 1 (receive FIFO message lost has occurred) in the receive FIFO.
11b		Mailbox [28] is not output.

43.2.16 Channel Search Support Register (CSSR)

Address(es): CAN0.CSSR 0009 0851h, CAN1.CSSR 0009 1851h, CAN2.CSSR 0009 2851h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Description	R/W
b7 to b0	When the value for the channel search is input, the channel number is output to MSSR.	R/W

The bits in CSSR, which are set to 1, are encoded by an 8/3 encoder (the LSB position has the higher priority) and output to the MBNST[4:0] flags in MSSR.
MSSR outputs the updated value whenever MSSR is read by a program.
Write to CSSR only when the MSMR.MBSM[1:0] bits are 11b (channel search mode). Write to CSSR in CAN operation mode or CAN halt mode.

Figure 43.4 shows the write and read of CSSR and MSSR.

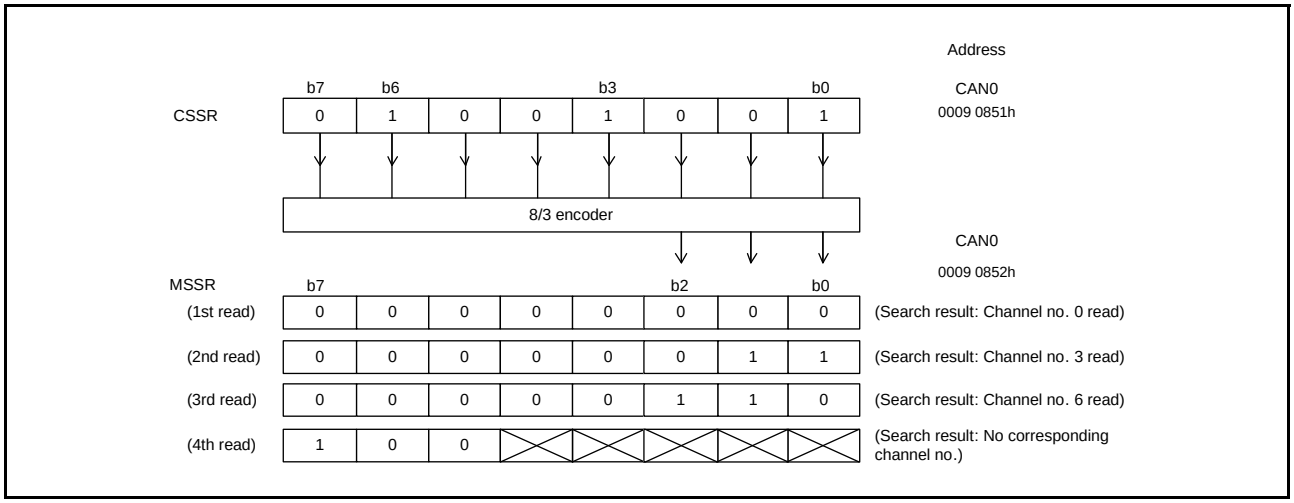


Figure 43.4 Write and Read of CSSR and MSSR

The value of CSSR is also updated whenever MSSR is read. When read, the value prior to conversion by the 8/3 encoder can be read.

43.2.17 Acceptance Filter Support Register (AFSR)

Address(es): CAN0.AFSR 0009 0856h, CAN1.AFSR 0009 1856h, CAN2.AFSR 0009 2856h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Description	R/W
b15 to b0	After the standard ID of a received message is written, the value converted for data table search can be read.	R/W

Note: Write to AFSR in CAN operation mode or CAN halt mode.

The acceptance filter support unit (ASU) can be used for data table (8 bits × 256) search. In the data table, all standard IDs created by the user are set to be valid/invalid in bit units. When AFSR is written with data in 16-bit units including the SID[10:0] bit in MBj (j = 0 to 31), in which a received standard ID is stored, a decoded row (byte offset) position and column (bit) position for data table search can be read. The ASU can be used for standard (11-bit) IDs only.
The ASU is enabled in the following cases:

- When the ID to receive cannot be masked by the acceptance filter
  (Example) IDs to receive: 078h, 087h, and 111h
- When there are too many IDs to receive and software filtering time is expected to be shortened
  It should be noted that AFSR cannot be set in CAN reset mode.

Figure 43.5 shows the write and read of AFSR.

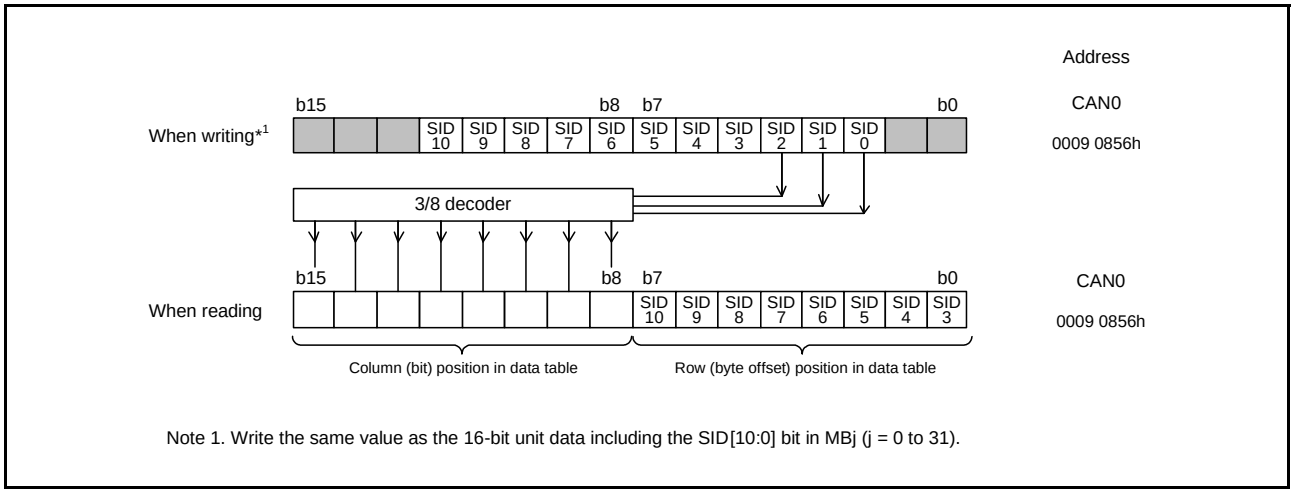


Figure 43.5 Write and Read of AFSR

43.2.18 Error Interrupt Enable Register (EIER)

Address(es): CAN0.EIER 0009 084Ch, CAN1.EIER 0009 184Ch, CAN2.EIER 0009 284Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	BLIE	OLIE	ORIE	BORIE	BOEIE	EPIE	EWIE	BEIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BEIE	Bus Error Interrupt Enable	0: Bus error interrupt disabled 1: Bus error interrupt enabled	R/W
b1	EWIE	Error-Warning Interrupt Enable	0: Error-warning interrupt disabled 1: Error-warning interrupt enabled	R/W
b2	EPIE	Error-Passive Interrupt Enable	0: Error-passive interrupt disabled 1: Error-passive interrupt enabled	R/W
b3	BOEIE	Bus-Off Entry Interrupt Enable	0: Bus-off entry interrupt disabled 1: Bus-off entry interrupt enabled	R/W
b4	BORIE	Bus-Off Recovery Interrupt Enable	0: Bus-off recovery interrupt disabled 1: Bus-off recovery interrupt enabled	R/W
b5	ORIE	Overrun Interrupt Enable	0: Receive overrun interrupt disabled 1: Receive overrun interrupt enabled	R/W
b6	OLIE	Overload Frame Transmit Interrupt Enable	0: Overload frame transmit interrupt disabled 1: Overload frame transmit interrupt enabled	R/W
b7	BLIE	Bus Lock Interrupt Enable	0: Bus lock interrupt disabled 1: Bus lock interrupt enabled	R/W

EIER is used to enable or disable the error interrupt individually for each error interrupt source in EIFR.
Write to EIER in CAN reset mode.

BEIE Bit (Bus Error Interrupt Enable)

When the BEIE bit is 0, no error interrupt request is generated even if the BEIF flag in EIFR is set to 1. When the BEIE bit is 1, an error interrupt request is generated if the BEIF flag is set to 1.

EWIE Bit (Error-Warning Interrupt Enable)

When the EWIE bit is 0, no error interrupt request is generated even if the EWIF flag in EIFR is set to 1. When the EWIE bit is 1, an error interrupt request is generated if the EWIF flag is set to 1.

EPIE Bit (Error-Passive Interrupt Enable)

When the EPIE bit is 0, no error interrupt request is generated even if the EPIF flag in EIFR is set to 1. When the EPIE bit is 1, an error interrupt request is generated if the EPIF flag is set to 1.

BOEIE Bit (Bus-Off Entry Interrupt Enable)

When the BOEIE bit is 0, no error interrupt request is generated even if the BOEIF flag in EIFR is set to 1. When the BOEIE bit is 1, an error interrupt request is generated if the BOEIF flag is set to 1.

BORIE Bit (Bus-Off Recovery Interrupt Enable)

When the BORIE bit is 0, an error interrupt request is not generated even if the BORIF flag in EIFR is set to 1. When the BORIE bit is set to 1, an error interrupt request is generated if the BORIF flag is set to 1.

ORIE Bit (Overrun Interrupt Enable)

When the ORIE bit is 0, an error interrupt request is not generated even if the ORIF flag in EIFR is set to 1. When the ORIE bit is 1, an error interrupt request is generated if the ORIF flag is set to 1.

OLIE Bit (Overload Frame Transmit Interrupt Enable)

When the OLIE bit is 0, no error interrupt request is generated even if the OLIF flag in EIFR is set to 1. When the OLIE bit is 1, an error interrupt request is generated if the OLIF flag is set to 1.

BLIE Bit (Bus Lock Interrupt Enable)

When the BLIE bit is 0, no error interrupt request is generated even if the BLIF flag in EIFR is set to 1. When the BLIE bit is 1, an error interrupt request is generated if the BLIF flag is set to 1.

43.2.19 Error Interrupt Factor Judge Register (EIFR)

Address(es): CAN0.EIFR 0009 084Dh, CAN1.EIFR 0009 184Dh, CAN2.EIFR 0009 284Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	BLIF	OLIF	ORIF	BORIF	BOEIF	EPIF	EWIF	BEIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BEIF	Bus Error Detect Flag	0: No bus error detected 1: Bus error detected	R/W
b1	EWIF	Error-Warning Detect Flag	0: No error-warning detected 1: Error-warning detected	R/W
b2	EPIF	Error-Passive Detect Flag	0: No error-passive detected 1: Error-passive detected	R/W
b3	BOEIF	Bus-Off Entry Detect Flag	0: No bus-off entry detected 1: Bus-off entry detected	R/W
b4	BORIF	Bus-Off Recovery Detect Flag	0: No bus-off recovery detected 1: Bus-off recovery detected	R/W
b5	ORIF	Receive Overrun Detect Flag	0: No receive overrun detected 1: Receive overrun detected	R/W
b6	OLIF	Overload Frame Transmission Detect Flag	0: No overload frame transmission detected 1: Overload frame transmission detected	R/W
b7	BLIF	Bus Lock Detect Flag	0: No bus lock detected 1: Bus lock detected	R/W

If an event corresponding to each bit occurs, the corresponding bit in EIFR is set to 1 regardless of the setting of EIER. To set each bit to 0, write 0 by a program. If the set timing occurs simultaneously with the clear timing by the program, the bit becomes 1.

When a single bit is set to 0 by a program, do not use the logic operation instruction (AND) – use the transfer instruction (MOV) to ensure that only the specified bit is set to 0 and the other bits are set to 1. Writing 1 has no effect to these bit values.

BEIF Flag (Bus Error Detect Flag)

The BEIF flag is set to 1 when a bus error is detected.

EWIF Flag (Error-Warning Detect Flag)

The EWIF flag is set to 1 when the value of the receive error counter (REC) or transmit error counter (TEC) exceeds 95. The EWIF flag is set to 1 only when the REC or TEC initially exceeds 95. Thus, if 0 is written to the EWIF flag by a program while the REC or TEC remains greater than 95, the EWIF flag is not set to 1 until the REC and TEC go below 95 and then REC or TEC exceeds 95 again.

EPIF Flag (Error-Passive Detect Flag)

The EPIF flag is set to 1 when the CAN error state becomes error-passive (the REC (receive error counter) or TEC (transmit error counter) value exceeds 127).
The EPIF flag is set to 1 only when the REC or TEC initially exceeds 127. Thus, if 0 is written by a program while the REC or TEC remains greater than 127, the EPIF flag is not set to 1 until the REC and TEC go below 127 and then REC or TEC exceeds 127 again.

BOEIF Flag (Bus-Off Entry Detect Flag)

The BOEIF flag is set to 1 when the CAN error state becomes bus-off (the TEC (transmit error counter) value exceeds 255). The BOEIF flag is also set to 1 when the BOM[1:0] bits in CTLR are 01b (entry to CAN halt mode automatically at bus-off entry) and the CAN module becomes the bus-off state.

BORIF Flag (Bus-Off Recovery Detect Flag)

The BORIF flag is set to 1 when the CAN module recovers from the bus-off state normally by detecting 11 consecutive bits 128 times in the following conditions:

- When the BOM[1:0] bits in CTLR are 00b
- When the BOM[1:0] bits in CTLR are 10b
- When the BOM[1:0] bits in CTLR are 11b

However, the BORIF flag is not set to 1 if the CAN module recovers from the bus-off state in the following conditions:

- When the CANM[1:0] bits in CTLR are set to 01b or 11b (CAN reset mode)
- When the RBOC bit in CTLR is set to 1 (forcible return from bus-off)
- When the BOM[1:0] bits in CTLR are set to 01b
- When the BOM[1:0] bits in CTLR are set to 11b and the CANM[1:0] bits in CTLR are set to 10b (CAN halt mode) before normal recovery occurs

Table 43.7 lists the behavior of BOEIF and BORIF flags according to the CTLR.BOM[1:0] bit setting.

Table 43.7 Behavior of BOEIF and BORIF Flags according to CTLR.BOM[1:0] Bit Setting

BOM[1:0] Bits	BOEIF Flag	BORIF Flag
00b	Set to 1 on entry to the bus-off state.	Set to 1 on exit from the bus-off state.
01b		Do not set to 1.
10b		Set to 1 on exit from the bus-off state.
11b		Set to 1 if normal bus-off recovery occurs before the CANM[1:0] bits are set to 10b (CAN halt mode).

ORIF Flag (Receive Overrun Detect Flag)

The ORIF flag is set to 1 when a receive overrun occurs. This flag is not to set to 1 in overwrite mode.
In overwrite mode, a reception complete interrupt request is generated if an overwrite condition occurs and the ORIF flag is not set to 1.
In normal mailbox mode, if an overrun occurs in any of mailboxes [0] to [31] in overrun mode, this flag is set to 1. In

FIFO mailbox mode, if an overrun occurs in any of mailboxes [0] to [23] or the receive FIFO in overrun mode, this flag is set to 1.

OLIF Flag (Overload Frame Transmission Detect Flag)

The OLIF flag is set to 1 if the transmitting condition of an overload frame is detected when the CAN module performs transmission or reception.

BLIF Flag (Bus Lock Detect Flag)

The BLIF flag is set to 1 if 32 consecutive dominant bits are detected on the CAN bus while the CAN module is in CAN operation mode.

After the BLIF flag is set to 1, a bus lock is detected again under either of the following conditions:

- After this flag is set to 0 from 1, recessive bits are detected
- After this flag is set to 0 from 1, the CAN module enters CAN reset mode or CAN halt mode and then enters CAN operation mode again (internal reset).

43.2.20 Receive Error Count Register (RECR)

Address(es): CAN0.RECR 0009 084Eh, CAN1.RECR 0009 184Eh, CAN2.RECR 0009 284Eh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Description	R/W
b7 to b0	Receive error count function RECR increments or decrements the counter value according to the error status of the CAN module during reception.	R

RECR indicates the value of the receive error counter.

Refer to the ISO 11898-1 Standards about the increment/decrement conditions of the receive error counter.

The value of RECR in the bus-off state is undefined.

43.2.21 Transmit Error Count Register (TECR)

Address(es): CAN0.TECR 0009 084Fh, CAN1.TECR 0009 184Fh, CAN2.TECR 0009 284Fh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Description	R/W
b7 to b0	Transmit error count function TECR increments or decrements the counter value according to the error status of the CAN module during transmission.	R

TECR indicates the value of the transmit error counter.
Refer to the ISO 11898-1 Standards about the increment/decrement conditions of the transmit error counter.
The value of TECR in the bus-off state is undefined.

43.2.22 Error Code Store Register (ECSR)

Address(es): CAN0.ECSR 0009 0850h, CAN1.ECSR 0009 1850h, CAN2.ECSR 0009 2850h

	b7	b6	b5	b4	b3	b2	b1	b0
	EDPM	ADEF	BE0F	BE1F	CEF	AEF	FEF	SEF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SEF	Stuff Error Flag*1,*2	0: No stuff error detected 1: Stuff error detected	R/W
b1	FEF	Form Error Flag*1,*2	0: No form error detected 1: Form error detected	R/W
b2	AEF	ACK Error Flag*1,*2	0: No ACK error detected 1: ACK error detected	R/W
b3	CEF	CRC Error Flag*1,*2	0: No CRC error detected 1: CRC error detected	R/W
b4	BE1F	Bit Error (recessive) Flag*1,*2	0: No bit error (recessive) detected 1: Bit error (recessive) detected	R/W
b5	BE0F	Bit Error (dominant) Flag*1,*2	0: No bit error (dominant) detected 1: Bit error (dominant) detected	R/W
b6	ADEF	ACK Delimiter Error Flag*1,*2	0: No ACK delimiter error detected 1: ACK delimiter error detected	R/W
b7	EDPM	Error Display Mode Select*3,*4	0: Output of first detected error code 1: Output of accumulated error code	R/W

Note 1. Writing 1 has no effect to these flag values.
Note 2. To write 0 to SEF, FEF, AEF, CEF, BE1F, BE0F, and ADEF flags, do not use the logic operation instruction (AND). Use the transfer (MOV) instruction to ensure that only the specified flag is set to 0 and the other flags are set to 1.
Note 3. Write to the EDPM bit in CAN reset mode or CAN halt mode.
Note 4. If more than one error condition is detected simultaneously, all related flags are set to 1.

ECSR can be used to monitor whether an error has occurred on the CAN bus.
Refer to the ISO 11898-1 Standards to check the generation conditions of each error.

To set each flag except for the EDPM bit to 0, write 0 by a program. If the timing at which each flag is set to 1 and the timing at which 0 is written by a program are the same, the relevant flag is set to 1.

SEF Flag (Stuff Error Flag)

The SEF flag is set to 1 when a stuff error is detected.

FEF Flag (Form Error Flag)

The FEF flag is set to 1 when a form error is detected.

AEF Flag (ACK Error Flag)

The AEF flag is set to 1 when an ACK error is detected.

CEF Flag (CRC Error Flag)

The CEF flag is set to 1 when a CRC error is detected.

BE1F Flag (Bit Error (recessive) Flag)

The BE1F flag is set to 1 when a recessive bit error is detected.

BE0F Flag (Bit Error (dominant) Flag)

The BE0F flag is set to 1 when a dominant bit error is detected.

ADEF Flag (ACK Delimiter Error Flag)

The ADEF flag is set to 1 when a form error is detected with the ACK delimiter during transmission.

EDPM Bit (Error Display Mode Select)

The EDPM bit selects the output mode of ECSR. When the EDPM bit is set to 0, ECSR outputs the first error code. When the EDPM bit is set to 1, ECSR outputs the accumulated error code.

43.2.23 Time Stamp Register (TSR)

Address(es): CAN0.TSR 0009 0854h, CAN1.TSR 0009 1854h, CAN2.TSR 0009 2854h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	R/W
b15 to b0	Free-running counter value for the time stamp function	R

Note: Read TSR in 16-bit units.

When TSR is read, the value of the time stamp counter (16-bit free-running counter) at that moment is read.
The value of the time stamp counter reference clock is a multiple of 1 bit time, as configured by the TSPS[1:0] bits in CTLR.
The time stamp counter stops in CAN sleep mode and CAN halt mode, and is initialized in CAN reset mode.
The time stamp counter value is stored to bits TSL[7:0] and TSH[7:0] in MBj when a received message is stored in a receive mailbox.

43.2.24 Test Control Register (TCR)

Address(es): CAN0.TCR 0009 0858h, CAN1.TCR 0009 1858h, CAN2.TCR 0009 2858h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TSTM[1:0]		TSTE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSTE	CAN Test Mode Enable	0: CAN test mode disabled 1: CAN test mode enabled	R/W
b2, b1	TSTM[1:0]	CAN Test Mode Select	b2 b1 0 0: Other than CAN test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loopback) 1 1: Self-test mode 1 (internal loopback)	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TCR controls the CAN test mode. Write to TCR in CAN halt mode only.

(1) Listen-Only Mode

The ISO 11898-1 Standards recommend an optional bus monitoring mode. In listen-only mode, valid data frames and valid remote frames can be received. However, only recessive bits can be sent on the CAN bus, and the ACK bit, overload flag, and active error flag cannot be sent.

Listen-only mode can be used for baud rate detection.

Do not request transmission from any mailboxes in listen-only mode.

Figure 43.6 shows the connection when listen-only mode is selected (i = 0 to 2).

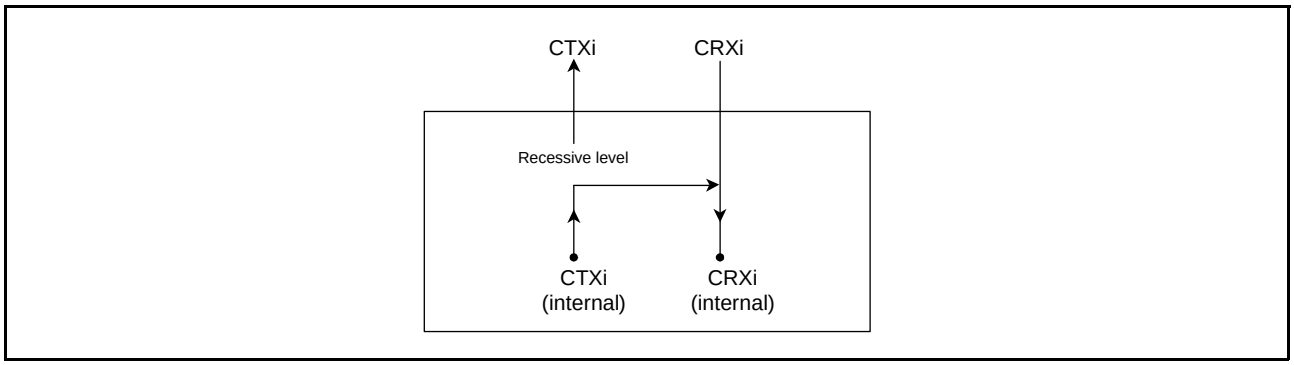


Figure 43.6 Connection when Listen-Only Mode is Selected (i = 0 to 2)

(2) Self-Test Mode 0 (External Loopback))

Self-test mode 0 is provided for CAN transceiver tests.
In self-test mode 0, the protocol controller treats its own transmitted messages as messages received via the CAN transceiver and stores them into the receive mailbox. To be independent from external stimulation, the protocol controller generates the ACK bit.
Connect the CTXi and CRXi pins to the transceiver.
Figure 43.7 shows the connection when self-test mode 0 is selected (i = 0 to 2).

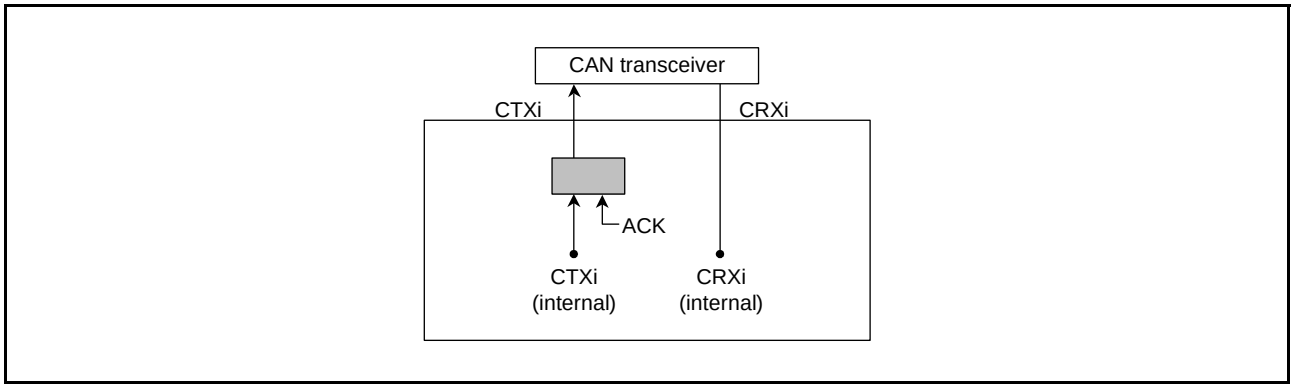


Figure 43.7 Connection when Self-Test Mode 0 is Selected (i = 0 to 2)

(3) Self-Test Mode 1 (Internal Loopback)

Self-test mode 1 is provided for self-test functions.
In self-test mode 1, the protocol controller treats its transmitted messages as received messages and stores them into the receive mailbox. To be independent from external stimulation, the protocol controller generates the ACK bit.
In self-test mode 1, the protocol controller performs an internal feedback from the internal CTXi pin to the internal CRXi pin. The input value of the external CRXi pin is ignored. The external CTXi pin outputs only recessive bits. The CTXi and CRXi pins do not need to be connected to the CAN bus or any external device.
Figure 43.8 shows the connection when self-test mode 1 is selected (i = 0 to 2).

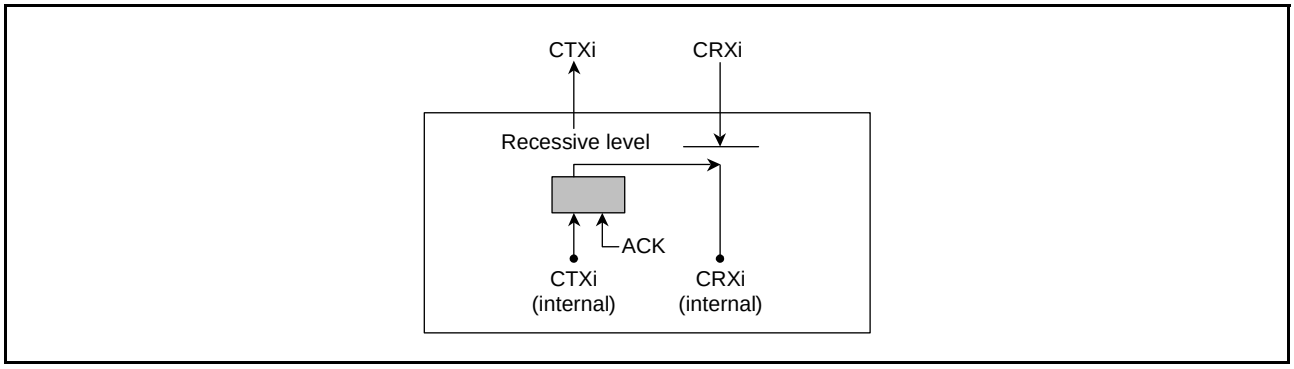


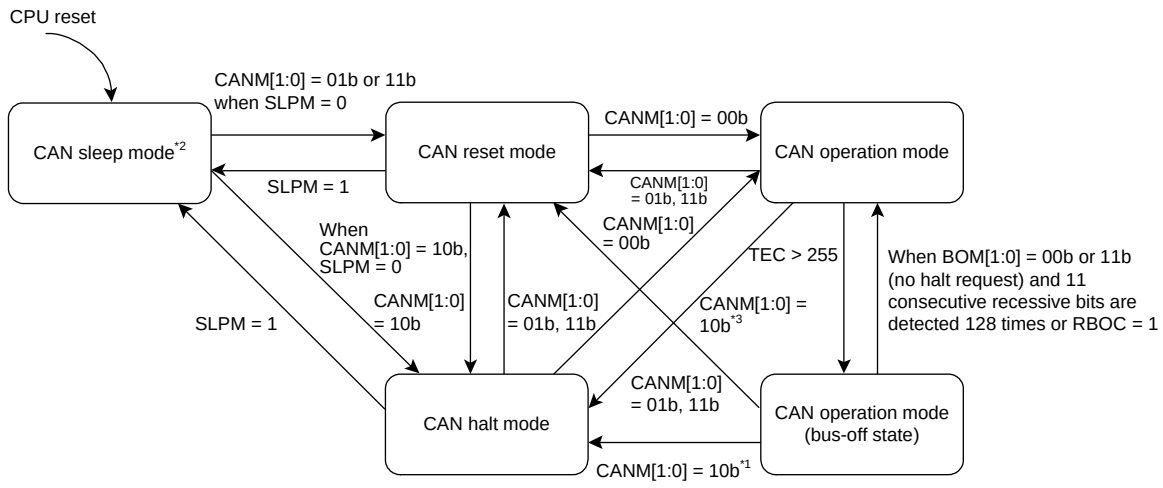
Figure 43.8 Connection when Self-Test Mode 1 is Selected (i = 0 to 2)

43.3 Operating Mode

The CAN module has the following four operating modes.

- CAN reset mode
- CAN halt mode
- CAN operation mode
- CAN sleep mode

Figure 43.9 shows the transition between CAN operating modes.



CANM, SLPM, BOM, RBOC: Bits in CTLR

Note 1. The transition timing from the bus-off state to CAN halt mode depends on the setting of the CTLR.BOM[1:0] bits.
When the CTLR.BOM[1:0] bits are 01b, the state transition timing is immediately after entering the bus-off state.
When the CTLR.BOM[1:0] bits are 10b, the state transition timing is at the end of the bus-off state.
When the CTLR.BOM[1:0] bits are 11b, the state transition timing is at the setting of the CTLR.CANM[1:0] bits to 10b (CAN halt mode).
Note 2. Change the CTLR.SLPM bit to set or cancel CAN sleep mode.
Note 3. While the CAN bus is locked at the dominant level, the CAN module does not enter CAN halt mode. In that case, enter CAN reset mode.

Figure 43.9 Transition between CAN Operating Modes

43.3.1 CAN Reset Mode

CAN reset mode is provided for CAN communication configuration.
When the CTLR.CANM[1:0] bits are set to 01b or 11b, the CAN module enters CAN reset mode. Then, the STR.RSTST flag is set to 1. Do not change the CTLR.CANM[1:0] bits until the RSTST flag is set to 1. Set BCR before exiting CAN reset mode to any other modes.
The following registers are initialized to their reset values after entering CAN reset mode, and their initial values are retained during CAN reset mode:

- MCTLj
- STR (except for the SLPST and TFST flags)
- EIFR
- RECR
- TECR
- TSR
- MSSR
- MSMR
- RFCR
- TFCR
- TCR
- ECSR (except for the EDPM bit)

The following registers retain their previous values even after entering CAN reset mode.

- CTLR
- STR (only the SLPST and TFST flags)
- MIER
- EIER
- BCR
- CSSR
- ECSR (only the EDPM bit)
- MBj
- MKR0 to MKR7
- FIDCR0 and FIDCR1
- MKIVLR
- AFSR
- RFPCR
- TFPCR

43.3.2 CAN Halt Mode

CAN halt mode is used for mailbox configuration and test mode setting.
When the CTLR.CANM[1:0] bits are set to 10b, CAN halt mode is selected. Then the STR.HLTST flag is set to 1. Do not change the CTLR.CANM[1:0] bits until the HLTST flag is set to 1.
See Table 43.8 for the state transition conditions when transmitting or receiving.
All registers except for RSTST, HLTST, and SLPST flags in STR remain unchanged when the CAN enters CAN halt mode.
Do not change CTLR (except for bits CANM[1:0] and SLPM) and EIER in CAN halt mode. BCR can be changed in CAN halt mode only when listen-only mode is selected for automatic baud rate detection.

Table 43.8 Operation in CAN Reset Mode and CAN Halt Mode

Mode	Receiver	Transmitter	Bus-Off
CAN reset mode (forcible transition) CANM[1:0] = 11b	CAN module enters CAN reset mode without waiting for the end of message reception.	CAN module enters CAN reset mode without waiting for the end of message transmission.	CAN module enters CAN reset mode without waiting for the end of bus-off recovery.
CAN reset mode CANM[1:0] = 01b	CAN module enters CAN reset mode without waiting for the end of message reception.	CAN module enters CAN reset mode after waiting for the end of message transmission.*1,*4	CAN module enters CAN reset mode without waiting for the end of bus-off recovery.
CAN halt mode	CAN module enters CAN halt mode after waiting for the end of message reception.*2,*3	CAN module enters CAN halt mode after waiting for the end of message transmission.*1,*2,*4	[When the BOM[1:0] bits are 00b] A halt request from a program will be accepted only after bus-off recovery. [When the BOM[1:0] bits are 01b] CAN module automatically enters CAN halt mode without waiting for the end of bus-off recovery (regardless of a halt request from a program). [When the BOM[1:0] bits are 10b] CAN module automatically enters CAN halt mode after waiting for the end of bus-off recovery (regardless of a halt request from a program). [When the BOM[1:0] bits are 11b] CAN module enters CAN halt mode (without waiting for the end of bus-off recovery) if a halt is requested by a program during bus-off.

CANM[1:0], BOM[1:0]: Bits in CTLR

Note 1. If several messages are requested to be transmitted, mode transition occurs after the completion of the first message transmission. In a case that the CAN reset mode is being requested during suspend transmission, mode transition occurs when the bus is idle, the next transmission ends, or the CAN module becomes a receiver.

Note 2. If the CAN bus is locked at the dominant level, the program can detect this state by monitoring the EIFR.BLIF flag. While the CAN bus is locked at the dominant level, the CAN module does not enter CAN halt mode. In that case, enter CAN reset mode.

Note 3. If a CAN bus error occurs during reception after CAN halt mode is requested, the CAN module transits to CAN halt mode. However, while the CAN bus is locked at the dominant level, the CAN module does not enter CAN halt mode.

Note 4. If a CAN bus error or arbitration-lost occurs during transmission after CAN reset mode or CAN halt mode is requested, the CAN module transits to the requested operating mode. However, while the CAN bus is locked at the dominant level, the CAN module does not enter CAN halt mode.

43.3.3 CAN Sleep Mode

CAN sleep mode is used for reducing current consumption by stopping the clock supply to the CAN module. After a reset from an MCU pin or software reset, the CAN module starts from CAN sleep mode.
When the SLPM bit in CTLR is set to 1, the CAN module enters CAN sleep mode. Then, the SLPST flag in STR is set to 1. Do not change the value of the SLPM bit until the SLPST flag is set to 1. The other registers remain unchanged when the CAN module enters CAN sleep mode.
Write to the SLPM bit in CAN reset mode and CAN halt mode. Do not change any registers (except for the SLPM bit) during CAN sleep mode. Read operation is still allowed.
When the SLPM bit is set to 0, the CAN module is released from CAN sleep mode. When the CAN module exits CAN sleep mode, the other registers remain unchanged.

43.3.4 CAN Operation Mode (Excluding Bus-Off State)

CAN operation mode is used for CAN communication.
When the CANM[1:0] bits in CTLR are set to 00b, the CAN module enters CAN operation mode.
Then RSTST and HLTST flags in STR are set to 0. Do not change the value of the CANM[1:0] until bits RSTST and HLTST flags are set to 0.
If 11 consecutive recessive bits are detected after entering CAN operation mode, the CAN module is in the following states:

- The CAN module becomes an active node on the network, thus enabling transmission and reception of CAN messages.
- Error monitoring of the CAN bus, such as receive and transmit error counters, is performed.

During CAN operation mode, the CAN module may be in one of the following three sub-modes, depending on the status of the CAN bus.

- Idle mode: Transmission or reception is not being performed.
- Receive mode: A CAN message sent by another node is being received.
- Transmit mode:A CAN message is being transmitted. The CAN module receives a message transmitted by the local node simultaneously when self-test mode 0 (TSTM[1:0] bits in TCR = 10b) or self-test mode 1 (TSTM[1:0] bits = 11b) is selected.

Figure 43.10 shows the sub-modes of CAN operation mode.

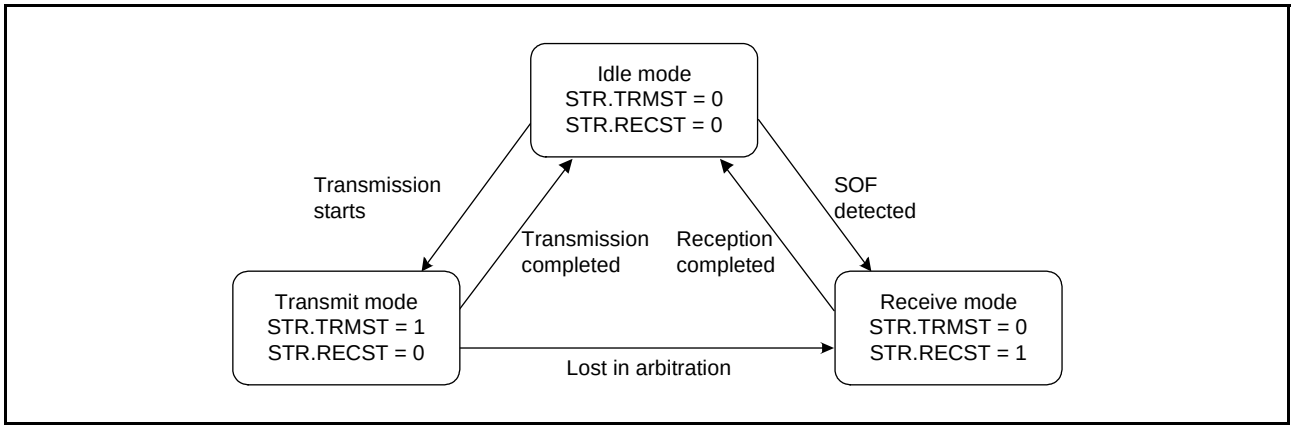


Figure 43.10 Sub-Modes of CAN Operation Mode

43.3.5 CAN Operation Mode (Bus-Off State)

The CAN module enters the bus-off state according to the increment/decrement rules for the transmit/error counters in the CAN Specifications.
The following cases apply when the CAN module is recovering from the bus-off state. When the CAN module is in the bus-off state, the values of the CAN-related registers, except for STR, EIFR, RECR, TECR and TSR, remain unchanged.

(1) When bits BOM[1:0] in CTLR are 00b (normal mode)

The CAN module enters the error-active state after it has completed the recovery from the bus-off state and CAN communication is enabled. The BORIF flag in EIFR is set to 1 (bus-off recovery detected) at this time.

(2) When bit RBOC in CTLR is set to 1 (forcible return from bus-off)

The CAN module enters the error-active state when it is in the bus-off state and the RBOC bit is set to 1. CAN communication is enabled again after 11b consecutive recessive bits are detected. The BORIF flag is not set to 1 at this time.

(3) When bits BOM[1:0] are 01b (automatic transition to CAN halt mode at bus-off entry)

The CAN module enters CAN halt mode when it reaches the bus-off state. The BORIF flag is not set to 1 at this time.

(4) When bits BOM[1:0] are 10b (automatic transition to CAN halt mode at bus-off end)

The CAN module enters CAN halt mode when it has completed the recovery from bus-off. The BORIF flag is set to 1 at this time.

(5) When bits BOM[1:0] are 11b (automatic transition to CAN halt mode by a program) and bits CANM[1:0] in CTLR are set to 10b (CAN halt mode) during bus-off state

The CAN module enters CAN halt mode when it is in the bus-off state and the CANM[1:0] bits are set to 10b (CAN halt mode). The BORIF flag is not set to 1 at this time.
If the CANM[1:0] bits are not set to 10b during bus-off, the same behavior as (1) applies.

43.4 CAN Communication Speed Setting

The following description explains about CAN communication speed setting.

43.4.1 CAN Clock Setting

The CAN module has a CAN clock selector.
The CAN clock can be set by the CCLKS bit and the BRP[9:0] bits in BCR.
Figure 43.11 shows a block diagram of the CAN clock generator.

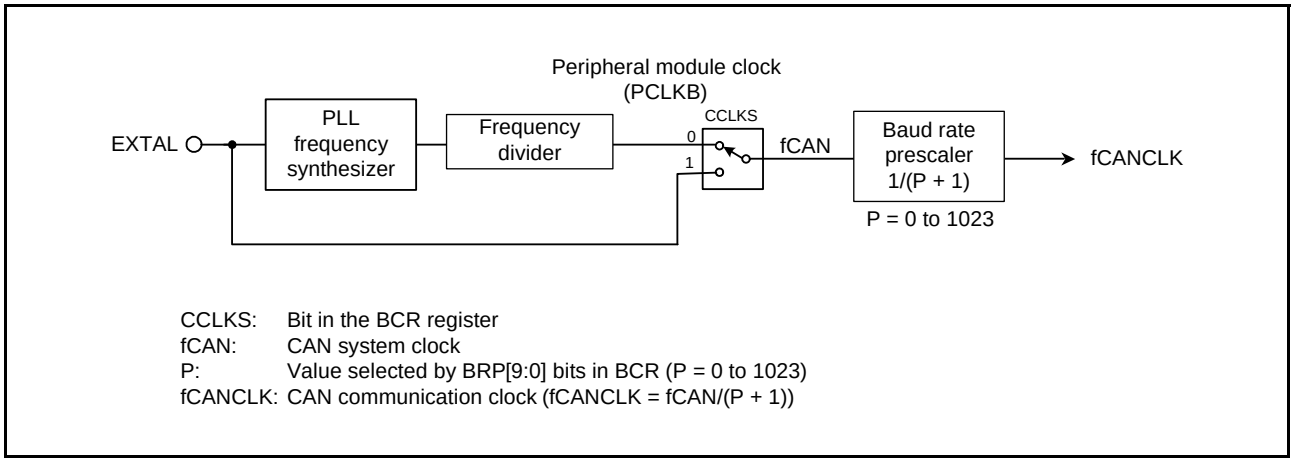


Figure 43.11 Block Diagram of CAN Clock Generator

43.4.2 Bit Timing Setting

The bit time consists of the following three segments.
Figure 43.12 shows the bit timing.

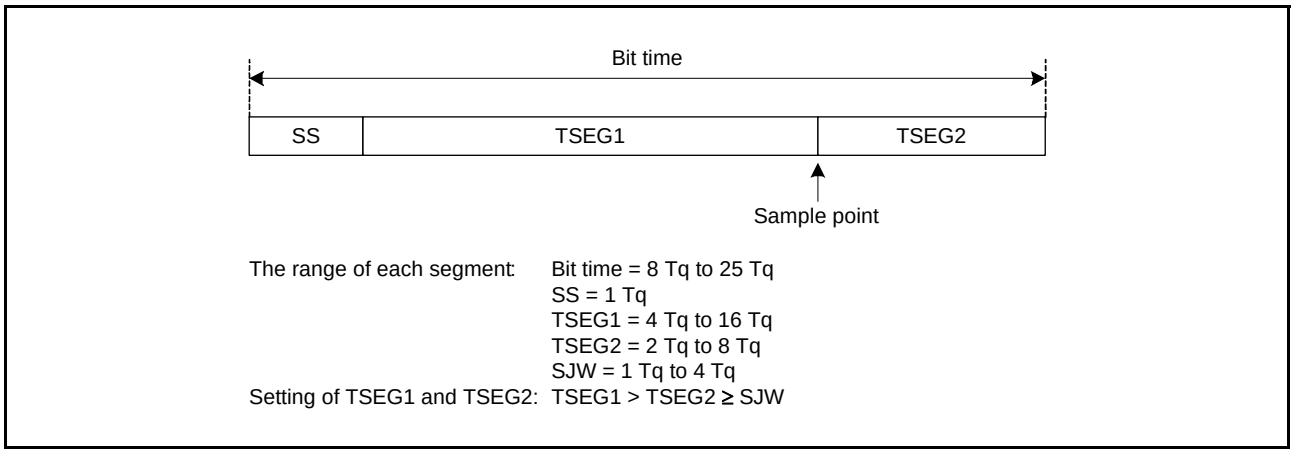


Figure 43.12 Bit Timing

43.4.3 Bit Rate

The bit rate depends on the division value of fCAN (CAN clock), the division value of the baud rate prescaler, and the number of Tq of 1 bit time.

$$\text{Bit rate [bps]} = \frac{\text{fCAN}}{\text{Baud rate prescaler division value}^{*1} \times \text{number of Tq of 1 bit time}} = \frac{\text{fCANCLK}}{\text{Number of Tq of 1 bit time}}$$

Note 1. Division value of baud rate prescaler = P + 1 (P: 0 to 1023)
P: Setting of the BRP[9:0] bits in BCR

Table 43.9 lists bit rate examples.

Table 43.9 Bit Rate Examples

fCAN	50 MHz		48 MHz		40 MHz		32 MHz	
Bit Rate	Number of Tq	P + 1	Number of Tq	P + 1	Number of Tq	P + 1	Number of Tq	P + 1
1 Mbps	10Tq 25Tq	5 2	8Tq 12Tq 16Tq	6 4 3	10Tq 20Tq	4 2	8Tq 16Tq	4 2
500 kbps	10Tq 25Tq	10 4	8Tq 12Tq 16Tq	12 8 6	10Tq 20Tq	8 4	8Tq 16Tq	8 4
250 kbps	10Tq 25Tq	20 8	8Tq 12Tq 16Tq	24 16 12	10Tq 20Tq	16 8	8Tq 16Tq	16 8
125 kbps	10Tq 25Tq	40 16	8Tq 12Tq 16Tq	48 32 24	10Tq 20Tq	32 16	8Tq 16Tq	32 16
83.3 kbps	10Tq 25Tq	60 24	8Tq 12Tq 16Tq	72 48 36	8Tq 10Tq 16Tq 20Tq	60 48 30 24	8Tq 16Tq	48 24
33.3 kbps	10Tq 25Tq	150 60	8Tq 12Tq 16Tq	180 120 90	8Tq 10Tq 20Tq	150 120 60	8Tq 10Tq 16Tq 20Tq	120 96 60 48

43.5 Mailbox and Mask Register Structure

Figure 43.13 shows the structure of MBj.
There are 32 mailboxes with the same structure.

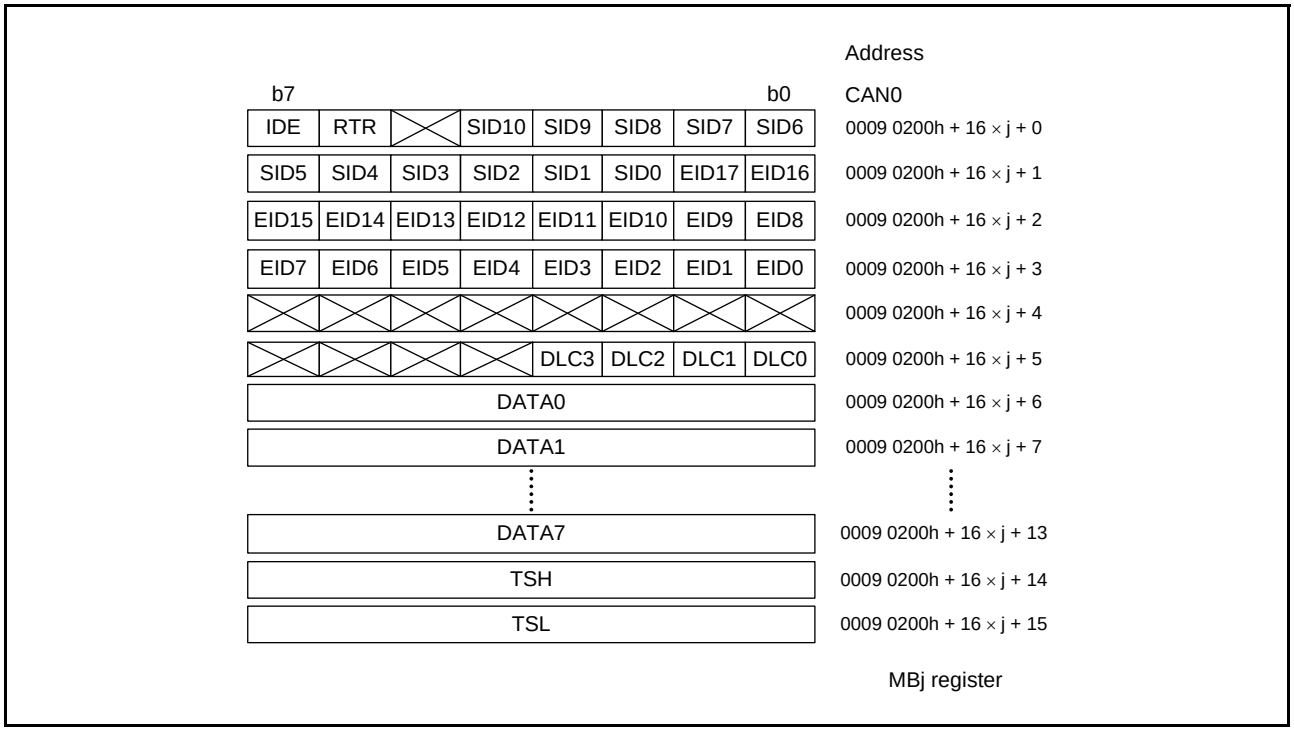


Figure 43.13 Structure of MBj (j = 0 to 31)

Figure 43.14 shows the structure of MKRk.
There are eight mask registers with the same structure.

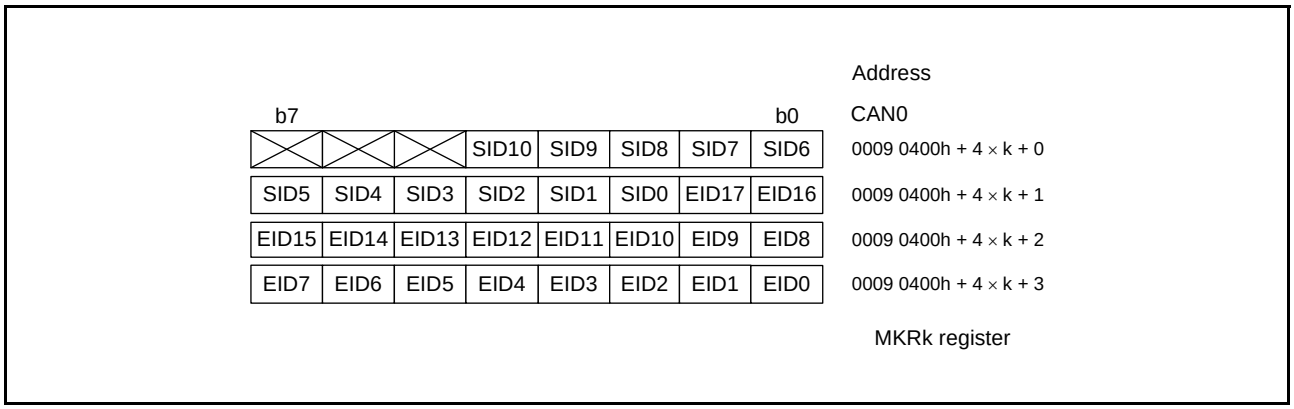


Figure 43.14 Structure of MKRk (k = 0 to 7)

Figure 43.15 shows the structure of FIDCR0 and FIDCR1.
There are two FIFO received ID compare registers with the same structure.

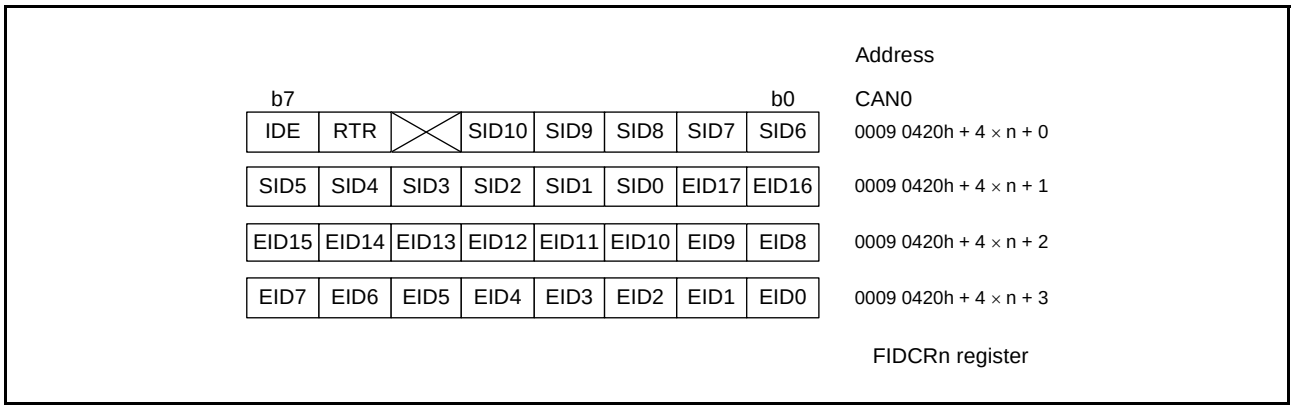


Figure 43.15 Structure of FIDCRn (n = 0, 1)

43.6 Acceptance Filtering and Masking Functions

The acceptance filtering function and masking function allows the user to select and receive messages with a specified range of multiple IDs for mailboxes.
Registers MKR0 to MKR7 can perform masking of the standard ID and the extended ID of 29 bits.

- MKR0 corresponds to mailboxes [0] to [3]
- MKR1 corresponds to mailboxes [4] to [7]
- MKR2 corresponds to mailboxes [8] to [11]
- MKR3 corresponds to mailboxes [12] to [15]
- MKR4 corresponds to mailboxes [16] to [19]
- MKR5 corresponds to mailboxes [20] to [23]
- MKR6 corresponds to mailboxes [24] to [27] in normal mailbox mode and the receive FIFO mailboxes [28] to [31] in FIFO mailbox mode.
- MKR7 corresponds to mailboxes [28] to [31] in normal mailbox mode and the receive FIFO mailboxes [28] to [31] in FIFO mailbox mode.

MKIVLR disables acceptance filtering individually for each mailbox.
The IDE bit in MBj is valid when the IDFM[1:0] bits in CTLR are 10b (mixed ID mode).
The RTR bit in MBj selects a data frame or a remote frame.

In FIFO mailbox mode, normal mailboxes (mailboxes [0] to [23]) use the single corresponding register among MKR0 to MKR5 for acceptance filtering. Receive FIFO mailboxes (mailboxes [28] to [31]) use two registers MKR6 and MKR7 for acceptance filtering.
Also, the receive FIFO uses two registers FIDCR0 and FIDCR1 for ID comparison. Bits EID[17:0], SID[10:0], RTR, and IDE in MB28 to MB31 for the receive FIFO are disabled. As acceptance filtering depends on the result of two logic AND operations, two ranges of IDs can be received into the receive FIFO.
MKIVLR is disabled for the receive FIFO.

If both the standard ID and extended ID are set in the IDE bits in FIDCR0 and FIDCR1 individually, both ID formats are received.
If both the data frame and remote frame are set in the RTR bits in FIDCR0 and FIDCR1 individually, both data and remote frames are received.
When combination with two ranges of IDs is not necessary, set the same mask value and the same ID into both the FIFO ID and mask register.
Figure 43.16 shows the correspondence between mask registers and mailboxes. Figure 43.17 shows acceptance filtering.

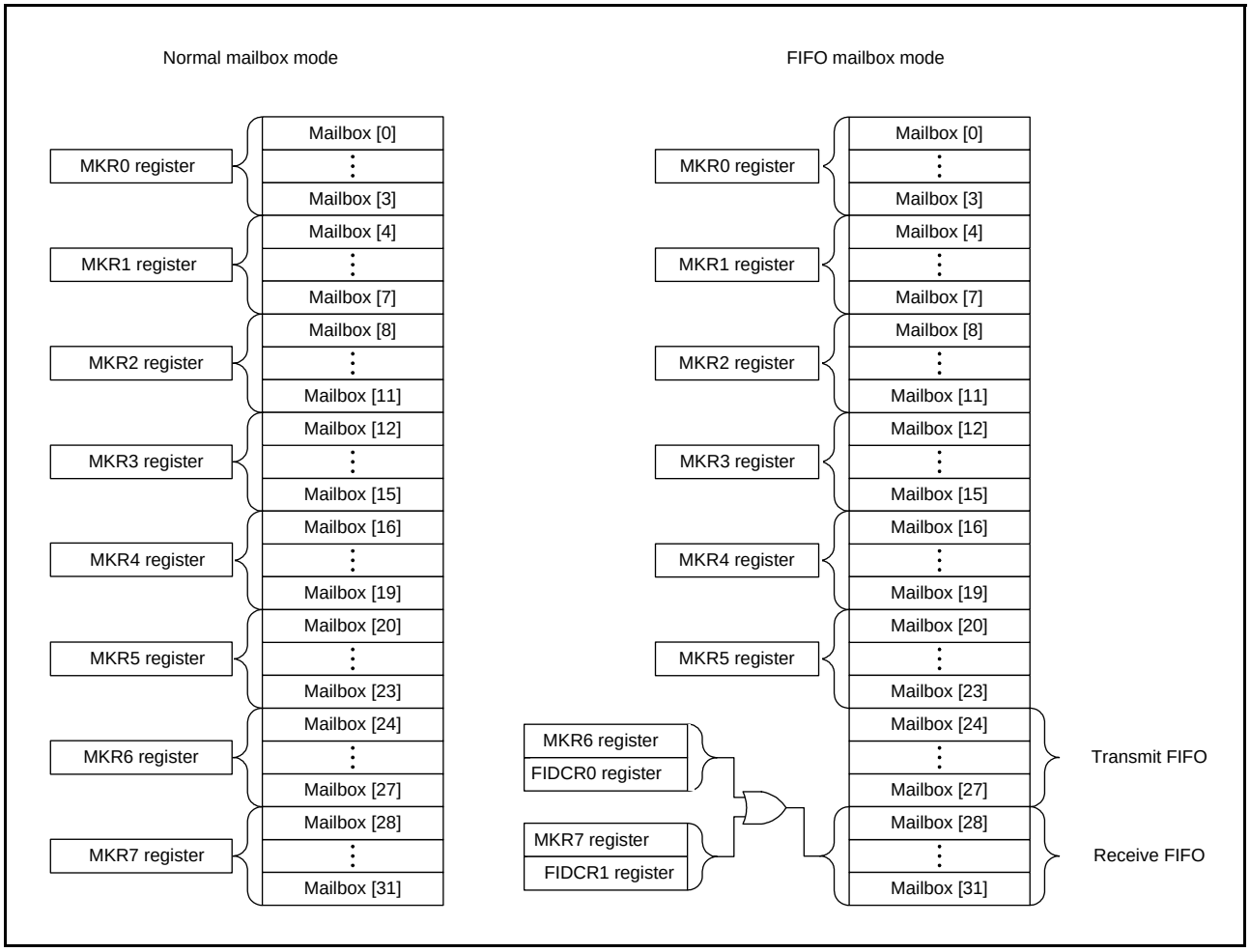
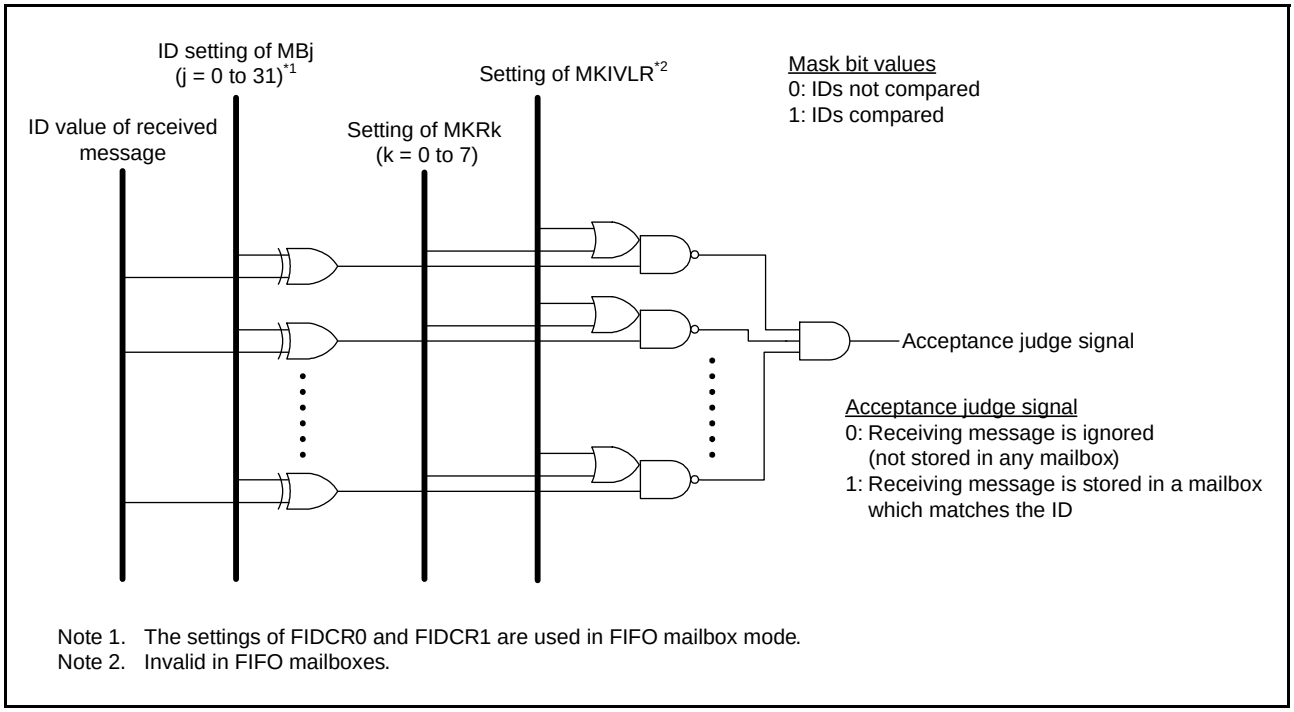


Figure 43.16 Correspondence between Mask Registers and Mailboxes



Note 1. The settings of FIDCR0 and FIDCR1 are used in FIFO mailbox mode.
Note 2. Invalid in FIFO mailboxes.

Figure 43.17 Acceptance Filtering

43.7 Reception and Transmission

Table 43.10 lists how to make the CAN communication mode settings.

Table 43.10 Setting of CAN Receive Mode and CAN Transmit Mode

MCTLj. TRMREQ	MCTLj. RECREQ	MCTLj. ONESHOT	Communication Mode of Mailbox
0	0	0	Mailbox disabled or transmission being aborted.
0	0	1	Can be configured only when transmission or reception from a mailbox programmed in one-shot mode is aborted.
0	1	0	Configured as a receive mailbox for a data frame or a remote frame.
0	1	1	Configured as a one-shot receive mailbox for a data frame or a remote frame.
1	0	0	Configured as a transmit mailbox for a data frame or a remote frame.
1	0	1	Configured as a one-shot transmit mailbox for a data frame or a remote frame.
1	1	0	Do not set.
1	1	1	Do not set.

j = 0 to 31

When a mailbox is configured as a receive mailbox or a one-shot receive mailbox, note the following:

1. Before a mailbox is configured as a receive mailbox or a one-shot receive mailbox, set MCTLj to 00h.
2. A received message is stored into the first mailbox that matches the condition according to the result of receive mode setting and acceptance filtering. Upon deciding the mailbox to store the received message, the mailbox with the smaller number has higher priority.
3. In CAN operation mode, when the CAN module transmits a message whose ID matches with the ID/mask set of a mailbox configured to receive messages, the CAN module never receives the transmitted data. In self-test mode, however, the CAN module will receive its transmitted data. In this case, the CAN module returns ACK.

When configuring a mailbox as a transmit mailbox or a one-shot transmit mailbox, note the following:

4. Before a mailbox is configured as a transmit mailbox or a one-shot transmit mailbox, ensure that MCTLj is 00h and that there is no pending abort process.

43.7.1 Reception

Figure 43.18 shows an operation example of data frame reception in overwrite mode.
This example shows the operation of overwriting the first message when the CAN module receives two consecutive CAN messages which match the receiving conditions of MCTLj (j = 0 to 31).

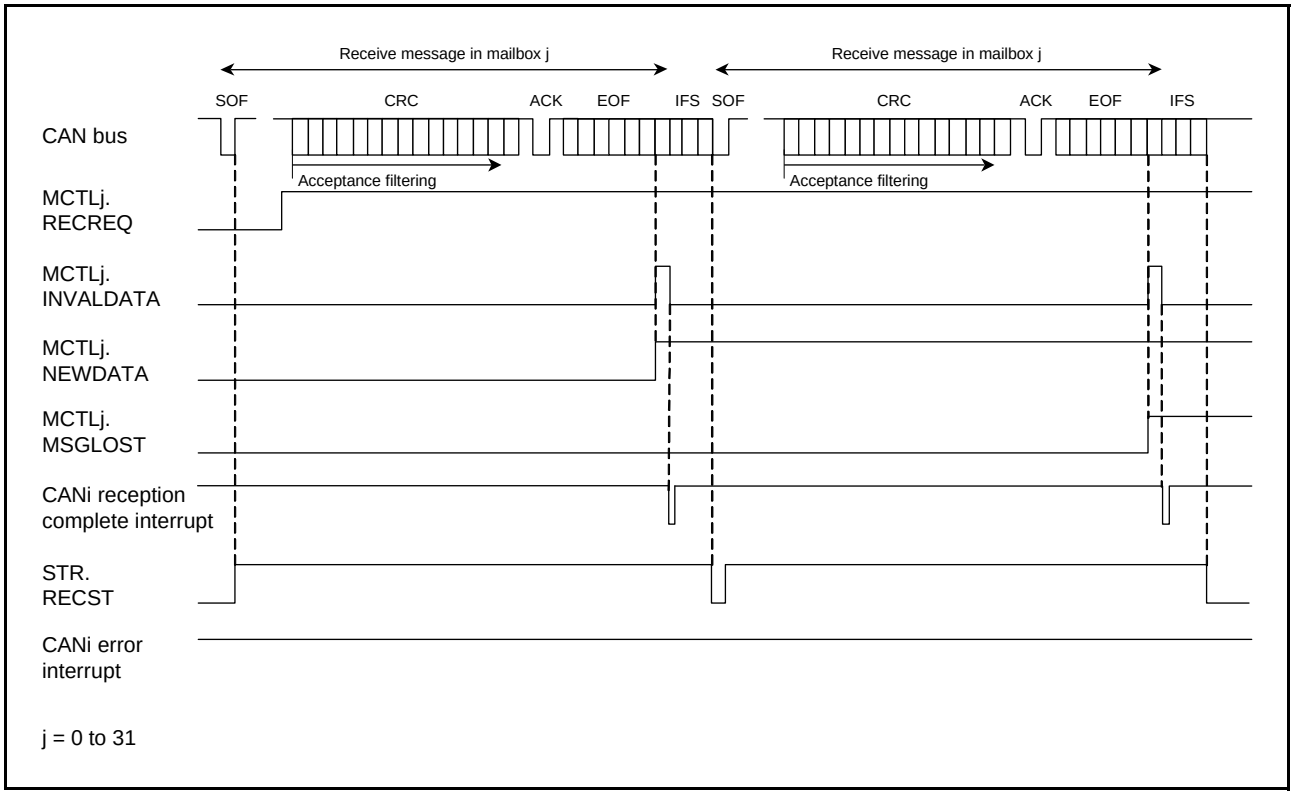


Figure 43.18 Operation Example of Data Frame Reception in Overwrite Mode

1. When an SOF is detected on the CAN bus, the RECST flag in STR is set to 1 (reception in progress) if the CAN module has no message ready to start transmission.
2. The acceptance filter processing starts at the beginning of the CRC field to select the receive mailbox.
3. After a message has been received, the NEWDATA flag in MCTLj for the receive mailbox is set to 1 (new message is being stored or has been stored to the mailbox). The INVALDATA flag in MCTLj is set to 1 (message is being updated) at the same time, and then the INVALDATA flag is set to 0 (message valid) again after the complete message is transferred to the mailbox.
4. When the interrupt enable bit in MIER for the receive mailbox is 1 (interrupt enabled), the CANi reception complete interrupt request is generated. This interrupt (CANi reception complete interrupt) is generated when the INVALDATA flag is set to 0.
5. After reading the message from the mailbox, the NEWDATA flag needs to be set to 0 by a program.
6. In overwrite mode, if the next CAN message has been received into a mailbox whose NEWDATA flag is still set to 1, the MSGLOST flag in MCTLj is set to 1 (message has been overwritten). The new received message is transferred to the mailbox. The CANi reception complete interrupt request is generated the same as in 4.

Figure 43.19 shows the operation example of data frame reception in overrun mode.
This example shows the operation of overrunning the second message when the CAN module receives two consecutive CAN messages which match the receiving conditions of MCTLj (j = 0 to 31).

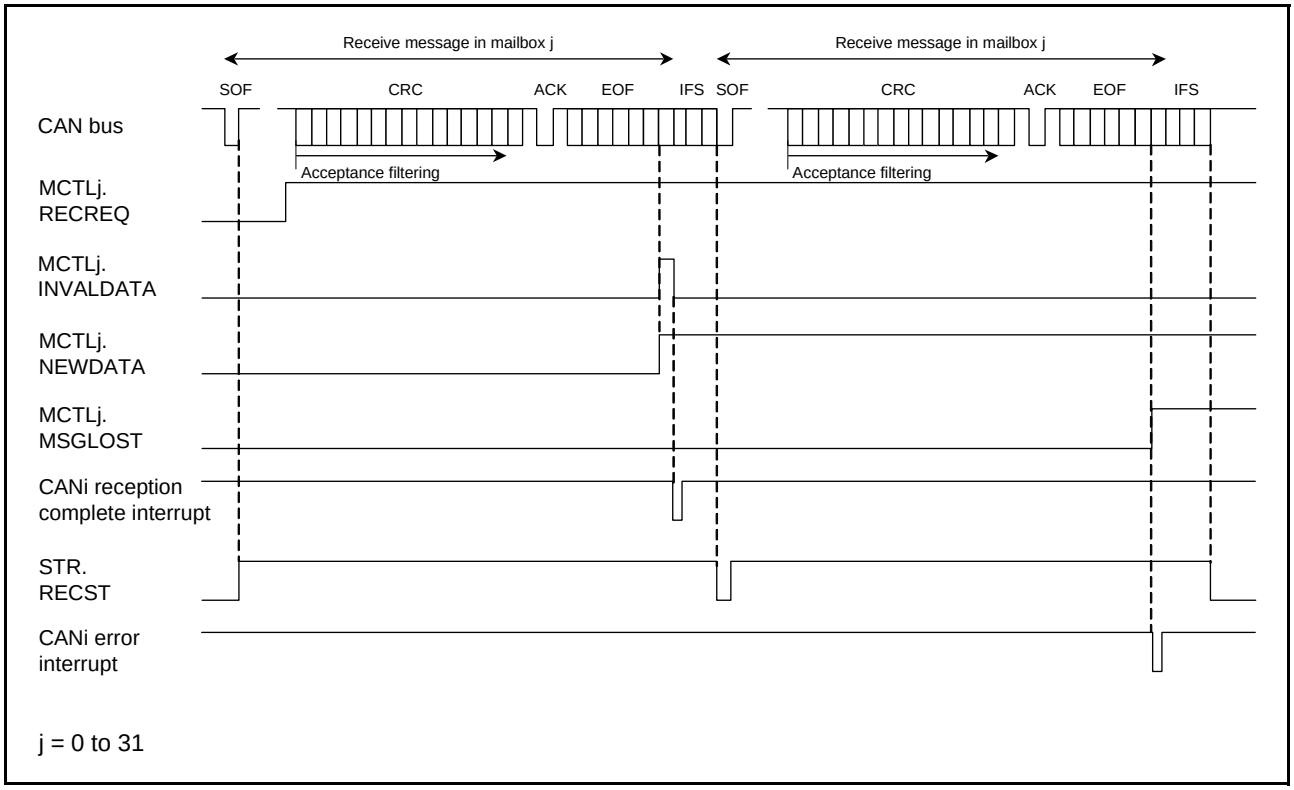


Figure 43.19 Operation Example of Data Frame Reception in Overrun Mode

1. to 5. are the same as in overwrite mode.

6. In overrun mode, if the next CAN message has been received before the NEWDATA flag in MCTLj is set to 0, the MSGLOST flag in MCTLj is set to 1 (message has been overrun). The new received message is discarded and a CANi error interrupt request is generated if the corresponding interrupt enable bit in EIER is set to 1 (interrupt enabled).

43.7.2 Transmission

Figure 43.20 shows an operation example of data frame transmission.

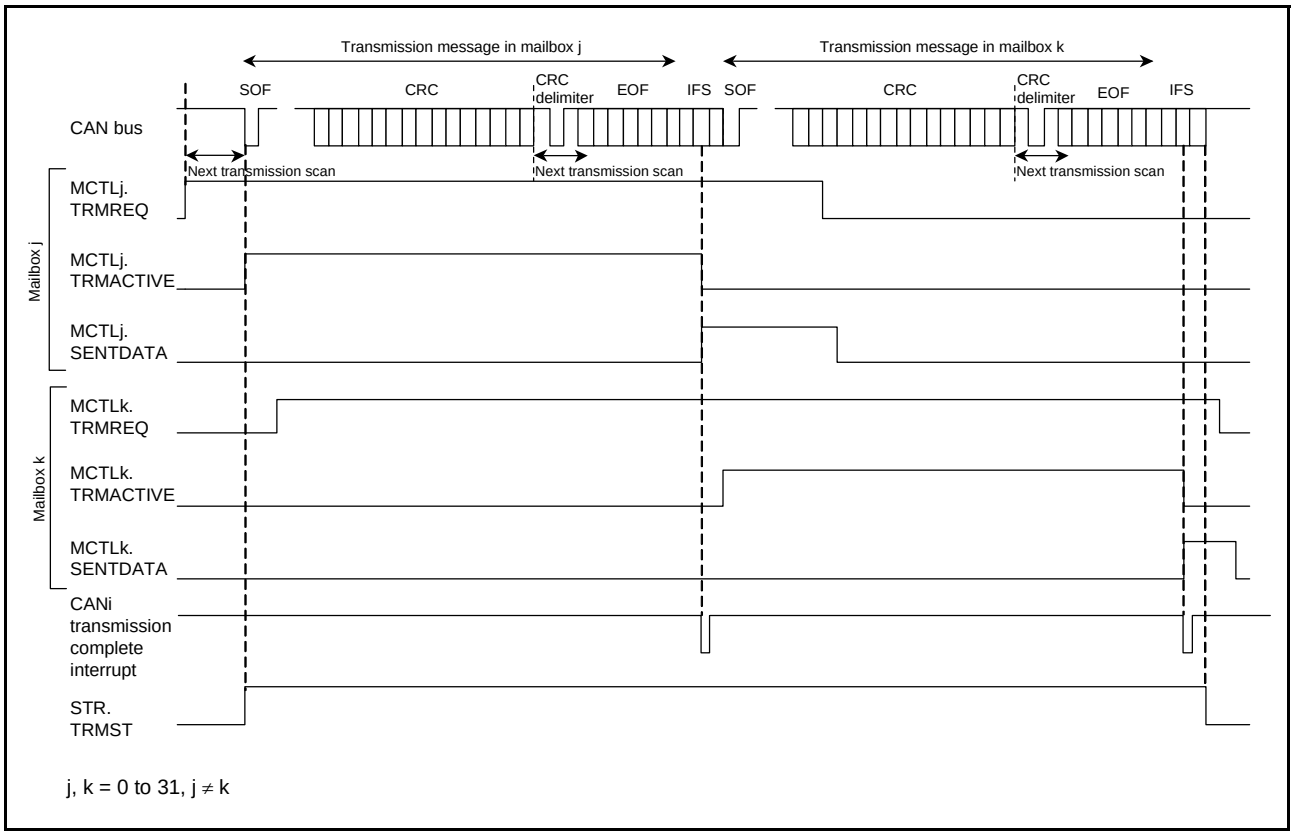


Figure 43.20 Operation Example of Data Frame Transmission

1. When a TRMREQ bit in MCTLj (j = 0 to 31) is set to 1 (transmit mailbox) in the bus-idle state, the mailbox scan processing starts to decide the highest-priority mailbox for transmission. Once the transmit mailbox is decided, the TRMACTIVE flag in MCTLj is set to 1 (from acceptance of transmission request to completion of transmission, or error/arbitration-lost), the TRMST flag in STR is set to 1 (transmission in progress), and the CAN module starts transmission.*1
2. If other TRMREQ bits are set, the transmission scan processing starts with the CRC delimiter for the next transmission.
3. If transmission is completed without losing arbitration, the SENTDATA flag in MCTLj is set to 1 (transmission completed) and the TRMACTIVE flag is set to 0 (transmission is pending or transmission is not requested). If the interrupt enable bit in MIER is 1 (interrupt enabled), the CANi transmission complete interrupt request is generated.
4. When requesting the next transmission from the same mailbox, set SENTDATA flag and TRMREQ bit to 0, then set the TRMREQ bit to 1 after checking that SENTDATA flag and TRMREQ bit have been set to 0.

Note 1. If arbitration is lost after the CAN module starts transmission, the TRMACTIVE flag is set to 0. The transmission scan processing is performed again to search for the highest-priority transmit mailbox from the beginning of the CRC delimiter. If an error occurs either during transmission or following the arbitration-lost, the transmission scan processing is performed again to search for the highest-priority transmit mailbox from the start of the error delimiter.

43.8 CAN Interrupt

The CAN module provides the following CAN interrupts for each channel. Table 43.11 lists CAN interrupts.

- CANi reception complete interrupt (mailboxes 0 to 31) [RXMi]
- CANi transmission complete interrupt (mailboxes 0 to 31) [TXMi]
- CANi receive FIFO interrupt [RXFi]
- CANi transmit FIFO interrupt [TXFi]
- CANi error interrupt [ERSi]

There are eight types of interrupt sources for the CANi error interrupts. These sources can be determined by checking EIFR.

- Bus error
- Error-warning
- Error-passive
- Bus-off entry
- Bus-off recovery
- Receive overrun
- Overload frame transmission
- Bus lock

Table 43.11 CAN Interrupts

Module	Interrupt Symbol	Interrupt Source	Source Flag
CANi	ERSi	Bus lock detected	EIFR.BLIF
		Overload frame transmission detected	EIFR.OLIF
		Overrun detected	EIFR.ORIF
		Bus-off recovery detected	EIFR.BORIF
		Bus-off entry detected	EIFR.BOEIF
		Error-passive detected	EIFR.EPIF
		Error-warning detected	EIFR.EWIF
		Bus error detected	EIFR.BEIF
	RXFi	Receive FIFO message received (MIER[29] = 0)	RFCR.RFUST[2:0]
		Receive FIFO warning (MIER[29] = 1)	
	TXFi	Transmit FIFO message transmission completed (MIER[25] = 0)	TFCR.TFUST[2:0]
		FIFO last message transmission completed (MIER[25] = 1)	
	RXMi	Mailbox 0 to 31 message received	MCTL0.NEWDATA to MCTL31.NEWDATA
	TXMi	Mailbox 0 to 31 message transmission completed	MCTL0.SENTDATA to MCTL31.SENTDATA

i = 0 to 2

43.9 Usage Notes

43.9.1 Setting for the Module-Stop State

Module stop control register B (MSTPCRB) can be used to enable or disable operation of the CAN module. The CAN module is stopped after a reset. The registers become accessible on release from the module-stop state. For details, refer to section 11, Low Power Consumption.

44. Serial Peripheral Interface (RSPIa)

In this section, “PCLK” is used to refer to PCLKA.

44.1 Overview

This MCU includes one channel of Serial Peripheral Interface (RSPI).
The RSPI channels are capable of high-speed, full-duplex synchronous serial communications with multiple processors and peripheral devices.
Table 44.1 lists the specifications of the RSPI, and Figure 44.1 shows a block diagram of the RSPI.
In this section, m as used with the RSPI command registers (SPCMDm) indicates 0 to 7.

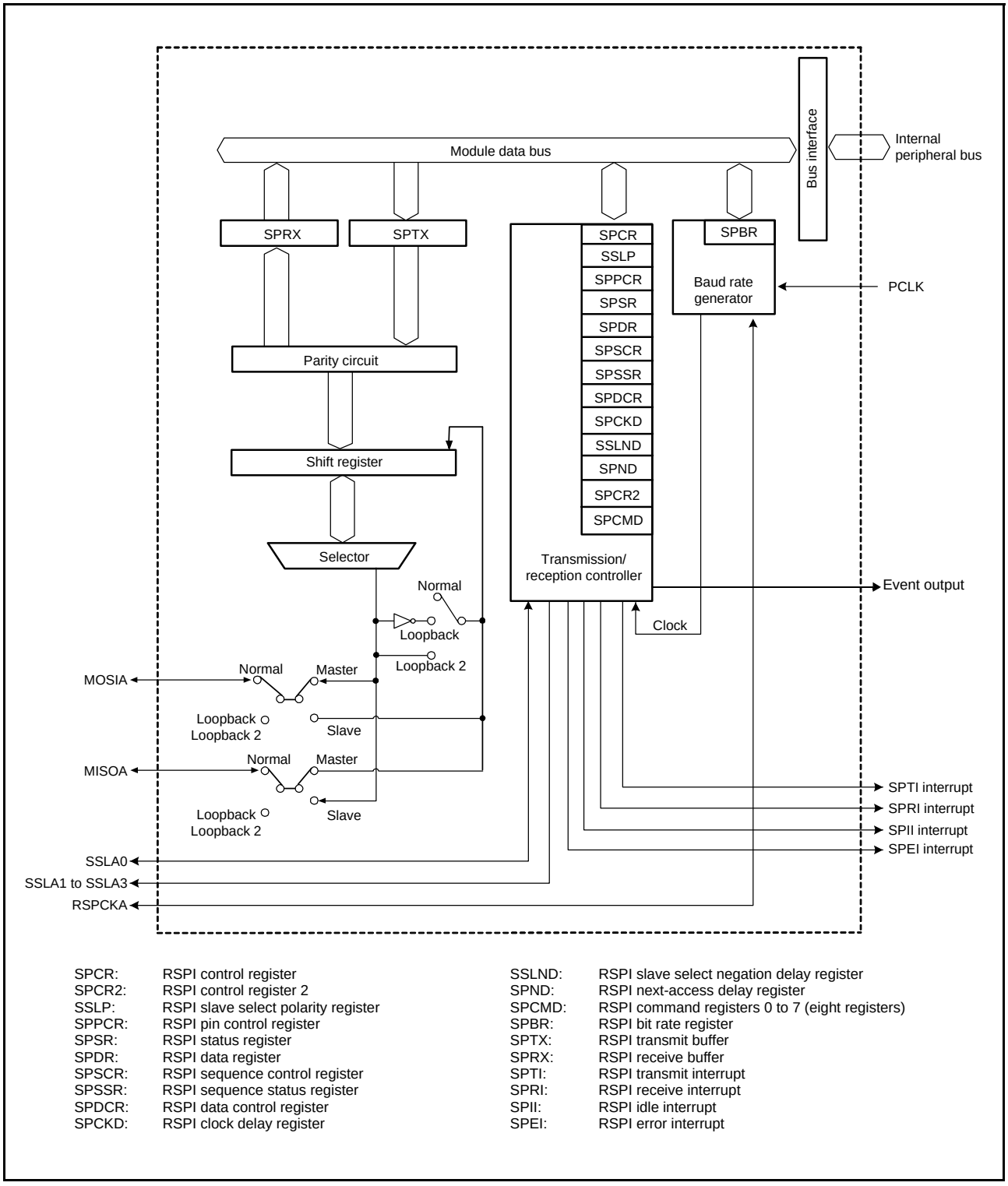
Table 44.1 RSPI Specifications (1/2)

Item	Description
Number of channels	One channel
RSPI transfer functions	• Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (RSPI clock) signals allows serial communications through SPI operation (4-wire method) or clock synchronous operation (3-wire method). • Transmit-only operation is available. • Communication mode: Full-duplex or transmit-only can be selected. • Switching of the polarity of RSPCK • Switching of the phase of RSPCK
Data format	• MSB first/LSB first selectable • Transfer bit length is selectable as 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, or 32 bits. • 128-bit transmit/receive buffers • Up to four frames can be transferred in one round of transmission/reception (each frame consisting of up to 32 bits).
Bit rate	• In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing PCLK (the division ratio ranges from divided by 2 to divided by 4096). • In slave mode, the minimum PCLK clock divided by 8 can be input as RSPCK (the maximum frequency of RSPCK is that of PCLK divided by 8). Width at high level: 4 cycles of PCLK; width at low level: 4 cycles of PCLK
Buffer configuration	• Double buffer configuration for the transmit/receive buffers • 128 bits for the transmit/receive buffers
Error detection	• Mode fault error detection • Overrun error detection*1 • Parity error detection
SSL control function	• Four SSL pins (SSLA0 to SSLA3) for each channel • In single-master mode, SSLA0 to SSLA3 pins are output. • In multi-master mode: SSLA0 pin for input, and SSLA1 to SSLA3 pins for either output or unused. • In slave mode: SSLA0 pin for input, and SSLA1 to SSLA3 pins for unused. • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Function for changing SSL polarity
Control in master transfer	• A transfer of up to eight commands can be executed sequentially in looped execution. • For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity/phase, transfer data length, MSB/LSB first, burst, RSPCK delay, SSL negation delay, and next-access delay • A transfer can be initiated by writing to the transmit buffer. • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function

Table 44.1 RSPI Specifications (2/2)

Item	Description
Interrupt sources	• Interrupt sources Receive buffer full interrupt Transmit buffer empty interrupt RSPI error interrupt (mode fault, overrun, or parity error) RSPI idle interrupt (RSPI idle)
Event link function (output)	• The following events can be output to the event link controller. (RSPI0) Receive buffer full signal Transmit buffer empty signal Mode fault, overrun, or parity error signal RSPI idle signal Transmission-completed signal
Others	• Function for switching between CMOS output and open-drain output • Function for initializing the RSPI • Loopback mode
Low power consumption function	Module stop state can be set.

Note 1. In master reception and when the RSPCK auto-stop function is enabled, an overrun error does not occur because the transfer clock is stopped at the timing of overrun error detection.



SPCR:	RSPI control register	SSLND:	RSPI slave select negation delay register
SPCR2:	RSPI control register 2	SPND:	RSPI next-access delay register
SSLP:	RSPI slave select polarity register	SPCMD:	RSPI command registers 0 to 7 (eight registers)
SPPCR:	RSPI pin control register	SPBR:	RSPI bit rate register
SPSR:	RSPI status register	SPTX:	RSPI transmit buffer
SPDR:	RSPI data register	SPRX:	RSPI receive buffer
SPSCR:	RSPI sequence control register	SPTI:	RSPI transmit interrupt
SPSSR:	RSPI sequence status register	SPRI:	RSPI receive interrupt
SPDCR:	RSPI data control register	SPII:	RSPI idle interrupt
SPCKD:	RSPI clock delay register	SPEI:	RSPI error interrupt

Figure 44.1 RSPI Block Diagram

Table 44.2 lists the I/O pins used in the RSPI.
The RSPI automatically switches the I/O direction of the SSLA0 pin. SSLA0 is set as an output when the RSPI is a single master and as an input when the RSPI is a multi-master or a slave. Pins RSPCKA, MOSIA, and MISOA are automatically set as inputs or outputs according to the setting of master or slave and the level input on the SSLA0 pin. Refer to section 44.3.2, Controlling RSPI Pins for details.

Table 44.2 RSPI Pin Configuration

Channel	Pin Name	I/O	Function
RSPI0	RSPCKA	I/O	Clock I/O
	MOSIA	I/O	Master transmit data I/O
	MISOA	I/O	Slave transmit data I/O
	SSLA0	I/O	Slave selection I/O
	SSLA1	Output	Slave selection output
	SSLA2	Output	Slave selection output
	SSLA3	Output	Slave selection output

44.2 Register Descriptions

44.2.1 RSPI Control Register (SPCR)

Address(es): RSPI0.SPCR 000D 0100h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	TXMD	SPMS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPMS	RSPI Mode Select	0: SPI operation (4-wire method) 1: Clock synchronous operation (3-wire method)	R/W
b1	TXMD	Communications Operating Mode Select	0: Full-duplex synchronous serial communications 1: Serial communications consisting of only transmit operations	R/W
b2	MODFEN	Mode Fault Error Detection Enable	0: Disables the detection of mode fault error 1: Enables the detection of mode fault error	R/W
b3	MSTR	RSPI Master/Slave Mode Select	0: Slave mode 1: Master mode	R/W
b4	SPEIE	RSPI Error Interrupt Enable	0: Disables the generation of RSPI error interrupt requests 1: Enables the generation of RSPI error interrupt requests	R/W
b5	SPTIE	Transmit Buffer Empty Interrupt Enable	0: Disables the generation of transmit buffer empty interrupt requests 1: Enables the generation of transmit buffer empty interrupt requests	R/W
b6	SPE	RSPI Function Enable	0: Disables the RSPI function 1: Enables the RSPI function	R/W
b7	SPRIE	RSPI Receive Buffer Full Interrupt Enable	0: Disables the generation of RSPI receive buffer full interrupt requests 1: Enables the generation of RSPI receive buffer full interrupt requests	R/W

If the SPCR.MSTR, SPCR.MODFEN, or SPCR.TXMD bit is changed while the SPCR.SPE bit is 1, subsequent operations should not be performed.

SPMS Bit (RSPI Mode Select)

The SPMS bit selects SPI operation (4-wire method) or clock synchronous operation (3-wire method).
The SSLA0 to SSLA3 pins are not used in clock synchronous operation. The RSPCKA, MOSIA, and MISOA pins handle communications. If clock synchronous operation is to proceed in master mode (SPCR.MSTR = 1), the SPCMDm.CPHA bit can be set to either 0 or 1. Set the CPHA bit to 1 if clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0). Operation should not be performed if the CPHA bit is set to 0 when clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0).

TXMD Bit (Communications Operating Mode Select)

The TXMD bit selects full-duplex synchronous serial communications or transmit operations only.
When performing communications with the TXMD bit set to 1, the RSPI performs only transmit operations and not receive operations (refer to section 44.3.6, Communications Operating Mode).
When the TXMD bit is set to 1, receive buffer full interrupt requests cannot be used.

MODFEN Bit (Mode Fault Error Detection Enable)

The MODFEN bit enables or disables the detection of mode fault error (refer to section 44.3.8, Error Detection). In addition, the RSPI determines the I/O direction of the SSLA0 to SSLA3 pins based on combinations of the MODFEN and MSTR bits (refer to section 44.3.2, Controlling RSPI Pins).

MSTR Bit (RSPI Master/Slave Mode Select)

The MSTR bit selects master/slave mode of the RSPI. According to MSTR bit settings, the RSPI determines the direction of pins RSPCKA, MOSIA, MISOA, and SSLA0 to SSLA3.

SPEIE Bit (RSPI Error Interrupt Enable)

The SPEIE bit enables or disables the generation of RSPI error interrupt requests when the RSPI detects a mode fault error and sets the SPSR.MODF flag to 1, when the RSPI detects an overrun error and sets the SPSR.OVRF flag to 1, or when the RSPI detects a parity error and sets the SPSR.PERF flag to 1 (refer to section 44.3.8, Error Detection).

SPTIE Bit (Transmit Buffer Empty Interrupt Enable)

The SPTIE bit enables or disables the generation of transmit buffer empty interrupt requests when the RSPI detects when the transmit buffer is empty.

A transmit buffer empty interrupt request when transmission starts is generated by setting the SPE and SPTIE bits to 1 at the same time or by setting the SPE bit to 1 after setting the SPTIE bit to 1.

Note that a transmit buffer interrupt is generated when the SPTIE bit is 1 even if the RSPI function is disabled (the SPTIE bit is changed to 0).

SPE Bit (RSPI Function Enable)

The SPE bit enables or disables the RSPI function.

When the SPSR.MODF flag is 1, the SPE bit cannot be set to 1. For details, refer to section 44.3.8, Error Detection.

Setting the SPE bit to 0 disables the RSPI function, and initializes a part of the module function. For details, refer to section 44.3.9, Initializing RSPI. Furthermore, a transmit buffer empty interrupt request is generated by the state of the SPE bit changing from 0 to 1 or from 1 to 0.

SPRIE Bit (RSPI Receive Buffer Full Interrupt Enable)

If the RSPI has detected a receive buffer full write after completion of a serial transfer, the SPRIE bit enables or disables the generation of an RSPI receive buffer full interrupt request.

44.2.2 RSPI Slave Select Polarity Register (SSLP)

Address(es): RSPI0.SSLP 000D 0101h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SSL0P	SSL0 Signal Polarity Setting	0: SSL0 signal is active low 1: SSL0 signal is active high	R/W
b1	SSL1P	SSL1 Signal Polarity Setting	0: SSL1 signal is active low 1: SSL1 signal is active high	R/W
b2	SSL2P	SSL2 Signal Polarity Setting	0: SSL2 signal is active low 1: SSL2 signal is active high	R/W
b3	SSL3P	SSL3 Signal Polarity Setting	0: SSL3 signal is active low 1: SSL3 signal is active high	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the contents of SSLP are changed while the SPCR.SPE bit is 1, subsequent operations should not be performed.

44.2.3 RSPI Pin Control Register (SPPCR)

Address(es): RSPI0.SPPCR 000D 0102h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPLP	RSPI Loopback	0: Normal mode 1: Loopback mode (data is inverted for transmission)	R/W
b1	SPLP2	RSPI Loopback 2	0: Normal mode 1: Loopback mode (data is not inverted for transmission)	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	MOIFV	MOSI Idle Fixed Value	0: The level output on the MOSIA pin during MOSI idling corresponds to low 1: The level output on the MOSIA pin during MOSI idling corresponds to high	R/W
b5	MOIFE	MOSI Idle Value Fixing Enable	0: MOSI output value equals final data from previous transfer 1: MOSI output value equals the value set in the MOIFV bit	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the contents of SPPCR are changed while the SPCR.SPE bit is 1, subsequent operations should not be performed.

SPLP Bit (RSPI Loopback)

The SPLP bit selects the mode of the RSPI pins.
When the SPLP bit is set to 1, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects (inverts) the input path and output path for the shift register (loopback mode).

SPLP2 Bit (RSPI Loopback 2)

The SPLP2 bit selects the mode of the RSPI pins.
When the SPLP2 bit is set to 1, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path for the shift register (loopback mode).

MOIFV Bit (MOSI Idle Fixed Value)

If the MOIFE bit is 1 in master mode, the MOIFV bit determines the MOSIA pin output value during the SSL negation period (including the SSL retention period during a burst transfer).

MOIFE Bit (MOSI Idle Value Fixing Enable)

The MOIFE bit fixes the MOSIA output value when the RSPI in master mode is in an SSL negation period (including the SSL retention period during a burst transfer). When the MOIFE bit is 0, the RSPI outputs the last data from the previous serial transfer during the SSL negation period to the MOSIA pin. When the MOIFE bit is 1, the RSPI outputs the fixed value set in the MOIFV bit to the MOSIA pin.

44.2.4 RSPI Status Register (SPSR)

Address(es): RSPI0.SPSR 000D 0103h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRF	—	SPTEF	—	PERF	MODF	IDLNF	OVRF
Value after reset:	0	0	1	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OVRF	Overrun Error Flag	0: No overrun error occurs 1: An overrun error occurs	R/(W) *1
b1	IDLNF	RSPI Idle Flag	0: RSPI is in the idle state 1: RSPI is in the transfer state	R
b2	MODF	Mode Fault Error Flag	0: No mode fault error occurs 1: A mode fault error occurs	R/(W) *1
b3	PERF	Parity Error Flag	0: No parity error occurs 1: A parity error occurs	R/(W) *1
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	SPTEF	Transmit Buffer Empty Flag	0: Transmit buffer has valid data 1: Transmit buffer has no valid data	R/(W) *2
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	SPRF	Receive Buffer Full Flag	0: Receive buffer has no valid data 1: Receive buffer has valid data	R/(W) *2

Note 1. Only 0 can be written to clear the flag after reading 1.
Note 2. The write value should be 1.

OVRF Flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error. In master mode (when the SPCR.MSTR bit is 1) and when the RSPCK clock auto-stop function is enabled (the SPCR2.SCKASE bit is 1), an overrun error does not occur; accordingly this flag does not become 1. For details, refer to section 44.3.8.1, Overrun Error.

[Setting condition]

- When the next serial transfer ends while the SPCR.TXMD bit is 0 and the receive buffer is full.

[Clearing condition]

- When SPSR is read while the OVRF flag is 1, and then 0 is written to the OVRF flag.

IDLNF Flag (RSPI Idle Flag)

The IDLNF flag indicates the transfer status of the RSPI.

[Setting condition]

Master mode

- Condition 1 and condition 2 are not satisfied in master mode under the [Clearing condition] below.

Slave mode

- The SPCR.SPE bit is 1 (enables the RSPI function)

[Clearing condition]

Master mode

- The following 1 is satisfied (condition 1) or all of the following 2 to 4 are satisfied (condition 2).

1. The SPCR.SPE bit is 0 (disables the RSPI function)
2. The transmit buffer (SPTX) is empty (data for the next transfer is not set)
3. The SPSSR.SPCP[2:0] bits are 000b (beginning of sequence control)

4. The RSPI internal sequencer has entered the idle state (status in which operations up to the next-access delay have finished)

Slave mode

- The SPCR.SPE bit is 0 (disables the RSPI function)

MODF Flag (Mode Fault Error Flag)

Indicates the occurrence of a mode fault error.

[Setting condition]

Multi-master mode

- When the input level of the SSLAi pin changes to the active level while the SPCR.MSTR bit is 1 (master mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPI detects a mode fault error

Slave mode

- When the SSLAi pin is negated before the RSPCK cycle necessary for data transfer ends while the SPCR.MSTR bit is 0 (slave mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPI detects a mode fault error

The active level of the SSLAi signal is determined by the SSLP.SSLiP bit (SSLi signal polarity setting bit).

[Clearing condition]

- When SPSR is read while the MODF flag is 1, and then 0 is written to the MODF flag

PERF Flag (Parity Error Flag)

Indicates the occurrence of a parity error.

[Setting condition]

- When a serial transfer ends while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1, the RSPI detects a parity error

[Clearing condition]

- When SPSR is read while the PERF flag is 1, and then 0 is written to the PERF flag

SPTEF Flag (Transmit Buffer Empty Flag)

Indicates whether the transmit buffer (SPTX) in the RSPI data register has valid data.

[Setting condition]

- When the SPCR.SPE bit is 0 (disables the RSPI function)
- When data is transferred from the transmit buffer to the shift register

[Clearing condition]

- When the number of frames of transmit data specified by the SPDCR.SPFC[1:0] bits is written to the SPDR register

The SPDR register can be set only when the SPTEF flag is 1. The data in the transmit buffer is not updated when the SPDR register is set while the SPTEF flag is 0.

SPRF Flag (Receive Buffer Full Flag)

Indicates whether the receive buffer (SPRX) in the RSPI data register has valid data.

[Setting condition]

- When the number of frames of receive data specified by the SPDCR.SPFC[1:0] bits is transferred from shift register to the receive buffer (SPRX) while the SPCR.TXMD bit is 0 (full duplex) and the SPRF flag is 0.
  Note that the SPRF flag does not become 1 when the OVRF flag is 1.

[Clearing condition]

- When all of the received data are read from the SPDR register

44.2.5 RSPI Data Register (SPDR)

- When accessing in longword size

Address(es): RSPI0.SPDR 000D 0104h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

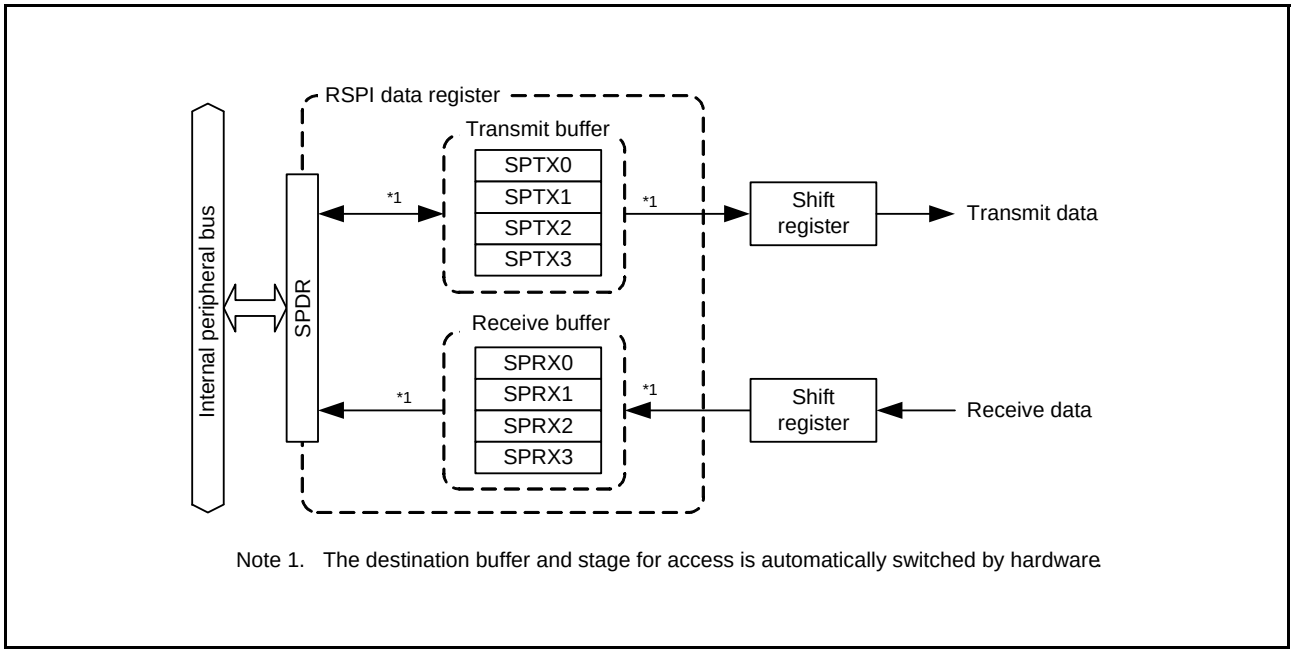
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When accessing in word size

Address(es): RSPI0.SPDR.H 000D 0104h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	H															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDR is the interface with the buffers that hold data for transmission and reception by the RSPI.
When accessing in longwords (the SPLW bit is 1), access SPDR in 32-bit units.
When accessing in words (the SPLW bit is 0), access SPDR.H in 16-bit units.
The transmit buffer (SPTX) and receive buffer (SPRX) are independent but are both mapped to SPDR. Figure 44.2 shows the Configuration of SPDR.



Note 1. The destination buffer and stage for access is automatically switched by hardware.

Figure 44.2 Configuration of SPDR

The transmit and receive buffers each have four stages. The number of stages to be used is selectable by the number of frames specification bits in the RSPI data control register (SPDCR.SPFC[1:0]). The eight stages of the buffer are all

mapped to the single address of SPDR.
Data written to SPDR are written to a transmit-buffer stage (SPTXn) (n = 0 to 3) and then transmitted from the buffer. The receive buffer holds received data on completion of reception. The receive buffer is not updated if an overrun is generated.
Furthermore, if the data length is other than 32 bits, bits not referred to in SPTXn (n = 0 to 3) are stored in the corresponding bits in SPRXn. For example, if the data length is 9 bits, received data are stored in the SPRXn[8:0] bits and the SPTXn[31:9] bits are stored in the SPRXn[31:9] bits.

(1) Bus Interface

SPDR is the interface with 32-bit wide transmit and receive buffers, each of which has four stages, for a total of 32 bytes. In other words, the 32 bytes are mapped to the 4-byte address space for SPDR. Furthermore, the unit of access for SPDR is selected by the SPDCR.SPLW bit.
Data for transmission should be flush with the LSB end of the register. Received data are stored flush with the LSB end. Operations involved in writing to and reading from SPDR are described below.

(a) Writing

Data written to SPDR are written to a transmit buffer (SPTXn). This is not influenced by the value of the SPDCR.SPRDTD bit unlike when reading from SPDR.
The transmit buffer includes a transmit buffer write pointer which is automatically updated to indicate the next stage each time data are written to SPDR.
Figure 44.3 shows the configuration of the bus interface with the transmit buffer in the case of writing to SPDR.

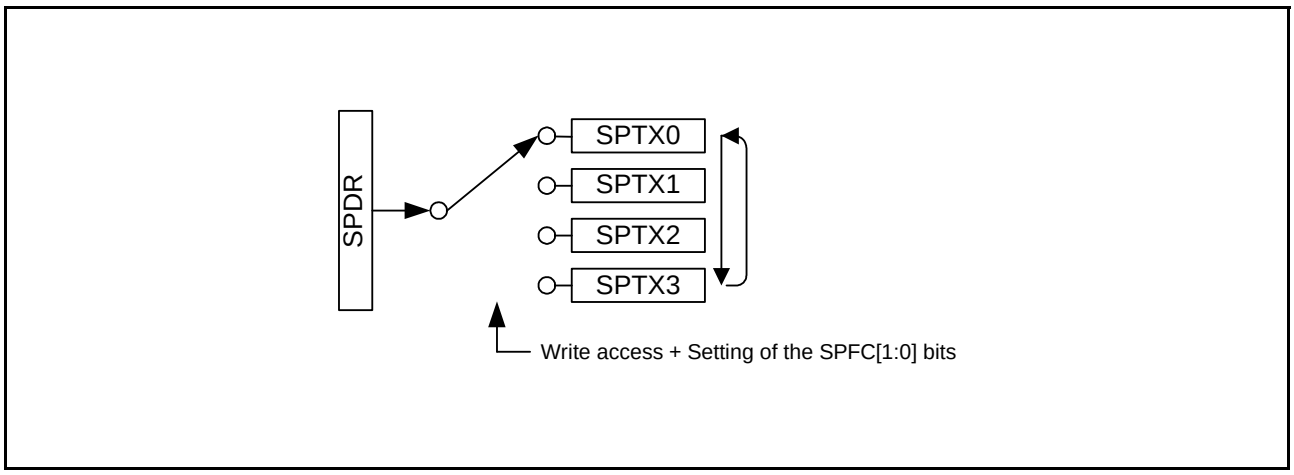


Figure 44.3 Configuration of SPDR (Writing)

The sequence for switching the transmit buffer write pointer differs with the setting of the number of frames specification bits in the RSPI data control register (SPDCR.SPFC[1:0]).

- Settings of the SPFC[1:0] bits and sequence of switching the pointer among SPTX0 to SPTX3.
  When the SPFC[1:0] bits are 00b: SPTX0 → SPTX0 → SPTX0 → …
  When the SPFC[1:0] bits are 01b: SPTX0 → SPTX1 → SPTX0 → SPTX1 → …
  When the SPFC[1:0] bits are 10b: SPTX0 → SPTX1 → SPTX2 → SPTX0 → SPTX1 → …
  When the SPFC[1:0] bits are 11b: SPTX0 → SPTX1 → SPTX2 → SPTX3 → SPTX0 → SPTX1 → …

When 1 is written to the RSPI function enable bit in the RSPI control register (SPCR.SPE) while the bit's current value is 0, SPTX0 will be the destination the next time writing proceeds.
When writing to the transmit buffer (SPTXn) after generation of the transmit buffer empty interrupt (after the SPSR.SPTEF flag becomes 1), write the number of frames set by the number of frames specification bits (SPFC[1:0]) in

the RSPI data control register (SPDCR). Even if the number of frames is written to the transmit buffer (SPTXn), the value of the buffer is not updated after completion of the writing and before generation of the next transmit buffer empty interrupt (while the SPSR.SPTEF flag is 0).

(b) Reading

SPDR can be read to read the value of a receive buffer (SPRXn) or a transmit buffer (SPTXn). The setting of the RSPI receive/transmit data select bit in the RSPI data control register (SPDCR.SPRDTD) selects whether reading is of the receive or transmit buffer.

The sequence of reading the SPDR register is controlled by independent pointers, receive buffer read pointer and transmit buffer read pointer.

Figure 44.4 shows the configuration of the bus interface with the receive and transmit buffers in the case of reading from SPDR.

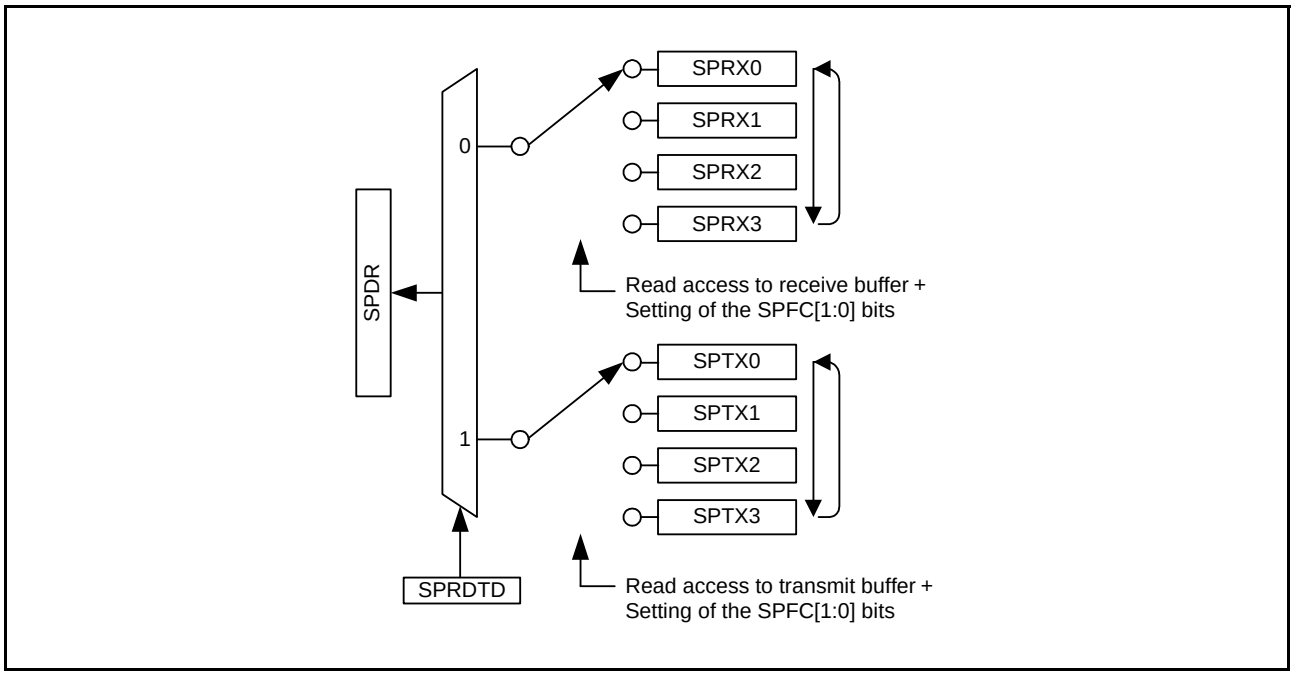


Figure 44.4 Configuration of SPDR (Reading)

Reading the receive buffer switches the receive buffer read pointer to the next buffer automatically.

The sequence of switching the receive buffer read pointer is the same as that for the transmit buffer write pointer. However, when 1 is written to the RSPI function enable bit in the RSPI control register (SPCR.SPE) while the bit's current value is 0, SPRX0 will be indicated by the buffer read pointer the next time reading proceeds.

The transmit buffer read pointer is updated when writing to SPDR, and not updated when reading from the transmit buffer. When reading from the transmit buffer, the value most recently written to SPDR is read. However, after generation of the transmit buffer empty interrupt, the values read from the transmit buffer are all 0 in the interval after completion of writing the number of frames of data specified in the number of frames specification bits (SPDCR.SPFC[1:0]) and before generation of the next buffer empty interrupt (while the SPSR.SPTEF flag is 0).

44.2.6 RSPI Sequence Control Register (SPSCR)

Address(es): RSPI0.SPSCR 000D 0108h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SPSLN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPSLN[2:0]	RSPI Sequence Length Specification	b2 b0 Sequence Length — Referenced SPCMD0 to SPCMD7 (No.) 0 0 0: 1 — 0→0→… 0 0 1: 2 — 0→1→0→… 0 1 0: 3 — 0→1→2→0→… 0 1 1: 4 — 0→1→2→3→0→… 1 0 0: 5 — 0→1→2→3→4→0→… 1 0 1: 6 — 0→1→2→3→4→5→0→… 1 1 0: 7 — 0→1→2→3→4→5→6→0→… 1 1 1: 8 — 0→1→2→3→4→5→6→7→0→… The order in which the SPCMD0 to SPCMD7 registers are to be referenced is changed according to the sequence length that is set in these bits. The relationship among the setting of these bits, sequence length, and SPCMD0 to SPCMD7 registers referenced by the RSPI is shown above. However, the RSPI in slave mode references SPCMD0.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SPSCR sets the sequence length when the RSPI operates in master mode. When changing the SPSCR.SPSLN[2:0] bits while both the SPCR.MSTR and SPCR.SPE bits are 1, the bits should be changed while the SPSR.IDLNF flag is 0.

SPSLN[2:0] Bits (RSPI Sequence Length Specification)

The SPSLN[2:0] bits specify a sequence length when the RSPI in master mode performs sequential operations. The RSPI in master mode changes SPCMD0 to SPCMD7 registers to be referenced and the order in which they are referenced according to the sequence length that is set in the SPSLN[2:0] bits.
In slave mode, SPCMD0 is referred.

44.2.7 RSPI Sequence Status Register (SPSSR)

Address(es): RSPI0.SPSSR 000D 0109h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SPECM[2:0]			—	SPCP[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPCP[2:0]	RSPI Command Pointer	b2 b0 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b3	—	Reserved	This bit is read as 0.	R
b6 to b4	SPECM[2:0]	RSPI Error Command	b6 b4 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b7	—	Reserved	This bit is read as 0.	R

SPSSR indicates the sequence control status when the RSPI operates in master mode.
Any writing to SPSSR is ignored.

SPCP[2:0] Bits (RSPI Command Pointer)

The SPCP[2:0] bits indicate SPCMDm that is currently pointed to by the pointer during sequence control by the RSPI. For the RSPI's sequence control, refer to section 44.3.10.1, Master Mode Operation.

SPECM[2:0] Bits (RSPI Error Command)

The SPECM[2:0] bits indicate SPCMDm that is specified by the SPCP[2:0] bits when an error is detected during sequence control by the RSPI. The RSPI updates the SPECM[2:0] bits only when an error is detected. If both the SPSR.OVRF and SPSR.MODF flags are 0 and there is no error, the values of the SPECM[2:0] bits have no meaning. For the RSPI's error detection function, refer to section 44.3.8, Error Detection. For the RSPI's sequence control, refer to section 44.3.10.1, Master Mode Operation.

44.2.8 RSPI Bit Rate Register (SPBR)

Address(es): RSPI0.SPBR 000D 010Ah

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

SPBR sets the bit rate in master mode. If the contents of SPBR are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations should not be performed.

When the RSPI is used in slave mode, the bit rate depends on the bit rate of the input clock (bit rate satisfying the electrical characteristics should be used) regardless of the settings of SPBR and the SPCMDm.BRDV[1:0] bits (bit rate division setting bits).
The bit rate is determined by combinations of the SPBR setting and the SPCMDm.BRDV[1:0] bit setting. The equation for calculating the bit rate is given below. In the equation, n denotes an SPBR setting (0, 1, 2, …, 255), and N denotes a BRDV[1:0] bit setting (0, 1, 2, 3).

$$\text{Bit rate} = \frac{f(PCLK)}{2 \times (n + 1) \times 2^{N}}$$

Table 44.3 lists examples of the relationship among the SPBR settings, the BRDV[1:0] settings, and bit rates.
Use the bit rate that meets electrical characteristics based on the AC specifications of the target device.

Table 44.3 Relationship among SPBR Settings, BRDV[1:0] Settings, and Bit Rates

SPBR (n)	BRDV[1:0] Bits (N)	Division Ratio	Bit Rate							
			PCLK = 32 MHz	PCLK = 36 MHz	PCLK = 40 MHz	PCLK = 50 MHz	PCLK = 60 MHz	PCLK = 80 MHz	PCLK = 100 MHz	PCLK = 120 MHz
0	0	2	16.0 Mbps	18.0 Mbps	20.0 Mbps	25.0 Mbps	30.0 Mbps	40.0 Mbps	—	—
1	0	4	8.00 Mbps	9.00 Mbps	10.0 Mbps	12.5 Mbps	15.0 Mbps	20.0 Mbps	25.0 Mbps	30.0 Mbps
2	0	6	5.33 Mbps	6.00 Mbps	6.67 Mbps	8.33 Mbps	10.0 Mbps	13.3 Mbps	16.7 Mbps	20.0 Mbps
3	0	8	4.00 Mbps	4.50 Mbps	5.00 Mbps	6.25 Mbps	7.50 Mbps	10.0 Mbps	12.5 Mbps	15.0 Mbps
4	0	10	3.20 Mbps	3.60 Mbps	4.00 Mbps	5.00 Mbps	6.00 Mbps	8.00 Mbps	10.0 Mbps	12.0 Mbps
5	0	12	2.67 Mbps	3.00 Mbps	3.33 Mbps	4.16 Mbps	5.00 Mbps	6.67 Mbps	8.33 Mbps	10.0 Mbps
5	1	24	1.33 Mbps	1.50 Mbps	1.67 Mbps	2.08 Mbps	2.50 Mbps	3.33 Mbps	4.17 Mbps	5.00 Mbps
5	2	48	667 kbps	750 kbps	833 kbps	1.04 Mbps	1.25 Mbps	1.67 Mbps	2.08 Mbps	2.50 Mbps
5	3	96	333 kbps	375 kbps	417 kbps	521 kbps	625 kbps	833 kbps	1.04 Mbps	1.25 Mbps
255	3	4096	7.81 kbps	8.80 kbps	9.78 kbps	12.2 kbps	14.6 kbps	19.5 kbps	24.4 kbps	29.3 kbps

44.2.9 RSPI Data Control Register (SPDCR)

Address(es): RSPI0.SPDCR 000D 010Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	SPLW	SPRDTD	—	—	SPFC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SPFC[1:0]	Number of Frames Specification	b1 b0 0 0: 1 frame 0 1: 2 frames 1 0: 3 frames 1 1: 4 frames	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SPRDTD	RSPI Receive/Transmit Data Select	0: SPDR values are read from the receive buffer 1: SPDR values are read from the transmit buffer (but only if the transmit buffer is empty)	R/W
b5	SPLW	RSPI Longword Access/ Word Access Specification	0: SPDR is accessed in words 1: SPDR is accessed in longwords	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Up to four frames can be transmitted or received in one round of transmission or reception activation. The amount of data in each transfer is controlled by the combination of the SPCMDm.SPB[3:0] bits, the SPSCR.SPSLN[2:0] bits, and the SPDCR.SPFC[1:0] bits.
When changing the SPDCR.SPFC[1:0] bits while the SPCR.SPE bit is 1, the bits should be changed while the SPSR.IDLNF flag is 0.

SPFC[1:0] Bits (Number of Frames Specification)

The SPFC[1:0] bits specify the number of frames that can be stored in SPDR (per transfer activation). Up to four frames can be transmitted or received in one round of transmission or reception, and the amount of data is determined by the combination of the SPSCR.SPSLN[2:0] bits, and the SPDCR.SPFC[1:0] bits. Furthermore, the setting of the SPFC[1:0] bits adjusts the number of frames for generation of RSPI receive buffer full interrupt, and start of transmission or generation of transmit buffer empty interrupts.
When the number of frames of transmit data specified by SPFC[1:0] bits is written to the SPDR register, the SPSR.SPTEF flag becomes 0 and transmission starts. Then, when the specified number of frames of transmit data has been transferred to the shift register, the SPTEF flag becomes 1 and the RSPI transmit buffer empty interrupt is generated.
When the number of frames specified by the SPFC[1:0] bits are received, the SPSR.SPRF flag becomes 1 and the RSPI receive buffer full interrupt is generated.
Table 44.4 lists the frame configurations that can be stored in SPDR and examples of combinations of settings for transmission and reception. If combinations of settings other than those shown in the examples are made, subsequent operations should not be performed.

Table 44.4 Settable Combinations of SPSLN[2:0] Bits and SPFC[1:0] Bits

Setting	SPSLN[2:0]	SPFC[1:0]	Number of Frames in a Single Sequence	Number of Frames at which Transmit Buffer or Receive Buffer Status Becomes “Has Valid Data”
1-1	000b	00b	1	1
1-2	000b	01b	2	2
1-3	000b	10b	3	3
1-4	000b	11b	4	4
2-1	001b	01b	2	2
2-2	001b	11b	4	4
3	010b	10b	3	3
4	011b	11b	4	4
5	100b	00b	5	1
6	101b	00b	6	1
7	110b	00b	7	1
8	111b	00b	8	1

SPRDTD Bit (RSPI Receive/Transmit Data Select)

The SPRDTD bit selects whether the SPDR reads values from the receive buffer or from the transmit buffer.
If reading is from the transmit buffer, the value written to SPDR register immediately beforehand is read.
When reading the transmit buffer, do so before writing of the number of frames set in the SPFC[1:0] bits is finished and after generation of the transmit buffer empty interrupt (While the SPSR.SPTEF flag is 1).
For details, refer to section 44.2.5, RSPI Data Register (SPDR).

SPLW Bit (RSPI Longword Access/Word Access Specification)

The SPLW bit specifies the access width for SPDR. Access to the SPDR register in words when the SPLW bit is 0 and in longwords when the SPLW bit is 1.
Also, when the SPLW bit is 0, set the SPCMDm.SPB[3:0] bits (RSPI data length setting bits) to 8 to 16 bits. When 20, 24, or 32 bits is specified, operations should not be performed.

44.2.10 RSPI Clock Delay Register (SPCKD)

Address(es): RSPI0.SPCKD 000D 010Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SCKDL[2:0]	RSPCK Delay Setting	b2 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SPCKD sets a period from the beginning of SSLAi signal assertion to RSPCK oscillation (RSPCK delay) when the SPCMDm.SCKDEN bit is 1. If the contents of SPCKD are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations should not be performed.

SCKDL[2:0] Bits (RSPCK Delay Setting)

The SCKDL[2:0] bits set an RSPCK delay value when the SPCMDm.SCKDEN bit is 1.
When using the RSPI in slave mode, set the SCKDL[2:0] bits to 000b.

44.2.11 RSPI Slave Select Negation Delay Register (SSLND)

Address(es): RSPI0.SSLND 000D 010Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SLNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SLNDL[2:0]	SSL Negation Delay Setting	b2 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

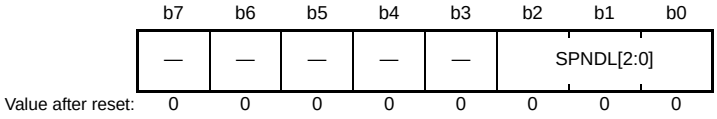
SSLND sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSLAi signal during a serial transfer by the RSPI in master mode. If the contents of SSLND are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations should not be performed.

SLNDL[2:0] Bits (SSL Negation Delay Setting)

The SLNDL[2:0] bits set an SSL negation delay value when the RSPI is in master mode.
When using the RSPI in slave mode, set the SLNDL[2:0] bits to 000b.

44.2.12 RSPI Next-Access Delay Register (SPND)

Address(es): RSPI0.SPND 000D 010Eh



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPNDL[2:0]	RSPI Next-Access Delay Setting	b2 b0 0 0 0: 1 RSPCK + 2 PCLK 0 0 1: 2 RSPCK + 2 PCLK 0 1 0: 3 RSPCK + 2 PCLK 0 1 1: 4 RSPCK + 2 PCLK 1 0 0: 5 RSPCK + 2 PCLK 1 0 1: 6 RSPCK + 2 PCLK 1 1 0: 7 RSPCK + 2 PCLK 1 1 1: 8 RSPCK + 2 PCLK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SPND sets a non-active period (next-access delay) of the SSLAi signal after termination of a serial transfer when the SPCMDm.SPNDEN bit is 1. If the contents of SPND are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations should not be performed.

SPNDL[2:0] Bits (RSPI Next-Access Delay Setting)

The SPNDL[2:0] bits set a next-access delay when the SPCMDm.SPNDEN bit is 1.
When using the RSPI in slave mode, set the SPNDL[2:0] bits to 000b.

44.2.13 RSPI Control Register 2 (SPCR2)

Address(es): RSPI0.SPCR2 000D 010Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SCKASE	PTE	SPIIE	SPOE	SPPE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPPE	Parity Enable	0: Does not add the parity bit to transmit data and does not check the parity bit of receive data 1: Adds the parity bit to transmit data and checks the parity bit of receive data (when SPCR.TXMD = 0) Adds the parity bit to transmit data but does not check the parity bit of receive data (when SPCR.TXMD = 1)	R/W
b1	SPOE	Parity Mode	0: Selects even parity for use in transmission and reception 1: Selects odd parity for use in transmission and reception	R/W
b2	SPIIE	RSPI Idle Interrupt Enable	0: Disables the generation of idle interrupt requests 1: Enables the generation of idle interrupt requests	R/W
b3	PTE	Parity Self-Diagnosis	0: Disables the self-diagnosis function of the parity circuit 1: Enables the self-diagnosis function of the parity circuit	R/W
b4	SCKASE	RSPCK Auto-Stop Function Enable	0: Disables the RSPCK auto-stop function 1: Enables the RSPCK auto-stop function	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the SPPE, SPOE, or SCKASE bit in SPCR2 is changed while the SPCR.SPE bit is 1, subsequent operations should not be performed.

SPPE Bit (Parity Enable)

The SPPE bit enables or disables the parity function.

The parity bit is added to transmit data and parity checking is performed for receive data when the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1.

The parity bit is added to transmit data but parity checking is not performed for receive data when the SPCR.TXMD bit is 1 and the SPCR2.SPPE bit is 1.

SPOE Bit (Parity Mode)

The SPOE bit specifies odd or even parity.

When even parity is set, parity bit addition is performed so that the total number of 1-bits in the transmit/receive character plus the parity bit is even. Similarly, when odd parity is set, parity bit addition is performed so that the total number of 1-bits in the transmit/receive character plus the parity bit is odd.

The SPOE bit is valid only when the SPPE bit is 1.

SPIIE Bit (RSPI Idle Interrupt Enable)

The SPIIE bit enables or disables the generation of RSPI idle interrupt requests when the RSPI being in the idle state is detected and the SPSR.IDLNF flag is set to 0.

PTE Bit (Parity Self-Diagnosis)

The PTE bit enables the self-diagnosis function of the parity circuit in order to check whether the parity function is operating correctly.

SCKASE Bit (RSPCK Auto-Stop Function Enable)

The SCKASE bit enables or disables the RSPCK auto-stop function. When this function is enabled, the RSPCK clock is stopped before an overrun error occurs when data is received in master mode. For details, refer to section 44.3.8.1, Overrun Error.

44.2.14 RSPI Command Registers 0 to 7 (SPCMD0 to SPCMD7)

Address(es): RSPI0.SPCMD0 000D 0110h, RSPI0.SPCMD1 000D 0112h, RSPI0.SPCMD2 000D 0114h, RSPI0.SPCMD3 000D 0116h, RSPI0.SPCMD4 000D 0118h, RSPI0.SPCMD5 000D 011Ah, RSPI0.SPCMD6 000D 011Ch, RSPI0.SPCMD7 000D 011Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]				SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	CPHA	RSPCK Phase Setting	0: Data sampling on odd edge, data variation on even edge 1: Data variation on odd edge, data sampling on even edge	R/W
b1	CPOL	RSPCK Polarity Setting	0: RSPCK is low when idle 1: RSPCK is high when idle	R/W
b3, b2	BRDV[1:0]	Bit Rate Division Setting	b3 b2 0 0: These bits select the base bit rate 0 1: These bits select the base bit rate divided by 2 1 0: These bits select the base bit rate divided by 4 1 1: These bits select the base bit rate divided by 8	R/W
b6 to b4	SSLA[2:0]	SSL Signal Assertion Setting	b6 b4 0 0 0: SSL0 0 0 1: SSL1 0 1 0: SSL2 0 1 1: SSL3 1 x x: Setting prohibited x: Don't care	R/W
b7	SSLKP	SSL Signal Level Keeping	0: Negates all SSL signals upon completion of transfer 1: Keeps the SSL signal level from the end of transfer until the beginning of the next access	R/W
b11 to b8	SPB[3:0]	RSPI Data Length Setting	b11 b8 0100 to 0111: 8 bits 1 0 0 0: 9 bits 1 0 0 1: 10 bits 1 0 1 0: 11 bits 1 0 1 1: 12 bits 1 1 0 0: 13 bits 1 1 0 1: 14 bits 1 1 1 0: 15 bits 1 1 1 1: 16 bits 0 0 0 0: 20 bits 0 0 0 1: 24 bits 0010, 0011: 32 bits	R/W
b12	LSBF	RSPI LSB First	0: MSB first 1: LSB first	R/W
b13	SPNDEN	RSPI Next-Access Delay Enable	0: A next-access delay of 1 RSPCK + 2 PCLK 1: A next-access delay is equal to the setting of the RSPI next-access delay register (SPND)	R/W
b14	SLNDEN	SSL Negation Delay Setting Enable	0: An SSL negation delay of 1 RSPCK 1: An SSL negation delay is equal to the setting of the RSPI slave select negation delay register (SSLND)	R/W
b15	SCKDEN	RSPCK Delay Setting Enable	0: An RSPCK delay of 1 RSPCK 1: An RSPCK delay is equal to the setting of the RSPI clock delay register (SPCKD)	R/W

SPCMDm register is used to set a transfer format for the RSPI in master mode. Each channel has eight RSPI command registers (SPCMD0 to SPCMD7). Some of the bits in SPCMD0 register is used to set a transfer mode for the RSPI in slave mode. The RSPI in master mode sequentially references SPCMDm register according to the settings in the SPSCR.SPSLN[2:0] bits, and executes the serial transfer that is set in the referenced SPCMDm register.
SPCMDm register should be set while the transmit buffer is empty (data for the next transfer is not set) and before setting of the data that is to be transmitted when that SPCMDm register is referenced.
SPCMDm that is referenced by the RSPI in master mode can be checked by means of the SPSSR.SPCP[2:0] bits. If the contents of SPCMDm are changed while the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1, subsequent operations should not be performed.

CPHA Bit (RSPCK Phase Setting)

The CPHA bit sets an RSPCK phase of the RSPI in master mode or slave mode. Data communications between RSPI modules require the same RSPCK phase setting between the modules.

CPOL Bit (RSPCK Polarity Setting)

The CPOL bit sets an RSPCK polarity of the RSPI in master mode or slave mode. Data communications between RSPI modules require the same RSPCK polarity setting between the modules.

BRDV[1:0] Bits (Bit Rate Division Setting)

The BRDV[1:0] bits are used to determine the bit rate. A bit rate is determined by combinations of the settings in the BRDV[1:0] bits and SPBR (refer to section 44.2.8, RSPI Bit Rate Register (SPBR)). The settings in SPBR determine the base bit rate. The settings in the BRDV[1:0] bits are used to select a bit rate which is obtained by dividing the base bit rate by 1, 2, 4, or 8. In SPCMDm register, different BRDV[1:0] bit settings can be specified. This enables execution of serial transfers at a different bit rate for each command.

SSLA[2:0] Bits (SSL Signal Assertion Setting)

The SSLA[2:0] bits control the SSLAi signal assertion when the RSPI performs serial transfers in master mode.
Setting the SSLA[2:0] bits controls the assertion for the SSLAi signal. When an SSLAi signal is asserted, its polarity is determined by the set value in the corresponding SSLP. When the SSLA[2:0] bits are set to 000b in multi-master mode, serial transfers are performed with all the SSL signals in the negated state (as the SSLA0 pin acts as input).
When using the RSPI in slave mode, set the SSLA[2:0] bits to 000b.

SSLKP Bit (SSL Signal Level Keeping)

When the RSPI in master mode performs a serial transfer, the SSLKP bit specifies whether the SSLAi signal level for the current command is to be kept or negated between the SSL negation timing associated with the current command and the SSL assertion timing associated with the next command.
Setting the SSLKP bit to 1 enables a burst transfer. For details, refer to section 44.3.10.1, Master Mode Operation (4) Burst Transfer.
When using the RSPI in slave mode, the SSLKP bit should be set to 0.

SPB[3:0] Bits (RSPI Data Length Setting)

The SPB[3:0] bits set a transfer data length for the RSPI in master mode or slave mode. When the SPDCR.SPLW bit is 0, set the SPB[3:0] bits to “0100b” (8 bits) to “1111b” (16 bits).

LSBF Bit (RSPI LSB First)

The LSBF bit sets the data format of the RSPI in master mode or slave mode to MSB first or LSB first.

SPNDEN Bit (RSPI Next-Access Delay Enable)

The SPNDEN bit sets the period from the time the RSPI in master mode terminates a serial transfer and sets the SSLAi signal inactive until the RSPI enables the SSLAi signal assertion for the next access (next-access delay). If the SPNDEN bit is 0, the RSPI sets the next-access delay to 1 RSPCK + 2 PCLK. If the SPNDEN bit is 1, the RSPI inserts a next-access delay in compliance with the SPND setting.

When using the RSPI in slave mode, the SPNDEN bit should be set to 0.

SLNDEN Bit (SSL Negation Delay Setting Enable)

The SLNDEN bit sets the period from the time the RSPI in master mode stops RSPCK oscillation until the RSPI sets the SSLAi signal inactive (SSL negation delay). If the SLNDEN bit is 0, the RSPI sets the SSL negation delay to 1 RSPCK. If the SLNDEN bit is 1, the RSPI negates the SSL signal at an SSL negation delay in compliance with the SSLND setting.

When using the RSPI in slave mode, the SLNDEN bit should be set to 0.

SCKDEN Bit (RSPCK Delay Setting Enable)

The SCKDEN bit sets the period from the point when the RSPI in master mode activates the SSLAi signal until the RSPCK starts oscillation (RSPI clock delay). If the SCKDEN bit is 0, the RSPI sets the RSPCK delay to 1 RSPCK. If the SCKDEN bit is 1, the RSPI starts the oscillation of RSPCK at an RSPCK delay in compliance with the SPCKD setting.

When using the RSPI in slave mode, the SCKDEN bit should be set to 0.

44.3 Operation

In this section, the serial transfer period means a period from the beginning of driving valid data to the fetching of the final valid data.

44.3.1 Overview of RSPI Operations

The RSPI is capable of synchronous serial transfers in slave mode (SPI operation), single-master mode (SPI operation), multi-master mode (SPI operation), slave mode (clock synchronous operation), and master mode (clock synchronous operation). A particular mode of the RSPI can be selected by using the MSTR, MODFEN, and SPMS bits in SPCR. Table 44.5 lists the relationship between RSPI modes and SPCR settings, and a description of each mode.

Table 44.5 Relationship between RSPI Modes and SPCR Settings and Description of Each Mode

Mode	Slave (SPI Operation)	Single-Master (SPI Operation)	Multi-Master (SPI Operation)	Slave (Clock Synchronous Operation)	Master (Clock Synchronous Operation)
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
RSPCKA signal	Input	Output	Output/Hi-Z	Input	Output
MOSIA signal	Input	Output	Output/Hi-Z	Input	Output
MISOA signal	Output/Hi-Z	Input	Input	Output	Input
SSLA0 signal	Input	Output	Input	Hi-Z*1	Hi-Z*1
SSLA1 to SSLA3 signals	Hi-Z*1	Output	Output/Hi-Z	Hi-Z*1	Hi-Z*1
SSL polarity change function	Supported	Supported	Supported	—	—
Transfer rate	Up to PCLK/8	Up to PCLK/2	Up to PCLK/2	Up to PCLK/8	Up to PCLK/2
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	Two				
Clock phase	Two	Two	Two	One (CPHA = 1)	Two
First transfer bit	MSB/LSB				
Transfer data length	8 to 16, 20, 24, 32 bits				
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0,1)	Possible (CPHA = 0,1)	—	—
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported
Transfer activation method	SSL input active or RSPCK oscillation	Transmit buffer is written to at generation of a transmit buffer empty interrupt request or when the SPTEF flag is 1	Transmit buffer is written to at generation of a transmit buffer empty interrupt request or when the SPTEF flag is 1	RSPCK oscillation	Transmit buffer is written to at generation of a transmit buffer empty interrupt request or when the SPTEF flag is 1
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported				
Receive buffer full detection	Supported*2				
Overrun error detection	Supported*2	Supported*2, *4	Supported*2, *4	Supported*2	Supported*2
Parity error detection	Supported*2,*3				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported

Note 1. This function is not supported in this mode.
Note 2. When the SPCR.TXMD bit is 1, receiver buffer full detection, overrun error detection, and parity error detection are not performed.
Note 3. When the SPCR2.SPPE bit is 0, parity error detection is not performed.
Note 4. When the SPCR2.SCKASE bit is 1, overrun error detection does not proceed.

44.3.2 Controlling RSPI Pins

According to the MSTR, MODFEN, and SPMS bits in SPCR and the ODRn.Bi bit for I/O ports, the RSPI can switch pin states. Table 44.6 lists the relationship between pin states and bit settings. Setting the ODRn.Bi bit for an I/O port to 0 selects CMOS output; setting it to 1 selects open-drain output. The I/O port settings should follow this relationship.

Table 44.6 Relationship between Pin States and Bit Settings

Mode	Pin	Pin State*2	
		ODRn.Bi Bit for I/O Ports = 0	ODRn.Bi Bit for I/O Ports = 1
Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCKA	CMOS output	Open-drain output
	SSLA0 to SSLA3	CMOS output	Open-drain output
	MOSIA	CMOS output	Open-drain output
	MISOA	Input	Input
Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCKA*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSLA0	Input	Input
	SSLA1 to SSLA3*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSIA*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISOA	Input	Input
Slave mode (SPI operation) (MSTR = 0, SPMS = 0)	RSPCKA	Input	Input
	SSLA0	Input	Input
	SSLA1 to SSLA3*5	Hi-Z*1	Hi-Z*1
	MOSIA	Input	Input
	MISOA*4	CMOS output/Hi-Z	Open-drain output/Hi-Z
Master mode (Clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCKA	CMOS output	Open-drain output
	SSLA0 to SSLA3*5	Hi-Z*1	Hi-Z*1
	MOSIA	CMOS output	Open-drain output
	MISOA	Input	Input
Slave mode (Clock synchronous operation) (MSTR = 0, SPMS = 1)	RSPCKA	Input	Input
	SSLA0 to SSLA3*5	Hi-Z*1	Hi-Z*1
	MOSIA	Input	Input
	MISOA	CMOS output	Open-drain output

Note 1. This function is not supported in this mode.
Note 2. RSPI settings are not reflected in the multiplex pins for which the RSPI function is not selected.
Note 3. When SSLA0 is at the active level, the pin state is Hi-Z.
Note 4. When SSLA0 is at the non-active level or the SPCR.SPE bit is 0, the pin state is Hi-Z.
Note 5. These pins are available for use as I/O port pins.

The RSPI in single-master mode (SPI operation) or multi-master mode (SPI operation) determines MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer) according to MOIFE and MOIFV bit settings in SPPCR, as listed in Table 44.7.

Table 44.7 MOSI Signal Value Determination during SSL Negation Period

MOIFE Bit	MOIFV Bit	MOSIA Signal Value during SSL Negation Period
0	0, 1	Final data from previous transfer
1	0	Low
1	1	High

44.3.3 RSPI System Configuration Examples

44.3.3.1 Single Master/Single Slave (with This MCU Acting as Master)

Figure 44.5 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a master. In the single-master/single-slave configuration, the SSLA0 to SSLA3 output of this MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave is maintained in a select state.*1
This MCU (master) drives the RSPCKA and MOSIA. The SPI slave drives the MISO.

Note 1. In the transfer format corresponding to the case where the SPCMDm.CPHA bit is 0, there are slave devices for which the SSL signal cannot be fixed to the active level. In situations where the SSL signal cannot be fixed, the SSLAi output of this MCU should be connected to the SSL input of the slave device.

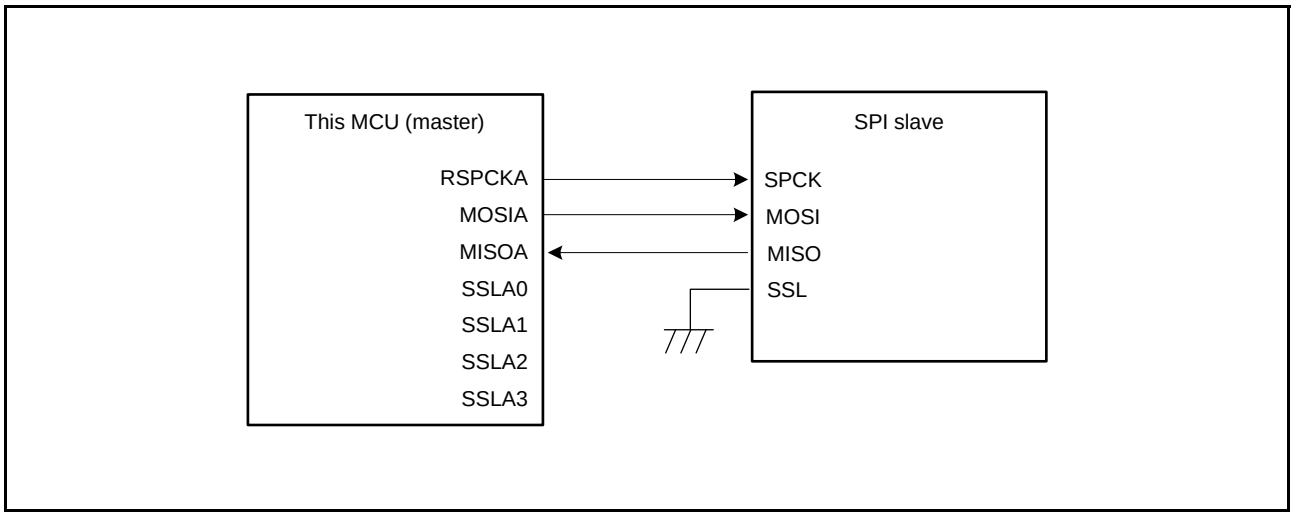


Figure 44.5 Single-Master/Single-Slave Configuration Example (This MCU = Master)

44.3.3.2 Single Master/Single Slave (with This MCU Acting as Slave)

Figure 44.6 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave, the SSLA0 pin is used as SSL input. The SPI master drives the RSPCK and MOSI. This MCU (slave) drives the MISOA.*1

In the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, the SSLA0 input of this MCU (slave) is fixed to the low level, this MCU (slave) is maintained in a select state, and in this manner it is possible to execute serial transfer (Figure 44.7).

Note 1. When SSLA0 is at the non-active level, the pin state is Hi-Z.

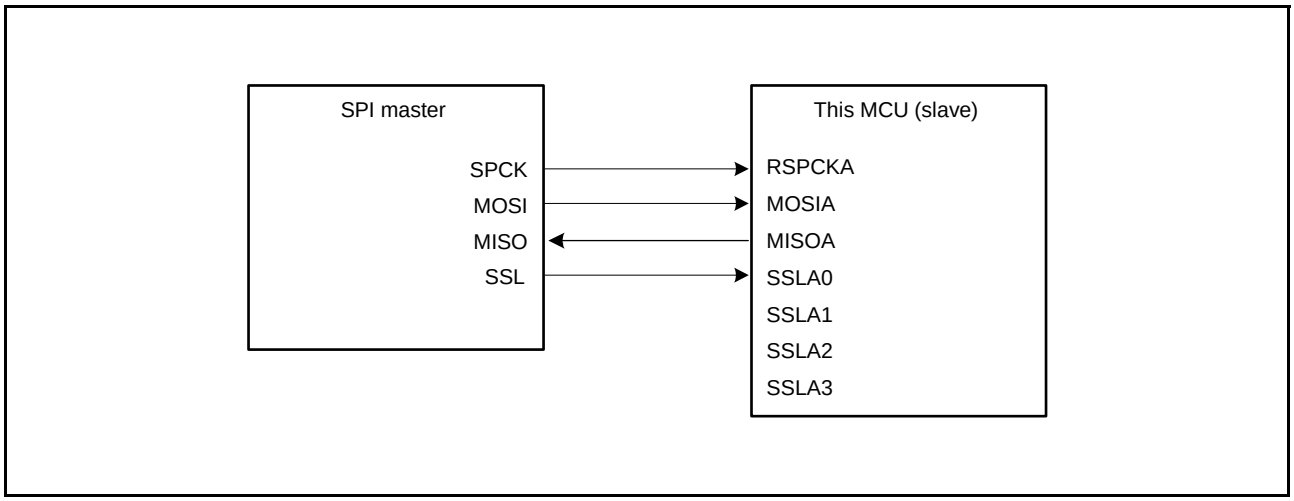


Figure 44.6 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 0)

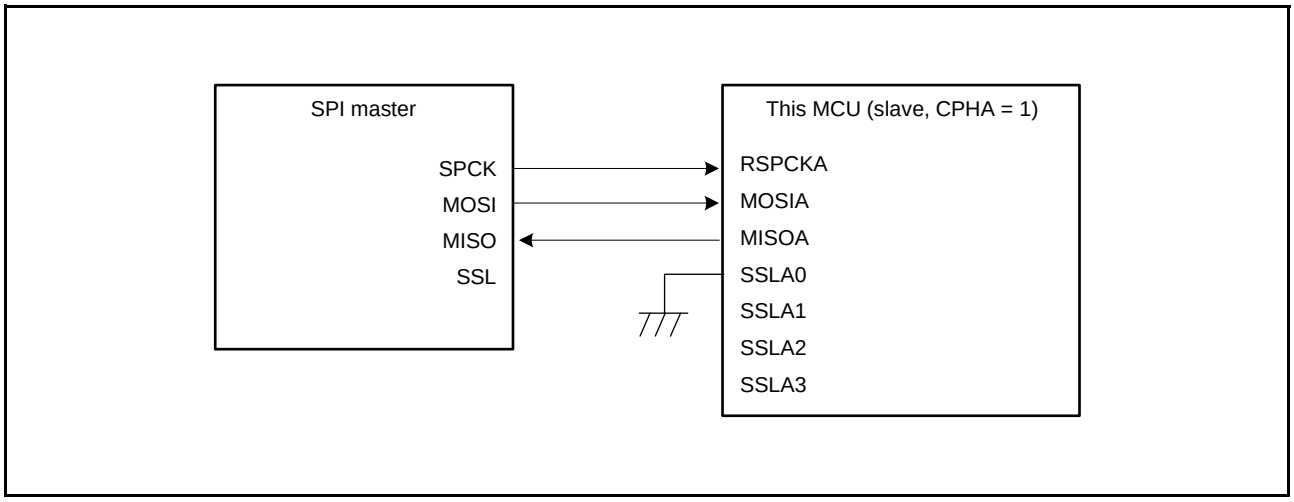


Figure 44.7 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 1)

44.3.3.3 Single Master/Multi-Slave (with This MCU Acting as Master)

Figure 44.8 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 44.8, the RSPI system is comprised of this MCU (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCKA and MOSIA outputs of this MCU (master) are connected to the RSPCK and MOSI inputs of SPI slave 0 to SPI slave 3. The MISO outputs of SPI slave 0 to SPI slave 3 are all connected to the MISOA input of this MCU (master). SSLA0 to SSLA3 outputs of this MCU (master) are connected to the SSL inputs of SPI slave 0 to SPI slave 3, respectively.

This MCU (master) drives RSPCKA, MOSIA, and SSLA0 to SSLA3. Of the SPI slave 0 to SPI slave 3, the slave that receives low-level input into the SSL input drives MISO.

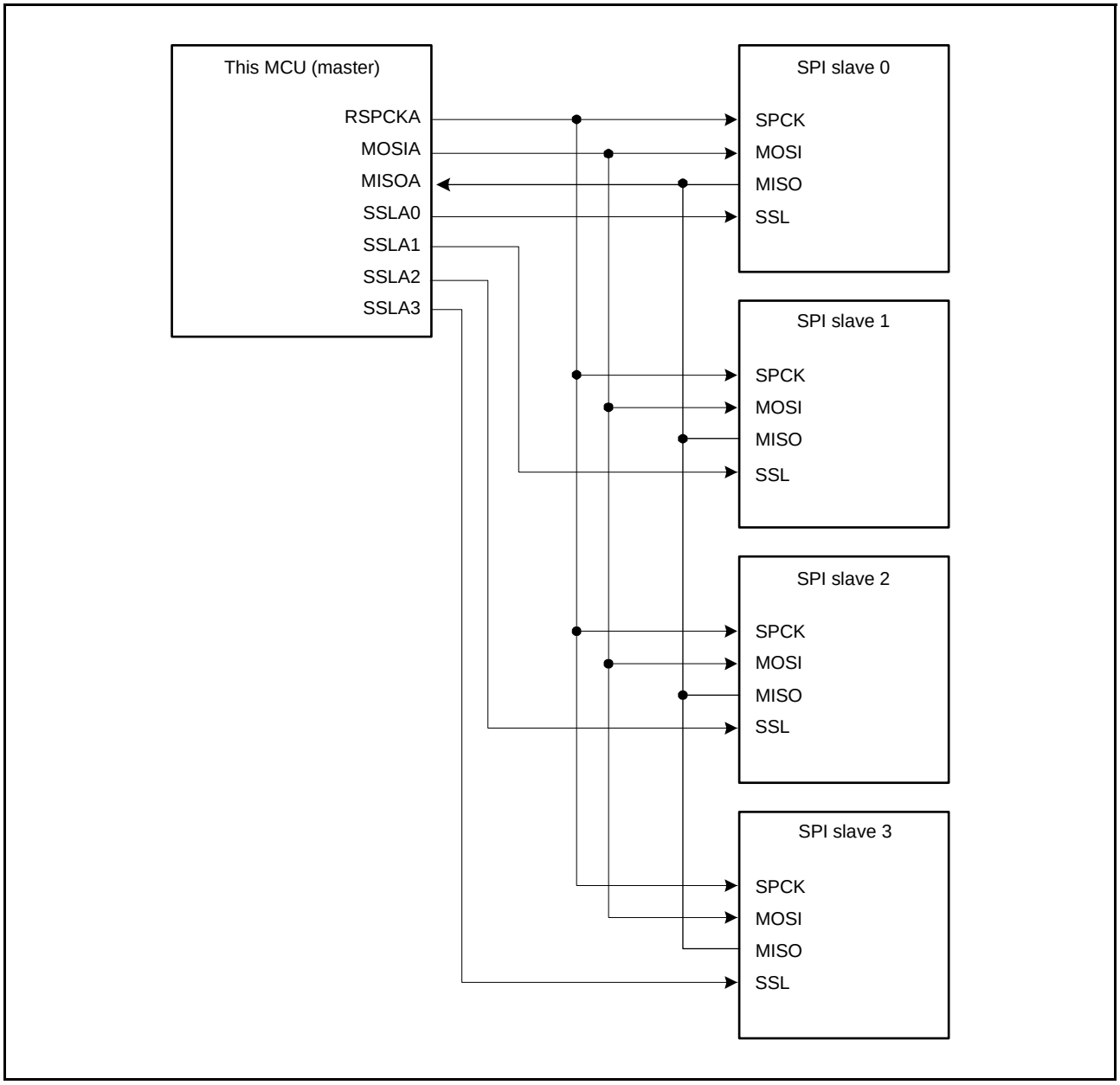


Figure 44.8 Single-Master/Multi-Slave Configuration Example (This MCU = Master)

44.3.3.4 Single Master/Multi-Slave (with This MCU Acting as Slave)

Figure 44.9 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a slave. In the example of Figure 44.9, the RSPI system is comprised of an SPI master and two MCUs (slave X and slave Y). The SPCK and MOSI outputs of the SPI master are connected to the RSPCKA and MOSIA inputs of the MCUs (slave X and slave Y). The MISOA outputs of the MCUs (slave X and slave Y) are all connected to the MISO input of the SPI master. SSLX and SSLY outputs of the SPI master are connected to the SSLA0 inputs of the MCUs (slave X and slave Y), respectively.
The SPI master drives SPCK, MOSI, SSLX, and SSLY. Of the MCUs (slave X and slave Y), the slave that receives low-level input into the SSLA0 input drives MISOA.

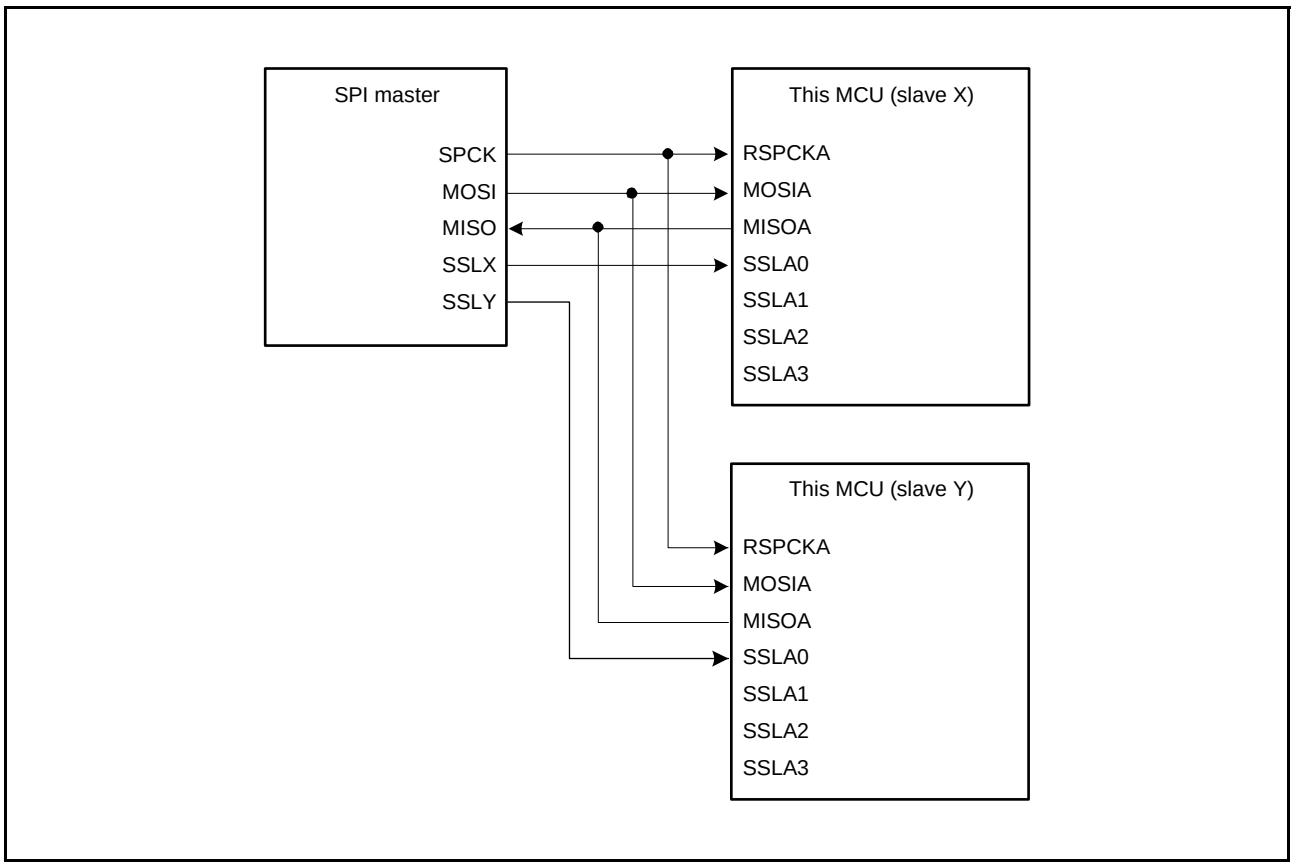


Figure 44.9 Single-Master/Multi-Slave Configuration Example (This MCU = Slave)

44.3.3.5 Multi-Master/Multi-Slave (with This MCU Acting as Master)

Figure 44.10 shows a multi-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 44.10, the RSPI system is comprised of two MCUs (master X and master Y) and two SPI slaves (SPI slave 1 and SPI slave 2).

The RSPCKA and MOSIA outputs of the MCUs (master X and master Y) are connected to the RSPCK and MOSI inputs of SPI slaves 1 and 2. The MISO outputs of SPI slaves 1 and 2 are connected to the MISOA inputs of the MCUs (master X and master Y). Any generic port Y output from this MCU (master X) is connected to the SSLA0 input of this MCU (master Y). Any generic port X output of this MCU (master Y) is connected to the SSLA0 input of this MCU (master X). The SSLA1 and SSLA2 outputs of the MCUs (master X and master Y) are connected to the SSL inputs of the SPI slaves 1 and 2. In this configuration example, since the system can be comprised solely of SSLA0 input, and SSLA1 and SSLA2 outputs for slave connections, the SSLA3 output of this MCU is not required.

This MCU drives RSPCKA, MOSIA, SSLA1, and SSLA2 when the SSLA0 input level is high. When the SSLA0 input level is low, this MCU detects a mode fault error, sets RSPCKA, MOSIA, SSLA1, and SSLA2 to Hi-Z, and releases the RSPI bus right to the other master. Of the SPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives MISO.

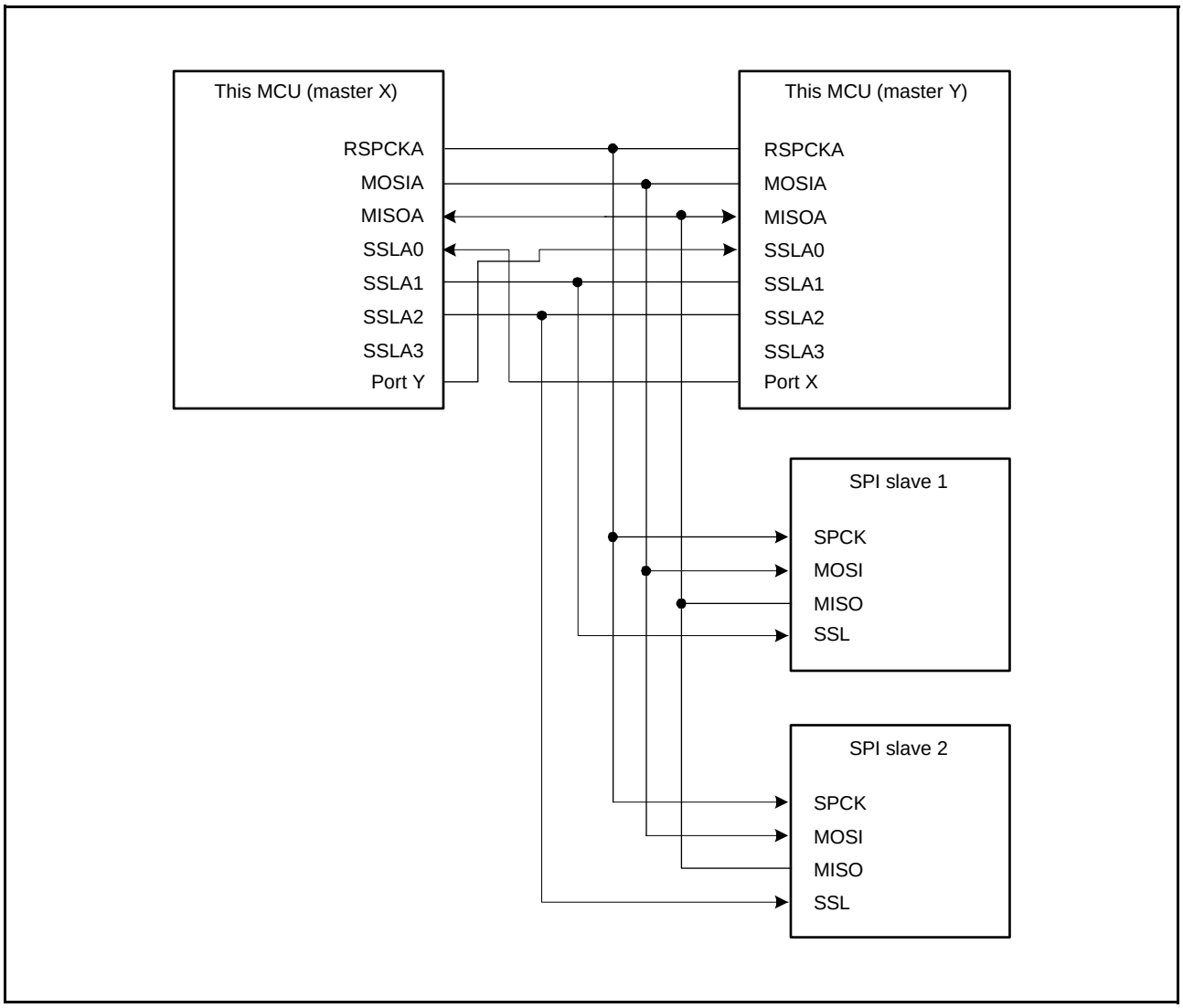


Figure 44.10 Multi-Master/Multi-Slave Configuration Example (This MCU = Master)

44.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Master)

Figure 44.11 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a master. In the master (clock synchronous operation)/slave (clock synchronous operation) configuration, SSLA0 to SSLA3 of this MCU (master) are not used.
This MCU (master) drives the RSPCKA and MOSIA. The SPI slave drives the MISO.

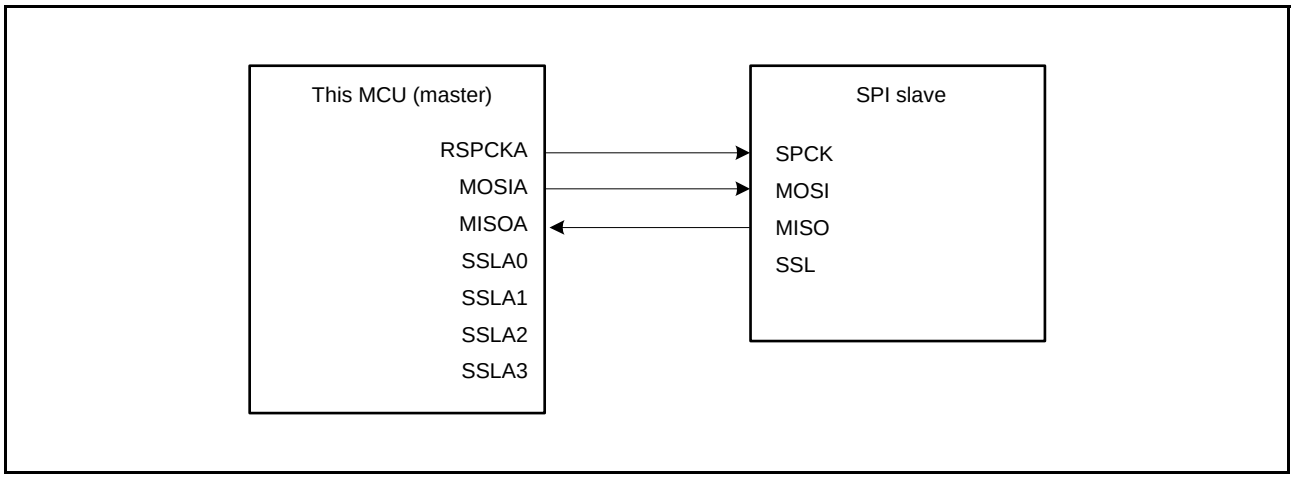


Figure 44.11 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Master)

44.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Slave)

Figure 44.12 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave (clock synchronous operation), this MCU (slave) drives the MISOA and the SPI master drives the SPCK and MOSI. In addition, SSLA0 to SSLA3 of this MCU (slave) are not used.
Only in the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, this MCU (slave) can execute serial transfer.

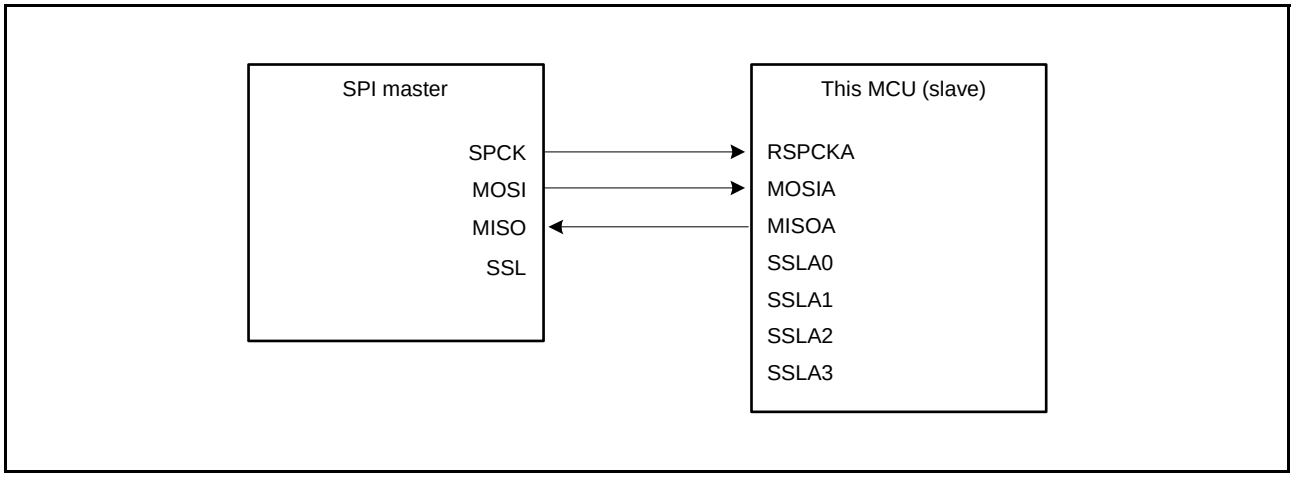


Figure 44.12 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Slave, CPHA = 1)

44.3.4 Data Format

The RSPI's data format depends on the settings in RSPI command register m (SPCMDm) (m = 0 to 7) and the parity enable bit in RSPI control register 2 (SPCR2.SPPE). Regardless of whether the MSB or LSB is first, the RSPI treats the range from the LSB bit in the RSPI data register (SPDR) to the selected data length as transfer data.
The format of one frame of data before or after transfer is shown below.

(a) With Parity Disabled

When parity is disabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[3:0]).

(b) With Parity Enabled

When parity is enabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[3:0]). In this case, however, the last bit is a parity bit.

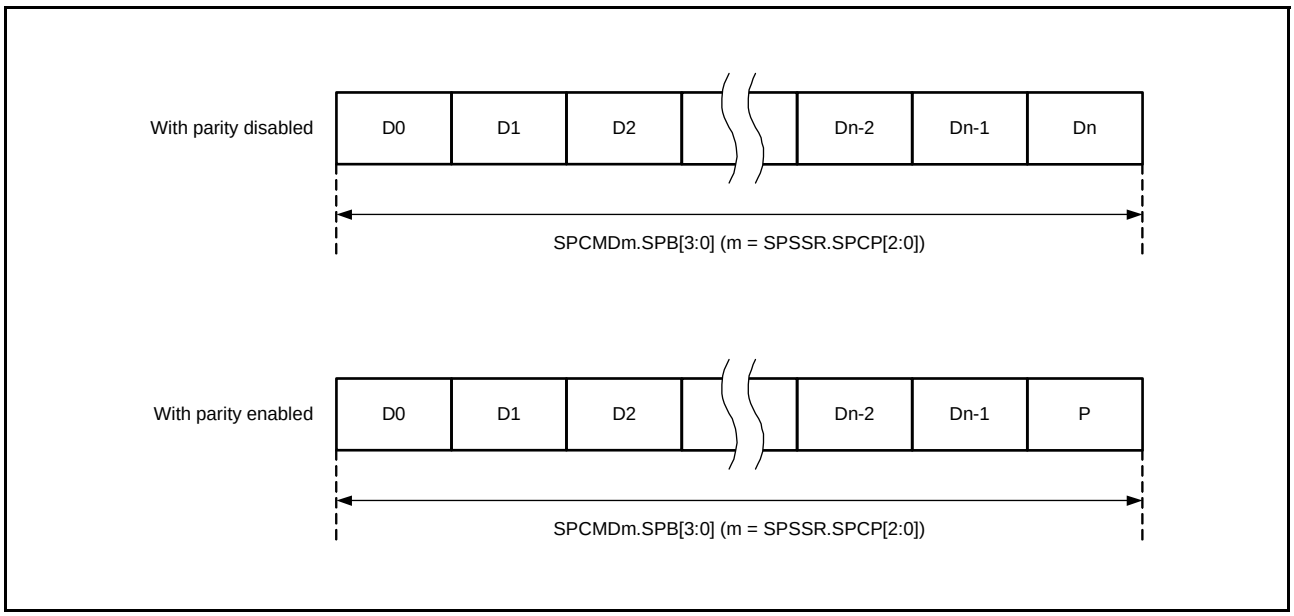


Figure 44.13 Outline of the Data Format (with Parity Disabled/Enabled)

44.3.4.1 When Parity is Disabled (SPCR2.SPPE = 0)

When parity is disabled, data for transmission are copied to the shift register with no prior processing. A description of the connection between the RSPI data register (SPDR) and the shift register in terms of the combination of MSB or LSB first and data length is given below.

(1) MSB First Transfer (32-Bit Data)

Figure 44.14 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and MSB first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T31, through T30, and so on to T00.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

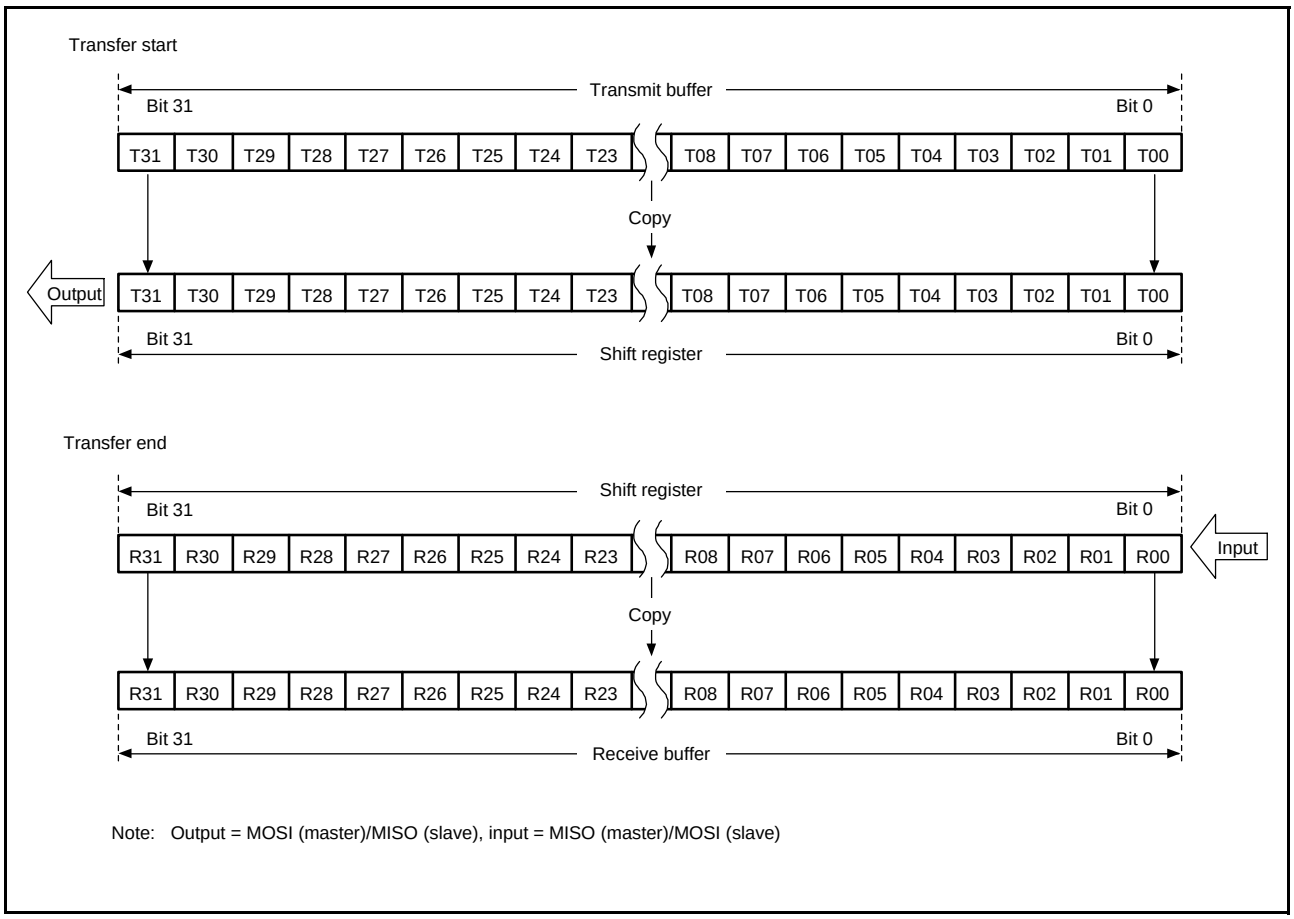


Figure 44.14 MSB First Transfer (32-Bit Data, Parity Disabled)

(2) MSB First Transfer (24-Bit Data)

Figure 44.15 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.

In transmission, the lower-order 24 bits (T23 to T00) from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T23, through T22, and so on to T00.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer.

Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

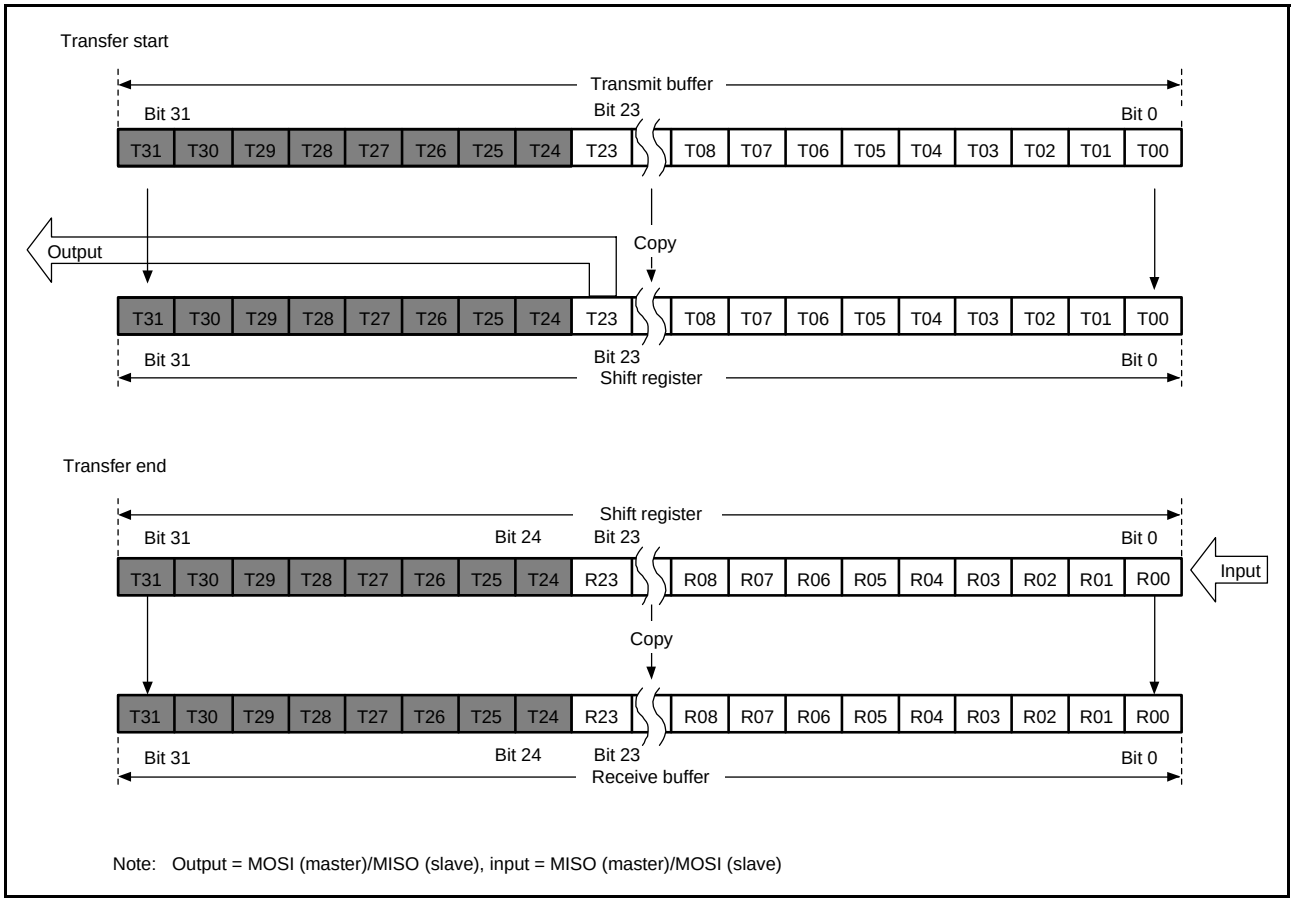


Figure 44.15 MSB First Transfer (24-Bit Data, Parity Disabled)

(3) LSB First Transfer (32-Bit Data)

Figure 44.16 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and LSB first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T31 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T31.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to R31 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

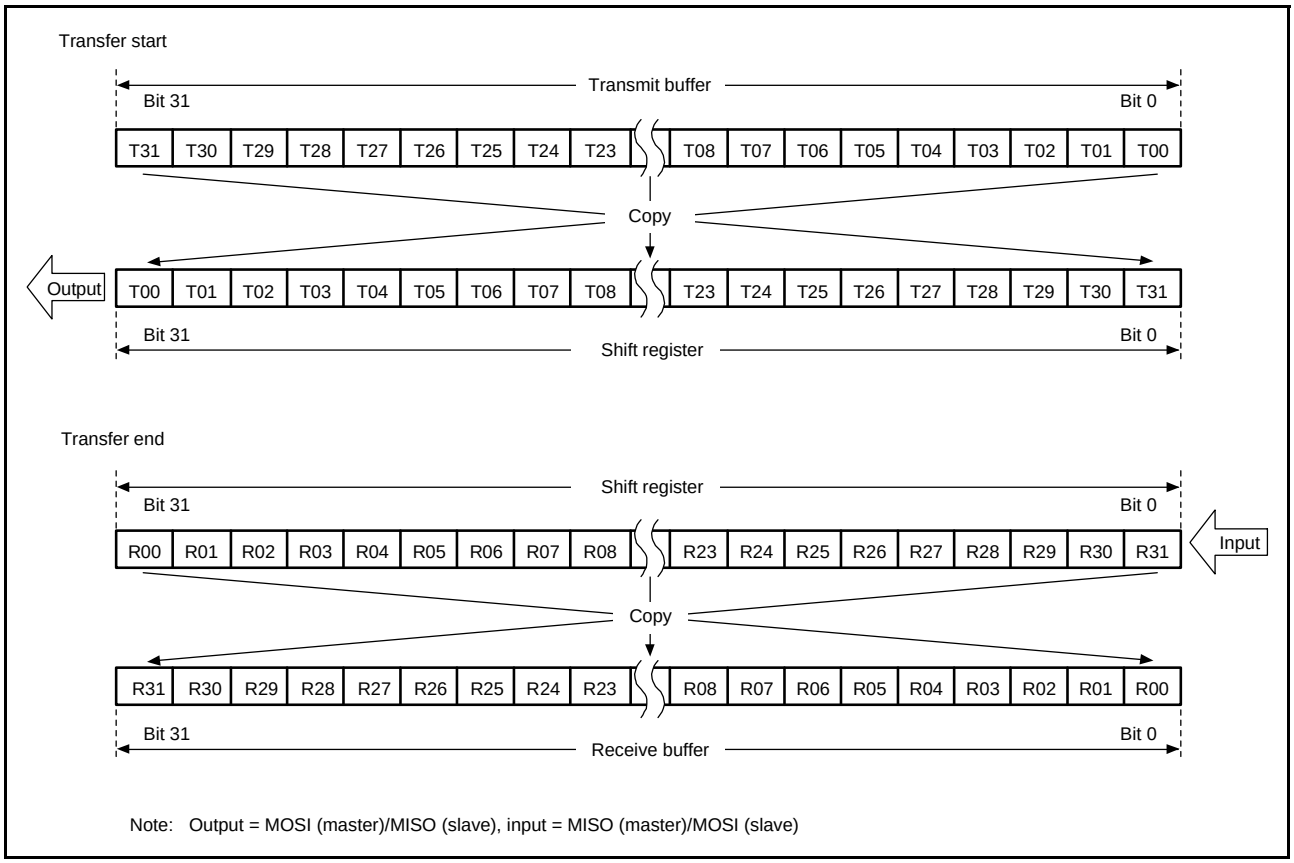


Figure 44.16 LSB First Transfer (32-Bit Data, Parity Disabled)

(4) LSB First Transfer (24-Bit Data)

Figure 44.17 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.

In transmission, the lower-order 24 bits (T23 to T00) from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T23 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T23.

In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to R23 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer.

Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

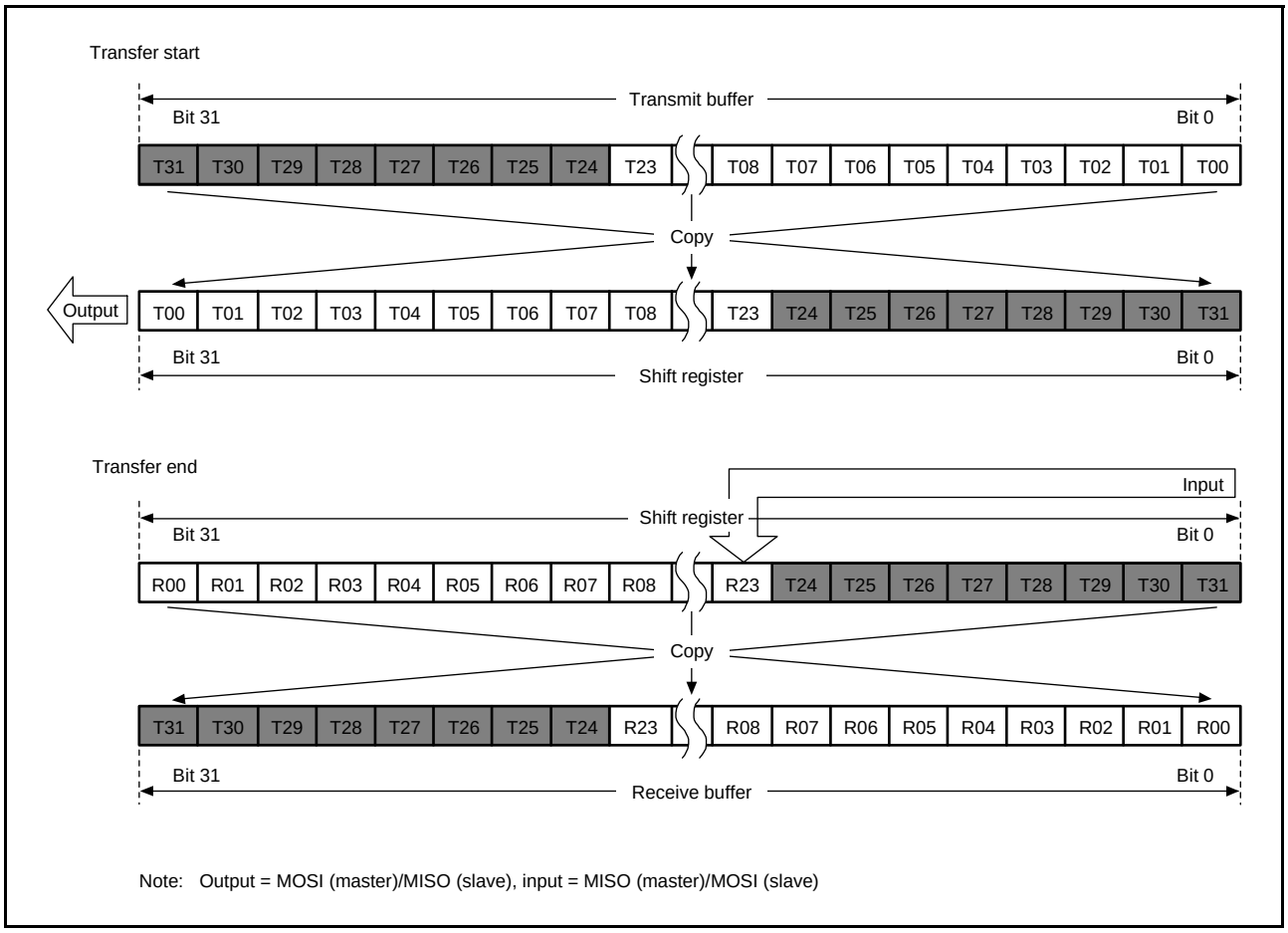


Figure 44.17 LSB First Transfer (24-Bit Data, Parity Disabled)

44.3.4.2 When Parity is Enabled (SPCR2.SPPE = 1)

When parity is enabled, the lowest-order bit of the data for transmission becomes a parity bit. Hardware calculates the value of the parity bit.

(1) MSB First Transfer (32-Bit Data)

Figure 44.18 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and MSB first selected.

In transmission, the value of the parity bit (P) is calculated from bits T31 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T31, T30, …, T01, and P.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R31 to P are checked by judging the parity.

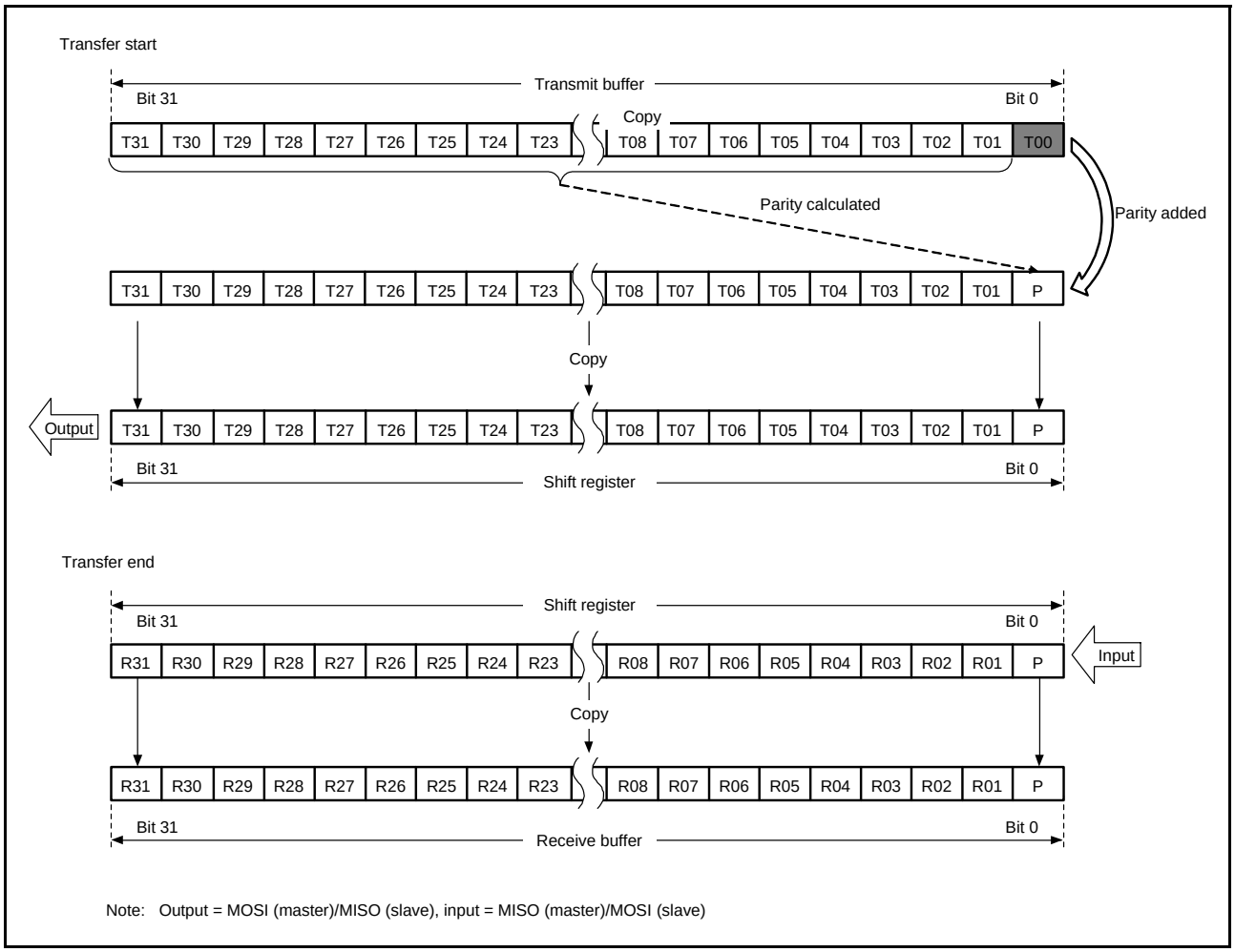


Figure 44.18 MSB First Transfer (32-Bit Data, Parity Enabled)

(2) MSB First Transfer (24-Bit Data)

Figure 44.19 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T23, T22, …, T01, and P.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R23 to P are checked by judging the parity. At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

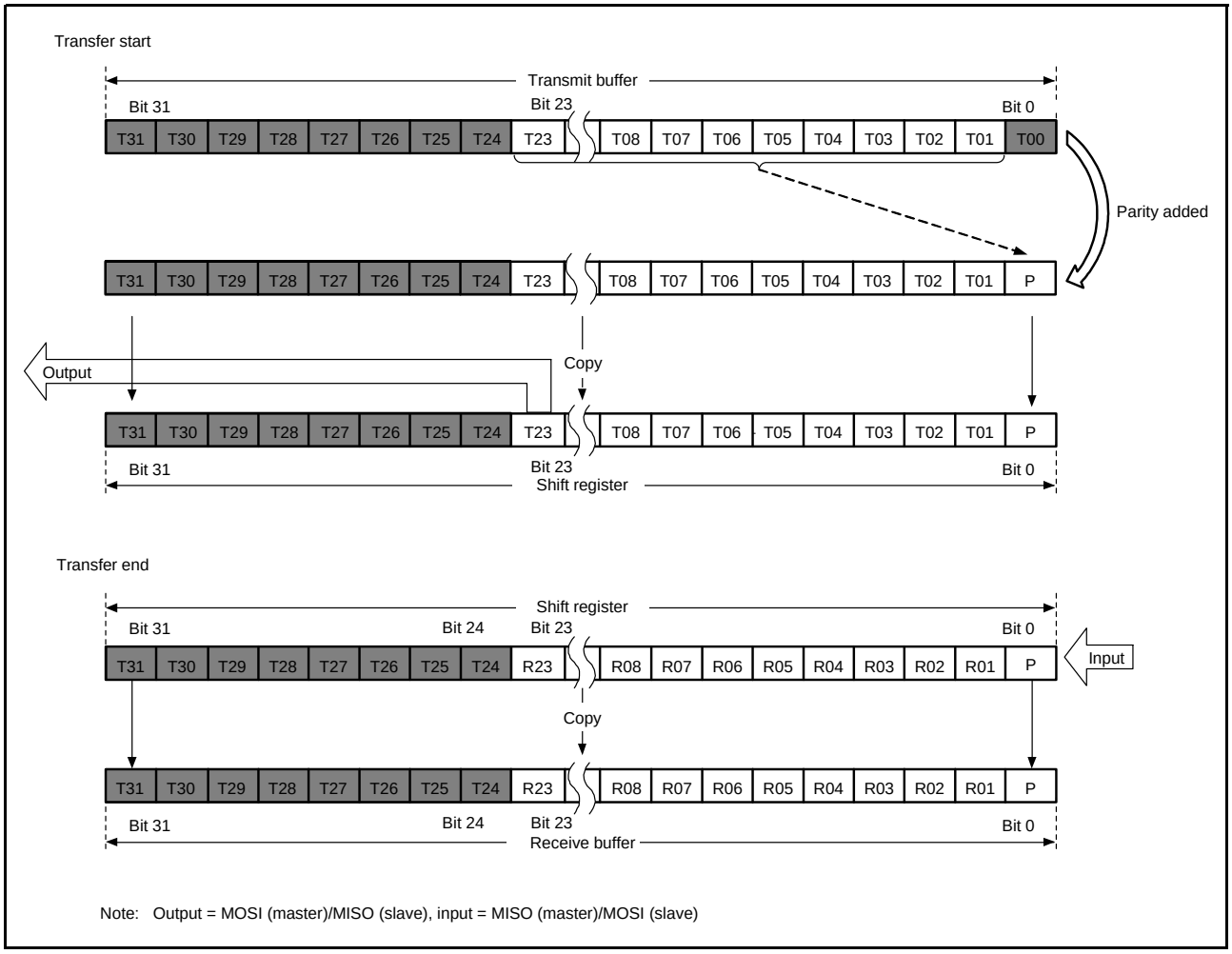


Figure 44.19 MSB First Transfer (24-Bit Data, Parity Enabled)

(3) LSB First Transfer (32-Bit Data)

Figure 44.20 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and LSB first selected.

In transmission, the value of the parity bit (P) is calculated from bits T30 to T00. This replaces the final bit, T31, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T30, and P.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P are checked by judging the parity.

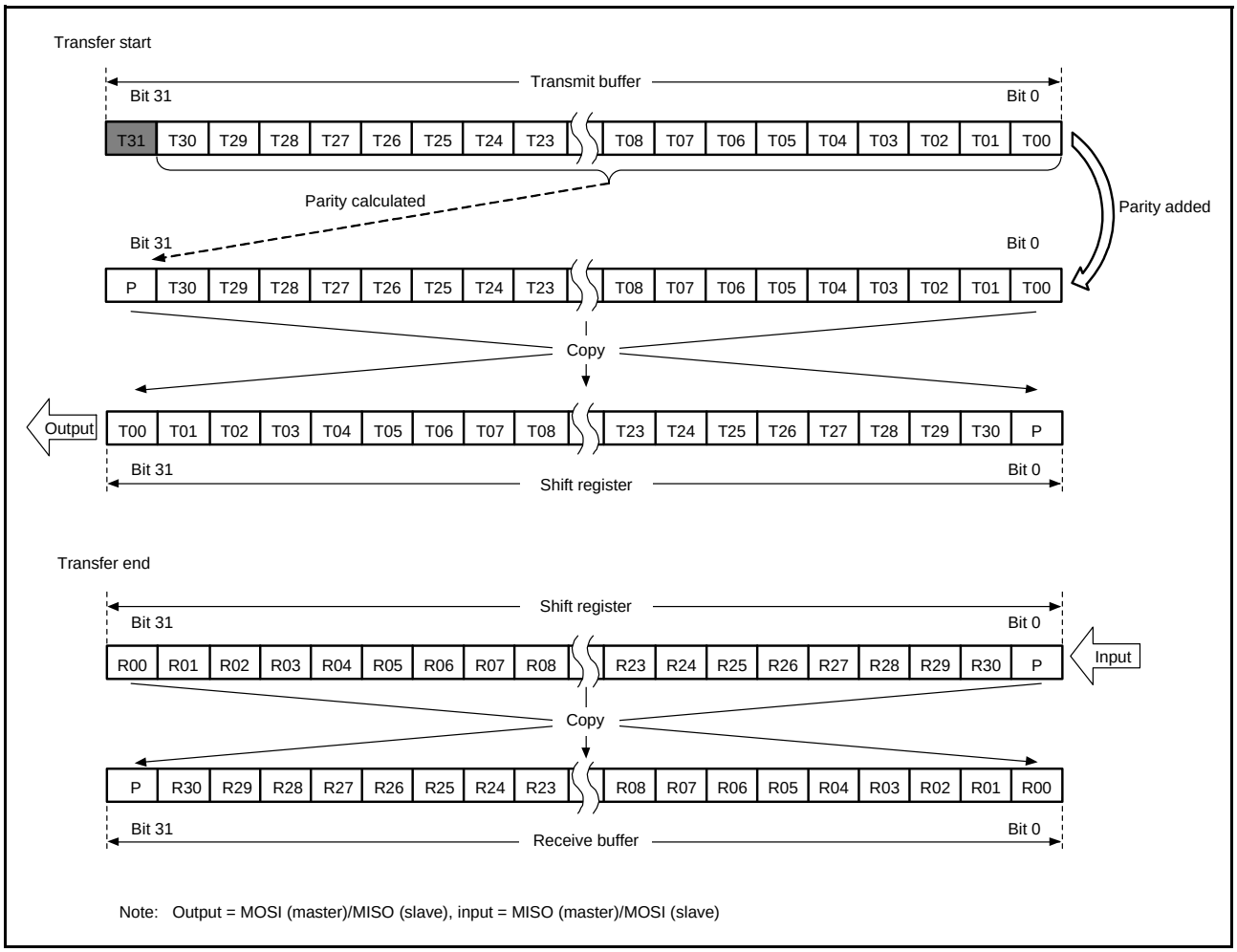


Figure 44.20 LSB First Transfer (32-Bit Data, Parity Enabled)

(4) LSB First Transfer (24-Bit Data)

Figure 44.21 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T22 to T00. This replaces the final bit, T23, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T22, and P.
In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P are checked by judging the parity. At this time, the higher-order 8 bits of the transmit buffer are stored in the higher-order 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order 8 bits of the receive buffer.

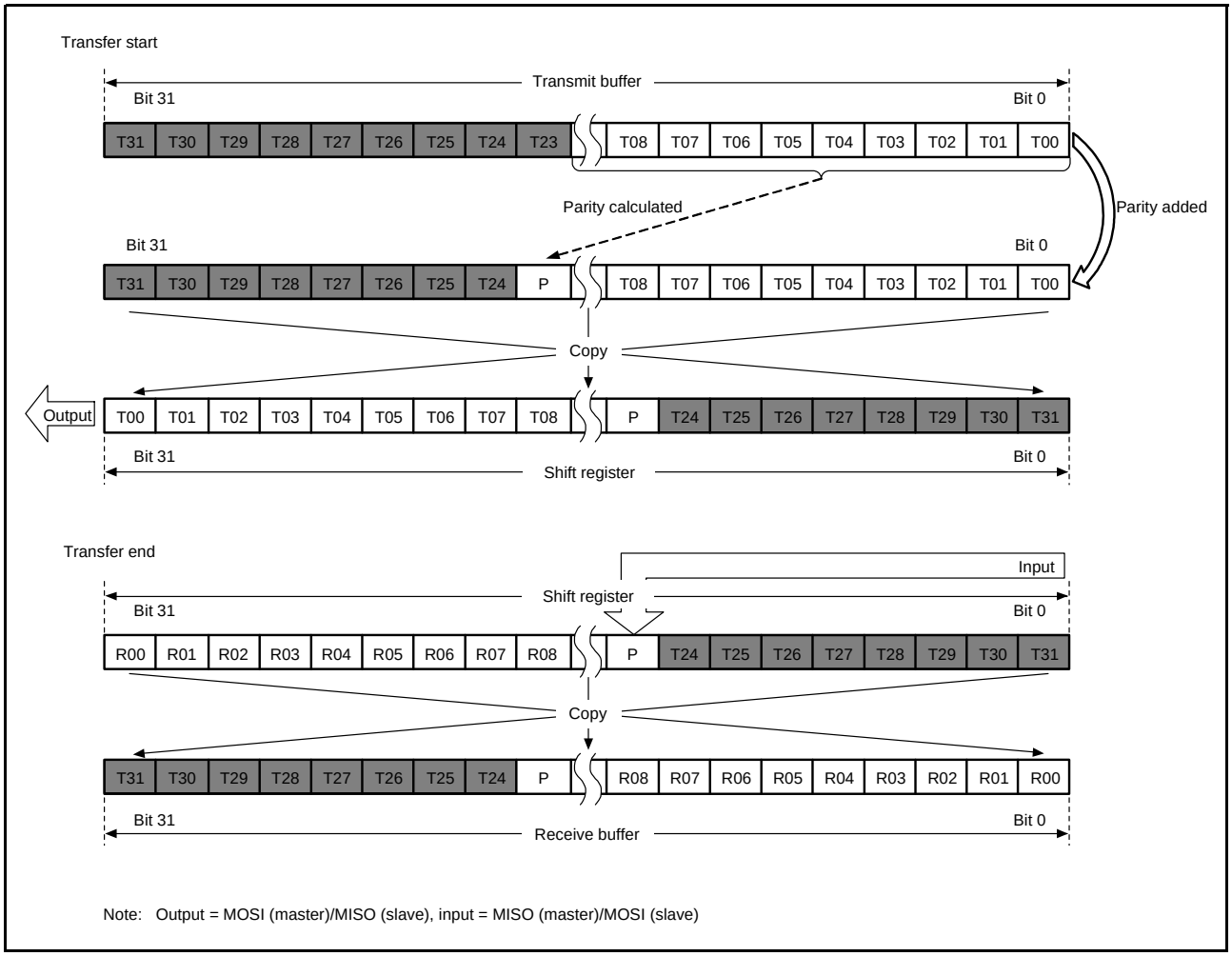


Figure 44.21 LSB First Transfer (24-Bit Data, Parity Enabled)

44.3.5 Transfer Format

44.3.5.1 CPHA = 0

Figure 44.22 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 0. Note that clock synchronous operation (the SPCR.SPMS bit is 1) should not performed when the RSPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 44.22, RSPCKA (CPOL = 0) indicates the RSPCKA signal waveform when the SPCMDm.CPOL bit is 0; RSPCKA (CPOL = 1) indicates the RSPCKA signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPI settings. For details, refer to section 44.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIA and MISOA signals commences at an SSLAi signal assertion timing. The first RSPCKA signal change timing that occurs after the SSLAi signal assertion becomes the first transfer data fetch timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSIA and MISOA signals is 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCKA signal operation timing; it only affects the signal polarity.

t1 denotes a period from an SSLAi signal assertion to RSPCKA oscillation (RSPCK delay). t2 denotes a period from the termination of RSPCKA oscillation to an SSLAi signal negation (SSL negation delay). t3 denotes a period in which SSLAi signal assertion is suppressed for the next transfer after the end of serial transfer (next-access delay). t1, t2, and t3 are controlled by a master device running on the RSPI system. For a description of t1, t2, and t3 when the RSPI of this MCU is in master mode, refer to section 44.3.10.1, Master Mode Operation.

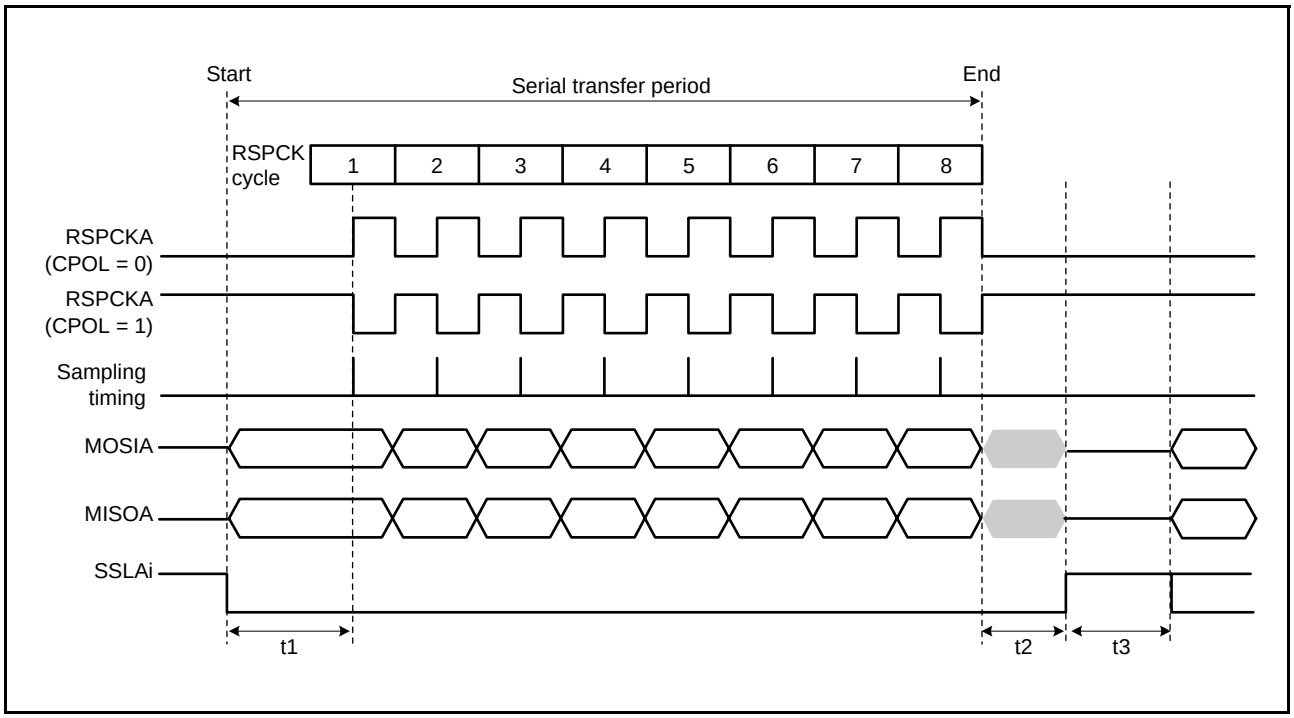


Figure 44.22 RSPI Transfer Format (CPHA = 0)

44.3.5.2 CPHA = 1

Figure 44.23 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSLAi signals are not used, and only the three signals RSPCKA, MOSIA, and MISOA handle communications. In Figure 44.23, RSPCK (CPOL = 0) indicates the RSPCKA signal waveform when the SPCMDm.CPOL bit is 0; RSPCKA (CPOL = 1) indicates the RSPCKA signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPI mode (master or slave). For details, refer to section 44.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISOA signal commences at an SSLAi signal assertion timing. The output of valid data to the MOSIA and MISOA signals commences at the first RSPCKA signal change timing that occurs after the SSLAi signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKA signal operation timing; it only affects the signal polarity.

t1, t2, and t3 are the same as those in the case of CPHA = 0. For a description of t1, t2, and t3 when the RSPI of this MCU is in master mode, refer to section 44.3.10.1, Master Mode Operation.

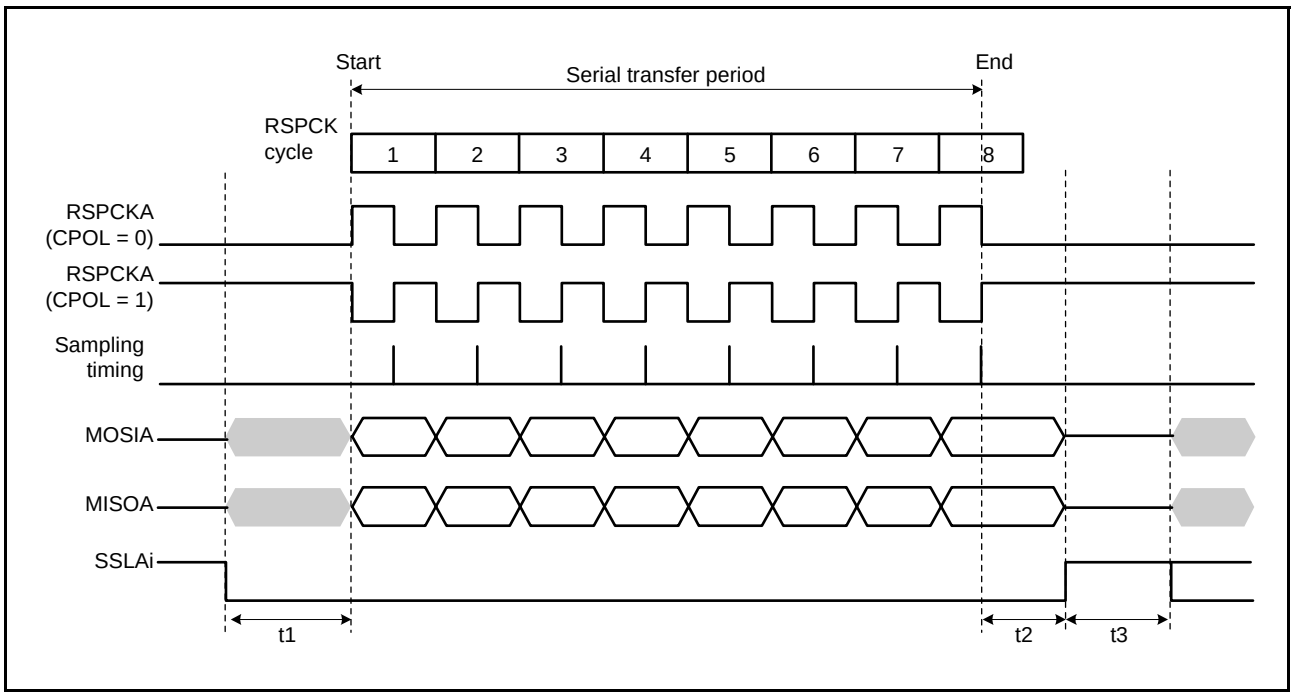


Figure 44.23 RSPI Transfer Format (CPHA = 1)

44.3.6 Communications Operating Mode

Full-duplex synchronous serial communications or transmit operations only can be selected by the communications operating mode select bit (SPCR.TXMD). The SPDR access shown in Figure 44.24 and Figure 44.25 indicate the condition of access to the SPDR register, where W denotes a write cycle.

44.3.6.1 Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0)

Figure 44.24 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 0. In the example in Figure 44.24, the RSPI performs an 8-bit serial transfer in which the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

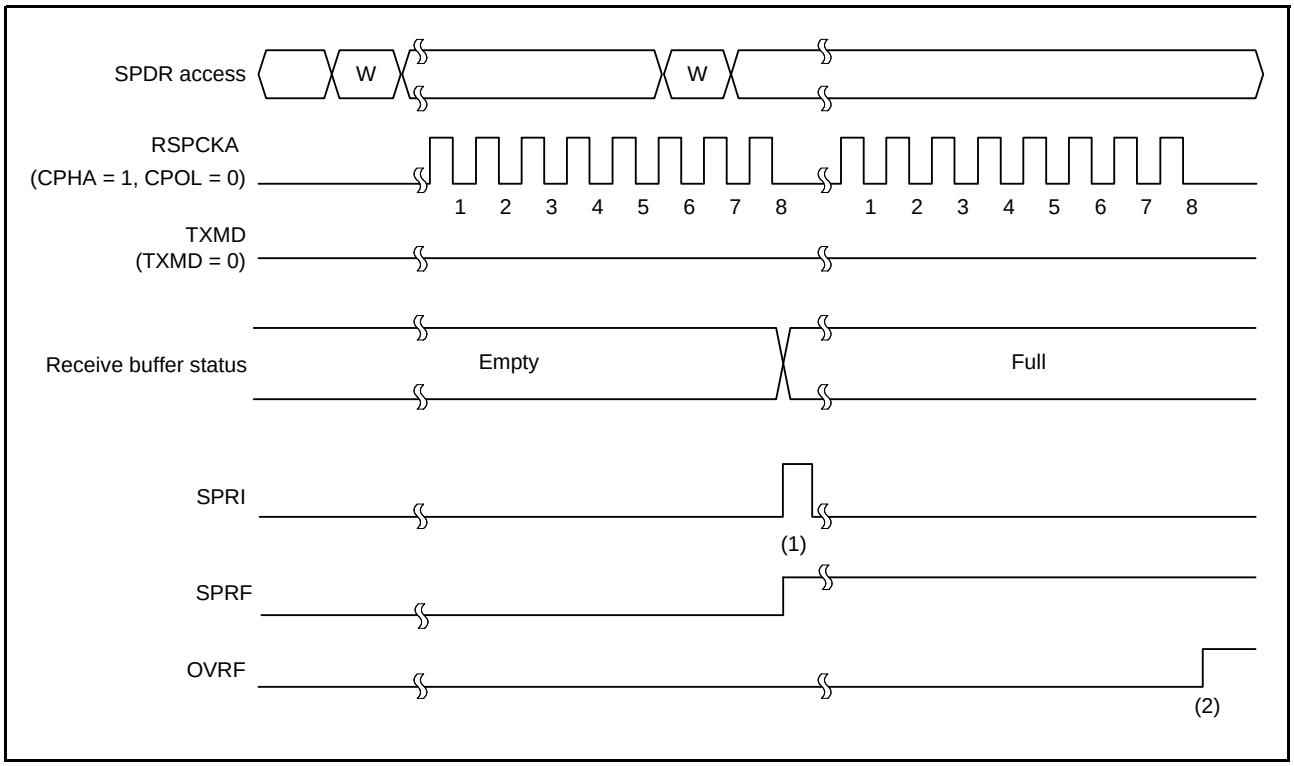


Figure 44.24 Operation Example of SPCR.TXMD = 0

The operation of the flags at timings shown in steps (1) and (2) in the figure is described below.

(1) When a serial transfer ends with the receive buffer of SPDR empty, the RSPI generates a receive buffer full interrupt request (SPRI) (sets the SPSR.SPRF flag to 1) and copies the received data in the shift register to the receive buffer.

(2) When a serial transfer ends with the receive buffer of SPDR holding data that was received in the previous serial transfer, the RSPI sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

When full-duplex synchronous serial communications (SPCR.TXMD = 0) is selected, reception occurs simultaneously with transmit operations. As such, the SPRF and OVRF flags in the SPSR register become 1 at the timing described in (1) and (2), respectively, according to the state of the receive buffer.

44.3.6.2 Transmit Operations Only (SPCR.TXMD = 1)

Figure 44.25 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 1. In the example in Figure 44.25, the RSPI performs an 8-bit serial transfer in which the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

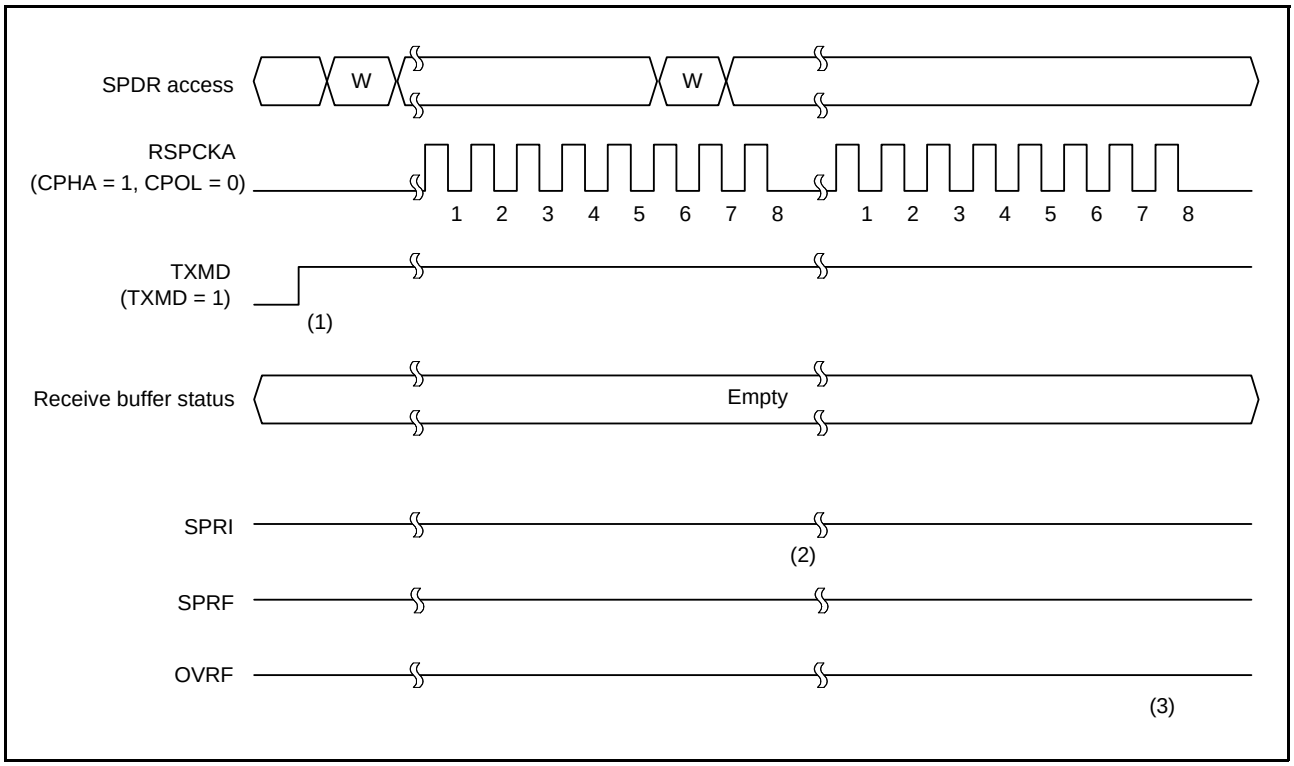


Figure 44.25 Operation Example of SPCR.TXMD = 1

The operation of the flags at timings shown in steps (1) to (3) in the figure is described below.

(1) Make sure there is no data left in the receive buffer and the SPSR.SPRF, OVRF flags are 0 before entering the mode of transmit operations only (SPCR.TXMD = 1).

(2) When a serial transfer ends with the receive buffer of SPDR empty, if the mode of transmit operations only is selected (SPCR.TXMD = 1), the SPRF flag remains 0 and the RSPI does not copy the data from the shift register to the receive buffer.

(3) Since the receive buffer of SPDR does not hold data that was received in the previous serial transfer, even when a serial transfer ends, the SPSR.OVRF flag retains the value of 0, and the data in the shift register is not copied to the receive buffer.

When performing transmit operations only (SPCR.TXMD = 1), the RSPI transmits data but does not receive data. Therefore, the SPSR.SPRF, OVRF flags remain 0 at the timings of (1) to (3).

44.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts

Figure 44.26 shows an example of operation of the transmit buffer empty interrupt (SPTI) and the receive buffer full interrupt (SPRI). The SPDR register access shown in Figure 44.26 indicates the condition of access to the SPDR register, where W denotes a write cycle, and R a read cycle. In the example in Figure 44.26, the RSPI performs an 8-bit serial transfer in which the SPCR.TXMD bit is 0, the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

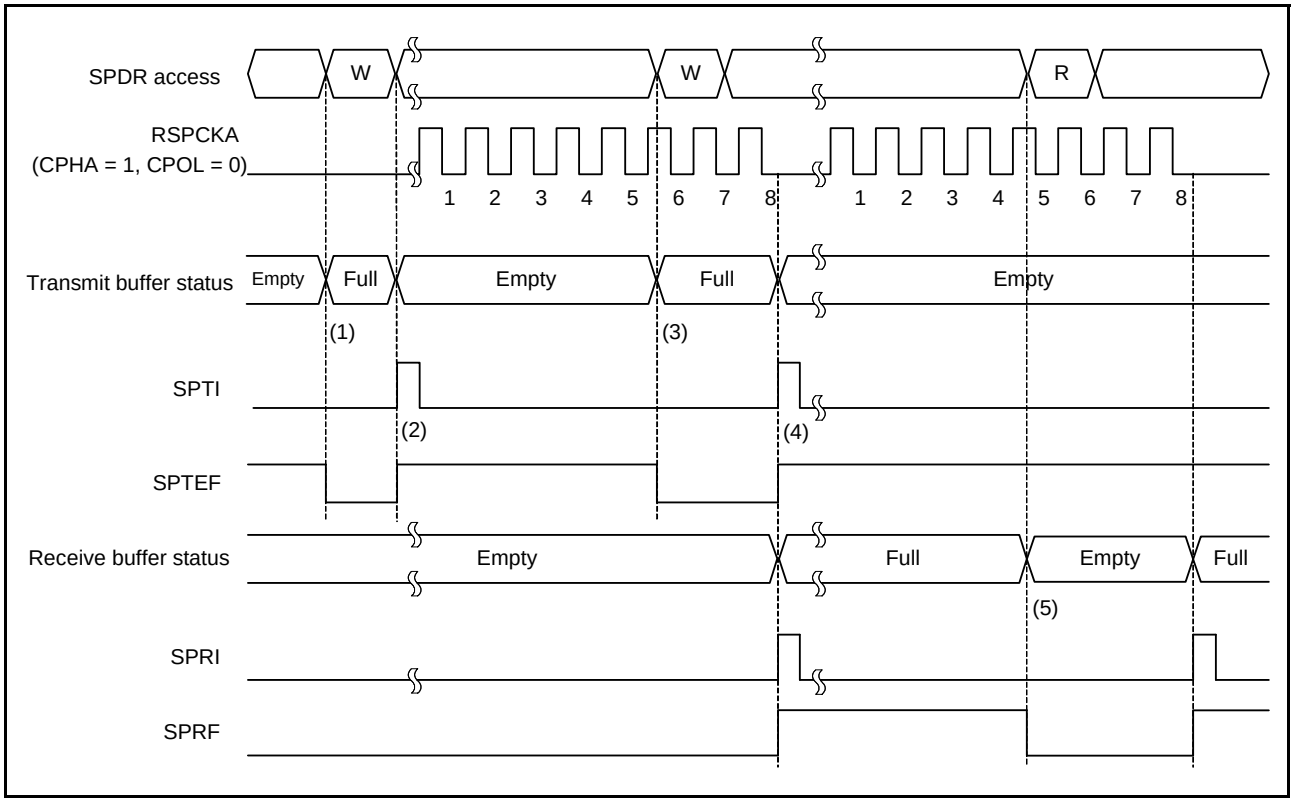


Figure 44.26 Operation Example of SPTI and SPRI Interrupts

The operation of the interrupts at timings shown in steps (1) to (5) in the figure is described below.

(1) When transmit data is written to SPDR when the transmit buffer of SPDR is empty (data for the next transfer is not set), the RSPI writes data to the transmit buffer and sets the SPSR.SPTEF flag to 0.

(2) If the shift register is empty, the RSPI copies the data from the transmit buffer to the shift register and generates a transmit buffer empty interrupt request (SPTI) and sets the SPSR.SPTEF flag to 1. How a serial transfer is started depends on the mode of the RSPI. For details, refer to section 44.3.10, SPI Operation, and section 44.3.11, Clock Synchronous Operation.

(3) When transmit data is written to SPDR in the transmit buffer empty interrupt routine or in the transmit buffer empty detecting process by polling the SPTEF flag, the data is transferred to the transmit buffer and the SPSR.SPTEF flag becomes 0. Because the data being transmitted is stored in the shift register, the RSPI does not copy the data from the transmit buffer to the shift register.

(4) When the serial transfer ends with the receive buffer of SPDR being empty, the RSPI copies the receive data from the shift register to the receive buffer, generates a receive buffer full interrupt request (SPRI), and sets the SPSR.SPRF flag to 1. Since the shift register becomes empty upon completion of serial transfer, when the transmit buffer had been full before the serial transfer ended, the RSPI sets the SPSR.SPTEF flag to 1 and copies the data from the transmit buffer to the shift register. Even when received data is not copied from the shift register to the receive buffer in an overrun error status, upon completion of the serial transfer, the RSPI determines that the shift register is empty, thus data transfer from the transmit buffer to the shift register is enabled.

(5) When SPDR is read in the receive buffer full interrupt routine or in the receive buffer full detecting process by polling the SPRF flag, the receive data can be read. When the receive data is read, the SPRF flag becomes 0.

If transmit data is written to SPDR while the transmit buffer holds data that has not yet been transmitted (the SPTEF flag is 0), the RSPI does not update the data in the transmit buffer. Transmit data should be written to SPDR in the transmit buffer empty interrupt request routine or in the transmit buffer empty detecting process by polling the SPTEF flag. To use a transmit buffer empty interrupt, set the SPTIE bit in SPCR to 1.
When setting the SPCR.SPE bit to 0 (RSPI disabled), the SPCR.SPTIE bit should also be set to 0. Otherwise (if the SPCR.SPE bit is 0 and the SPCR.SPTIE is 1), a transmit buffer empty interrupt request will occur.
When serial transfer ends with the receive buffer being full (the SPRF flag is 1), the RSPI does not copy data from the shift register to the receive buffer, and detects an overrun error (refer to section 44.3.8, Error Detection). To prevent a receive data overrun error, read the received data using a receive buffer full interrupt request before the next serial transfer ends. To use an RSPI receive buffer full interrupt, set the SPCR.SPRIE bit to 1.
Transmit and receive interrupts or the corresponding IRn.IR flags (where n is the interrupt vector number) in the ICU can be used to confirm the states of the transmit and receive buffers. Refer to section 15, Interrupt Controller (ICUA), for the interrupt vector numbers. The status of the transmit and receive buffers can be also confirmed by the SPTEF and SPRF flags.

44.3.8 Error Detection

In the normal RSPI serial transfer, the data written to the transmit buffer of SPDR is transmitted, and the received data can be read from the receive buffer of SPDR. If access is made to SPDR, depending on the status of the transmit/receive buffer or the status of the RSPI at the beginning or end of serial transfer, in some cases non-normal transfers can be executed.

If a non-normal transfer operation occurs, the RSPI detects the event as an overrun error, parity error, or mode fault error. Table 44.8 lists the relationship between non-normal transfer operations and the RSPI's error detection function.

Table 44.8 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function

	Occurrence Condition	RSPI Operation	Error Detection
1	SPDR is written when the transmit buffer is full.	• The contents of the transmit buffer are kept. • Missing write data.	None
2	SPDR is read when the receive buffer is empty.	Data received previously is output to the bus.	None
3	Serial transfer is started in slave mode when transmit data is still not loaded on the shift register.	Data received in previous serial transfer is transmitted.	None
4	Serial transfer terminates when the receive buffer is full.	• The contents of the receive buffer are kept. • Missing receive data.	Overrun error
5	An incorrect parity bit is received when performing full-duplex synchronous serial communications with the parity function enabled.	The parity error flag is asserted.	Parity error
6	The SSLA0 input signal is asserted when the serial transfer is idle in multi-master mode.	• Driving of the RSPCKA, MOSIA, SSLA1 to SSLA3 output signals is stopped. • RSPI function is disabled.	Mode fault error
7	The SSLA0 input signal is asserted during serial transfer in multi-master mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the RSPCKA, MOSIA, SSLA1 to SSLA3 output signals is stopped. • RSPI function is disabled.	Mode fault error
8	The SSLA0 input signal is negated during serial transfer in slave mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the MISOA output signal is stopped. • RSPI function is disabled.	Mode fault error

On operation 1 described in Table 44.8, the RSPI does not detect an error. To prevent data omission during the writing to SPDR, the SPDR register should be written when a transmit buffer empty interrupt request occurs or while the SPSR.SPTEF flag is 1.

Likewise, the RSPI does not detect an error on operation 2. To prevent extraneous data from being read, the SPDR register should be read when an RSPI receive buffer full interrupt request occurs or while the SPSR.SPRF flag is 1.

Similarly, the RSPI does not detect an error on operation 3. In a serial transfer that was started before the shift register was updated, the RSPI sends the data that was received in the previous serial transfer, and does not treat the operation indicated in 3 as an error. Note that the received data from the previous serial transfer is retained in the receive buffer of SPDR, thus it can be correctly read (if SPDR is not read before the end of the serial transfer, an overrun error may occur).

An overrun error shown in 4 is described in section 44.3.8.1, Overrun Error. A parity error shown in 5 is described in section 44.3.8.2, Parity Error. A mode fault error shown in 6 to 8 is described in section 44.3.8.3, Mode Fault Error. For the transmit and receive interrupts, refer to section 44.3.7, Transmit Buffer Empty/Receive Buffer Full Interrupts.

44.3.8.1 Overrun Error

If a serial transfer ends when the receive buffer of SPDR is full, the RSPI detects an overrun error, and sets the SPSR.OVRF flag to 1. When the OVRF flag is 1, the RSPI does not copy data from the shift register to the receive buffer so that the data prior to the occurrence of the error is retained in the receive buffer. To set the OVRF flag to 0, write 0 to the OVRF flag after the CPU has read SPSR with the OVRF flag set to 1.
Figure 44.27 shows an example of operations of the SPRF and OVRF flags. The SPSR and SPDR accesses shown in Figure 44.27 indicate the condition of accesses to SPSR and SPDR, respectively, where W denotes a write cycle, and R a read cycle. In the example in Figure 44.27, the RSPI performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

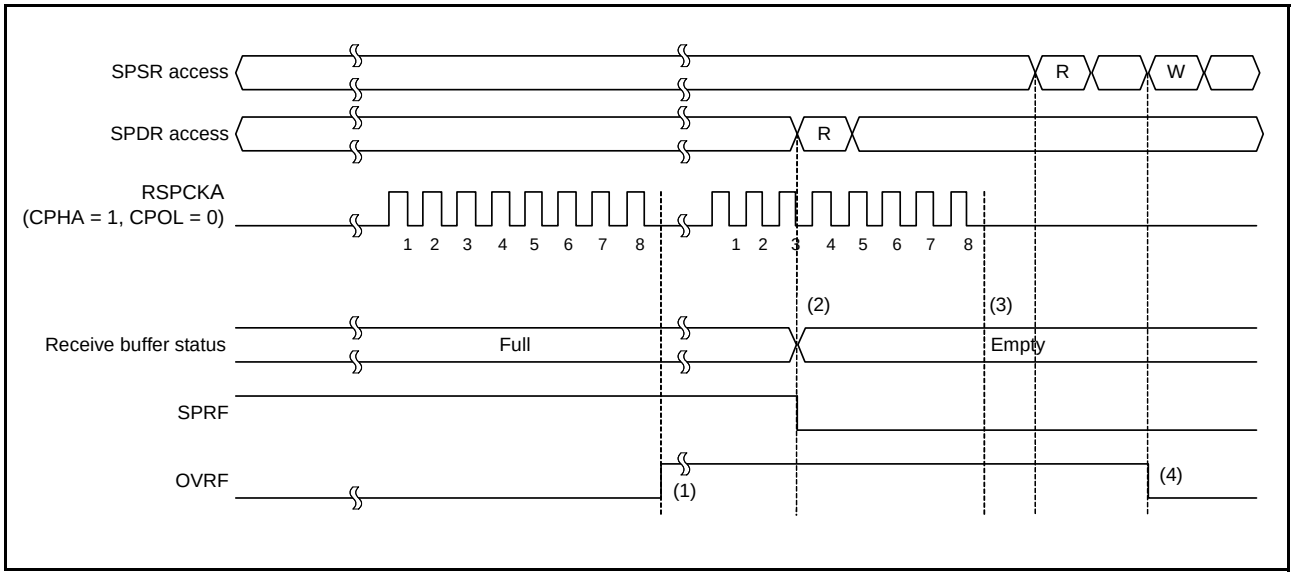


Figure 44.27 Operation Example of SPRF and OVRF Flags

The operation of the flags at the timing shown in steps (1) to (4) in the figure is described below.

(1) If a serial transfer terminates with the receive buffer full (the SPRF flag is 1), the RSPI detects an overrun error, and sets the OVRF flag to 1. The RSPI does not copy the data in the shift register to the receive buffer. Even if the SPPE bit is 1, parity errors are not detected. In master mode, the RSPI copies the pointer value to SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) When SPDR is read, the RSPI outputs the data in the receive buffer. At this time the SPRF flag becomes 0. Even if the receive buffer becomes empty, the OVRF flag does not become 0.

(3) If the serial transfer ends with the OVRF flag being 1 (an overrun error occurs), the RSPI does not copy the data in the shift register to the receive buffer (the SPRF flag remains 0). A receive buffer full interrupt is not generated. Even if the SPPE bit is 1, parity errors are not detected. When in master mode, the RSPI does not update the SPSSR.SPECM[2:0] bits. When in an overrun error state and the RSPI does not copy the received data from the shift register to the receive buffer, upon termination of the serial transfer, the RSPI determines that the shift register is empty; in this manner, data transfer from the transmit buffer to the shift register is enabled.

(4) If 0 is written to the OVRF flag after SPSR is read when the OVRF flag is 1, the OVRF flag is set to 0.

The occurrence of an overrun can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. When executing a serial transfer, measures should be taken to ensure the early detection of overrun errors, such as reading SPSR immediately after SPDR is read. When the RSPI is used in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.
If an overrun error occurs and the OVRF flag is set to 1, normal reception operations cannot be performed until the OVRF flag is set to 0.

When the RSPCK auto-stop function is enabled in master mode, an overrun error does not occur. Figure 44.28 and Figure 44.29 show the clock stop waveform when a serial transfer continues while the receive buffer is full in master mode.

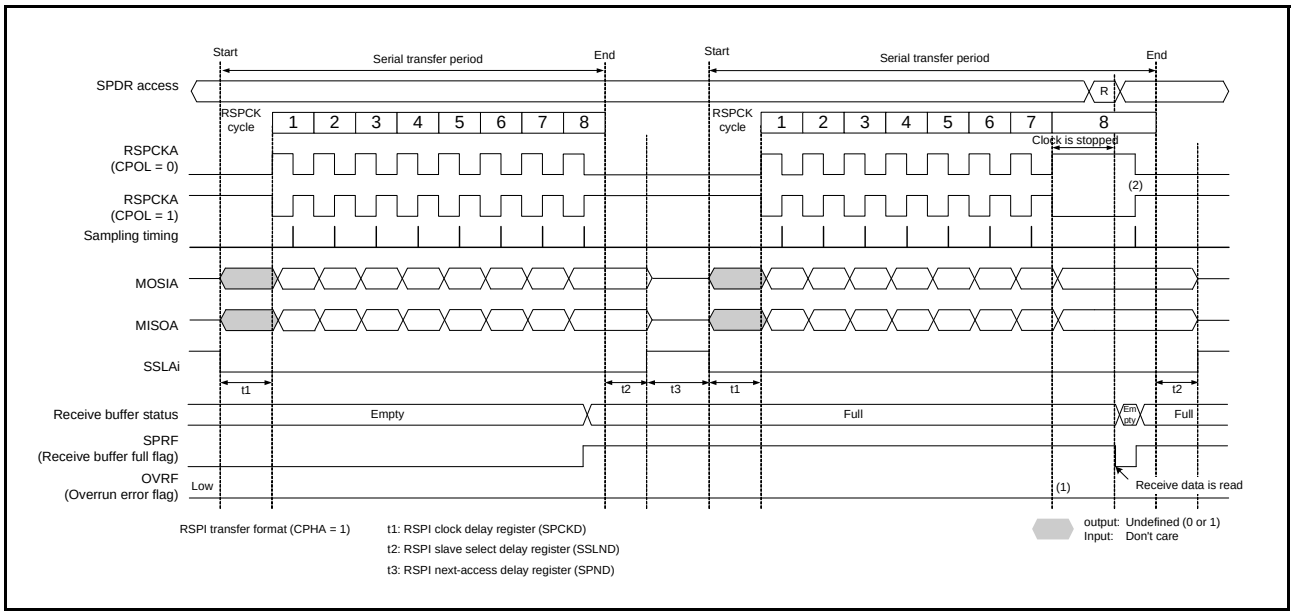


Figure 44.28 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 1)

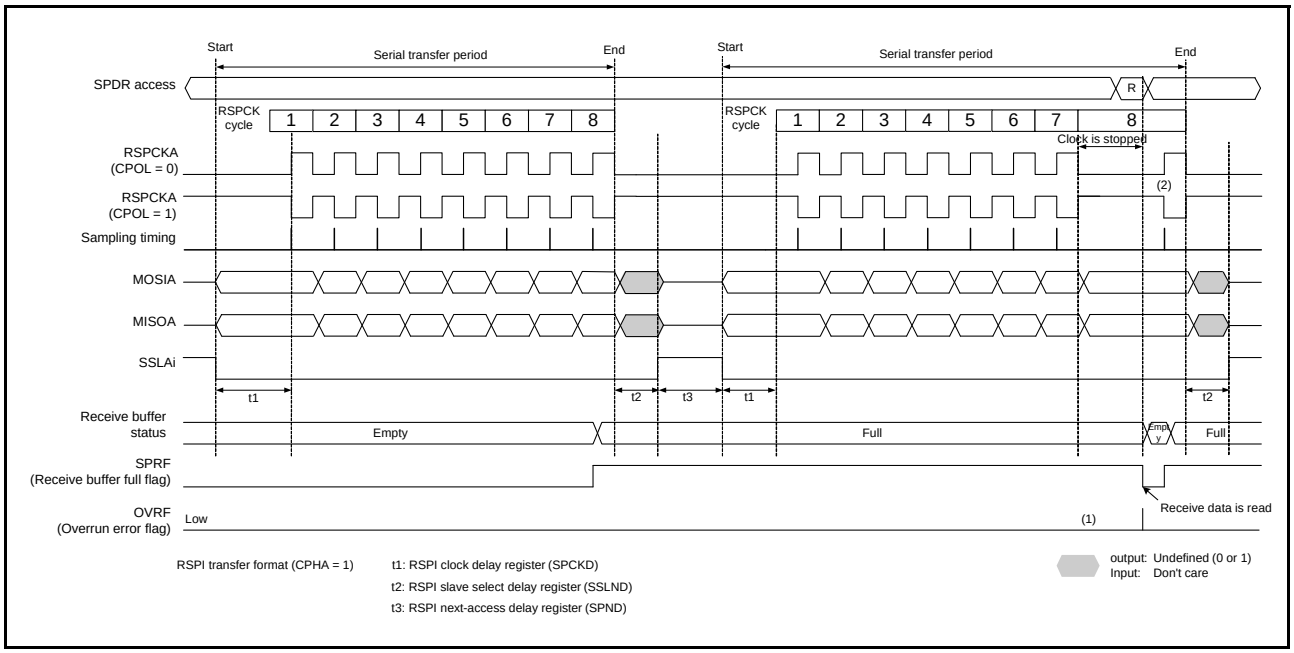


Figure 44.29 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 0)

The operation of the flags at the timings shown in steps (1) and (2) in the figure is described below.

(1) When the receive buffer is full, an overrun error does not occur because the RSPCK clock is stopped.

(2) If SPDR is read while the clock is stopped, data in the receive buffer can be read. The RSPCK clock restarts after reading the receive buffer (after the SPRF flag becomes 0).

44.3.8.2 Parity Error

If full-duplex synchronous serial communications is performed with the SPCR.TXMD bit set to 0 and the SPCR2.SPPE bit set to 1, when serial transfer ends, the RSPI checks whether there are parity errors. Upon detecting a parity error in the received data, the RSPI sets the SPSR.PERF flag to 1. Since the RSPI does not copy the data in the shift register to the receive buffer when the SPSR.OVRF flag is set to 1, parity error detection is not performed for the received data. To set the PERF flag to 0, write 0 to the PERF flag after SPSR register is read with the PERF flag set to 1.
Figure 44.30 shows an example of operation of the OVRF and PERF flags. The SPSR access shown in Figure 44.30 indicates the condition of access to SPSR register, where W denotes a write cycle, and R a read cycle. In the example of Figure 44.30, full-duplex synchronous serial communications is performed while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1. The RSPI performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKA waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

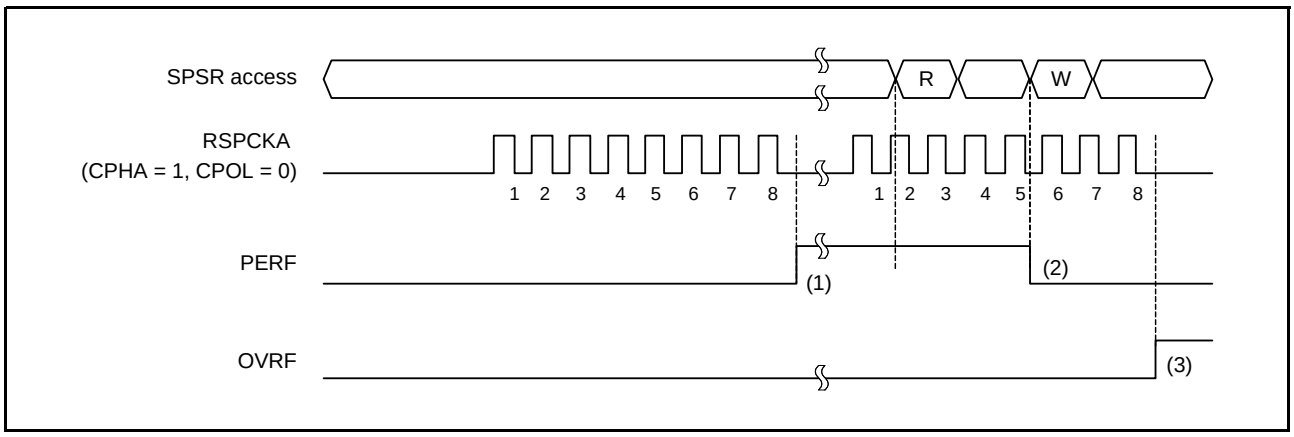


Figure 44.30 Operation Example of PERF Flag

The operation of the flags at the timing shown in steps (1) to (3) in the figure is described below.

(1) If a serial transfer terminates with the RSPI not detecting an overrun error, the RSPI copies the data in the shift register to the receive buffer. The RSPI judges the received data at this timing, and sets the PERF flag to 1 if a parity error is detected. In master mode, the RSPI copies the pointer value to SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) If 0 is written to the PERF flag after SPSR register is read when the PERF flag is 1, the PERF flag is set to 0.

(3) When the RSPI detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The RSPI does not perform parity error detection at this timing.

The occurrence of a parity error can be checked either by reading the SPSR register or by using an RSPI error interrupt and reading the SPSR register. When executing a serial transfer, measures should be taken to ensure the early detection of parity errors, such as reading SPSR. When the RSPI is used in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

44.3.8.3 Mode Fault Error

The RSPI operates in multi-master mode when the SPCR.MSTR bit is 1, the SPCR.SPMS bit is 0, and the SPCR.MODFEN bit is 1. If the active level is input with respect to the SSLA0 input signal of the RSPI in multi-master mode, the RSPI detects a mode fault error irrespective of the status of the serial transfer, and sets the SPSR.MODF flag to 1. Upon detecting the mode fault error, the RSPI copies the value of the pointer to SPCMDm to the SPSSR.SPECM[2:0] bits. The active level of the SSLA0 signal is determined by the SSLP.SSL0P bit.

When the MSTR bit is 0, the RSPI operates in slave mode. The RSPI detects a mode fault error if the MODFEN bit of the RSPI in slave mode is 1, and the SPMS bit is 0, and if the SSLA0 input signal is negated during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched).

Upon detecting a mode fault error, the RSPI stops driving of the output signals and clears the SPCR.SPE bit to 0 (refer to section 44.3.9, Initializing RSPI). In the case of multi-master configuration, detection of a mode fault error is used to stop driving of the output signals and the RSPI function, which allows the master right to be released.

The occurrence of a mode fault error can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. Detecting mode fault errors without utilizing the RSPI error interrupt requires polling of SPSR. When using the RSPI in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

When the MODF flag is 1, writing of the value 1 to the SPE bit is ignored by the RSPI. To enable the RSPI function after the detection of a mode fault error, set the MODF flag to 0.

44.3.9 Initializing RSPI

If 0 is written to the SPCR.SPE bit or the RSPI sets the SPE bit to 0 because of the detection of a mode fault error, the RSPI disables the RSPI function, and initializes some of the module functions. When a system reset is generated, the RSPI initializes all of the module functions. The following describes initialization by the clearing of the SPCR.SPE bit and initialization by a system reset.

44.3.9.1 Initialization by Clearing the SPE Bit

When the SPCR.SPE bit is set to 0, the RSPI performs the following initialization:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals (Hi-Z) in slave mode
- Initializing the internal state of the RSPI
- Initializing the transmit buffer of the RSPI (Set the SPTEF flag to 1)

Initialization by the clearing of the SPE bit does not initialize the control bits of the RSPI. For this reason, the RSPI can be started in the same transfer mode as prior to the initialization if the SPE bit is set to 1 again.
The SPSR.SPRF, SPSR.OVRF, SPSR.MODF, and SPSR.PERF flags are not initialized, nor is the value of the RSPI sequence status register (SPSSR) initialized. For this reason, even after the RSPI is initialized, data from the receive buffer can be read in order to check the status of error occurrence during an RSPI transfer.
The transmit buffer is initialized to an empty state (the SPTEF flag is 1). Therefore, if the SPCR.SPTIE bit is set to 1 after RSPI initialization, a transmit buffer empty interrupt is generated. When the RSPI is initialized, in order to disable any transmit buffer empty interrupt, 0 should be written to the SPTIE bit simultaneously with the writing of 0 to the SPE bit.

44.3.9.2 System Reset

The initialization by a system reset completely initializes the RSPI through the initialization of all bits for controlling the RSPI, initialization of the status bits, and initialization of data registers, in addition to the requirements described in section 44.3.9.1, Initialization by Clearing the SPE Bit.

44.3.10 SPI Operation

44.3.10.1 Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (refer to section 44.3.8, Error Detection). When operating in single-master mode, the RSPI does not detect mode fault errors whereas the RSPI running in multi-master mode does detect mode fault errors. This section explains operations that are common to single-master mode and multi-master mode.

(1) Starting a Serial Transfer

The RSPI updates the data in the transmit buffer (SPTX) when data is written to the RSPI data register (SPDR) with the RSPI transmit buffer being empty (the SPTEF flag is 1 and data for the next transfer is not set). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR, the RSPI copies data from the transmit buffer to the shift register and starts serial transfer. Upon copying transmit data to the shift register, the RSPI changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.
For details on the RSPI transfer format, refer to section 44.3.5, Transfer Format. The polarity of the SSLAi output pins depends on the SSLP register settings.

(2) Terminating a Serial Transfer

Irrespective of the SPCMDm.CPHA bit, the RSPI terminates the serial transfer after transmitting an RSPCKA edge corresponding to the final sampling timing. If free space is available in the receive buffer (SPRX) (the SPRF flag is 0), upon termination of serial transfer, the RSPI copies data from the shift register to the receive buffer of the SPDR register. It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the SPCMDm.SPB[3:0] bit setting. The polarity of the SSLAi output pin depends on the SSLP register settings.
For details on the RSPI transfer format, refer to section 44.3.5, Transfer Format.

(3) Sequence Control

The transfer format that is employed in master mode is determined by SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers.

SPSCR is a register used to determine the sequence configuration for serial transfers that are executed by the RSPI in master mode. The following items are set in SPCMDm register: SSLAi pin output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCK polarity/phase, whether SPCKD is to be referenced, whether SSLND is to be referenced, and whether SPND is to be referenced. SPBR holds some of the bit rate settings; SPCKD, an RSPI clock delay value; SSLND, an SSL negation delay; and SPND, a next-access delay value.

According to the sequence length that is assigned to SPSCR, the RSPI makes up a sequence comprised of a part or all of SPCMDm register. The RSPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. The RSPI increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, the RSPI sets the pointer in SPCMD0, and in this manner the sequence is executed repeatedly.

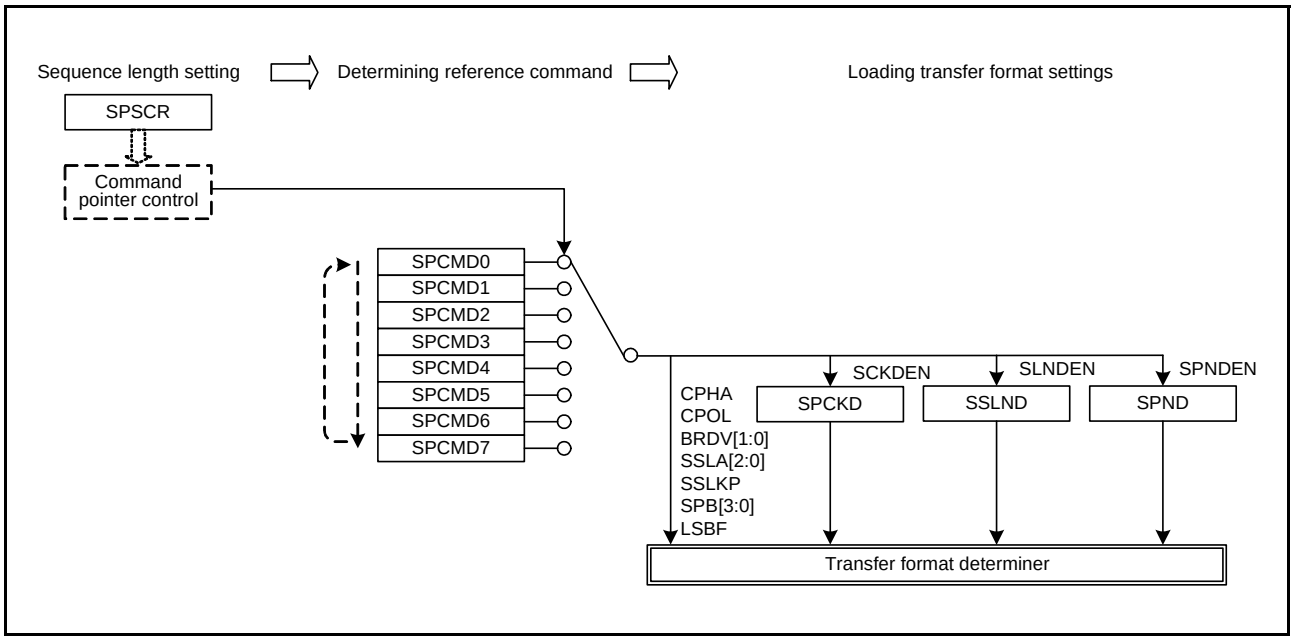


Figure 44.31 Procedure for Determining the Form of Serial Transfer in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

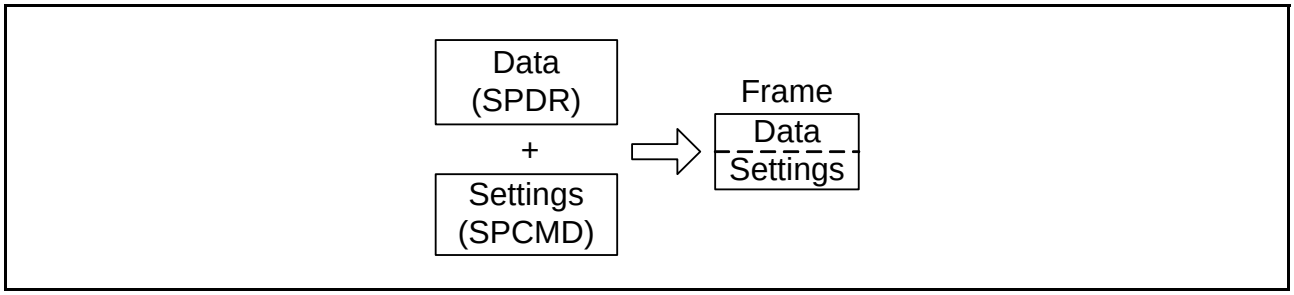


Figure 44.32 Concept of a Frame

Figure 44.33 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 44.4.

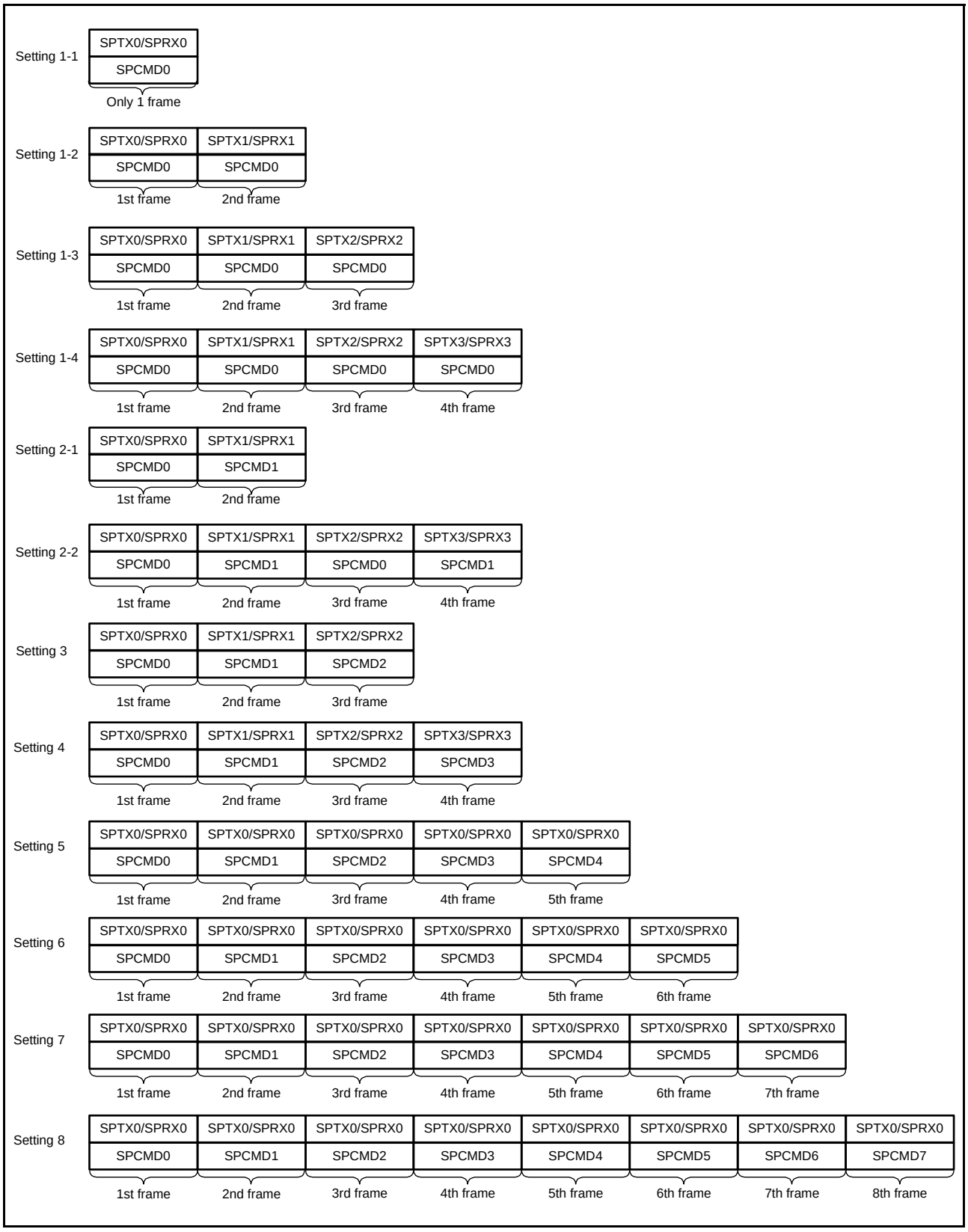


Figure 44.33 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(4) Burst Transfer

If the SPCMDm.SSLKP bit that the RSPI references during the current serial transfer is 1, the RSPI keeps the SSLAi signal level during the serial transfer until the beginning of the SSLAi signal assertion for the next serial transfer. If the SSLAi signal level for the next serial transfer is the same as the SSLAi signal level for the current serial transfer, the RSPI can execute continuous serial transfers while keeping the SSLAi signal assertion status (burst transfer).
Figure 44.34 shows an example of an SSLAi signal operation for the case where a burst transfer is implemented using SPCMD0 and SPCMD1 register settings. The text below explains the RSPI operations (1) to (7) as shown in Figure 44.34. It should be noted that the polarity of the SSLAi output signal depends on the SSLP register settings.

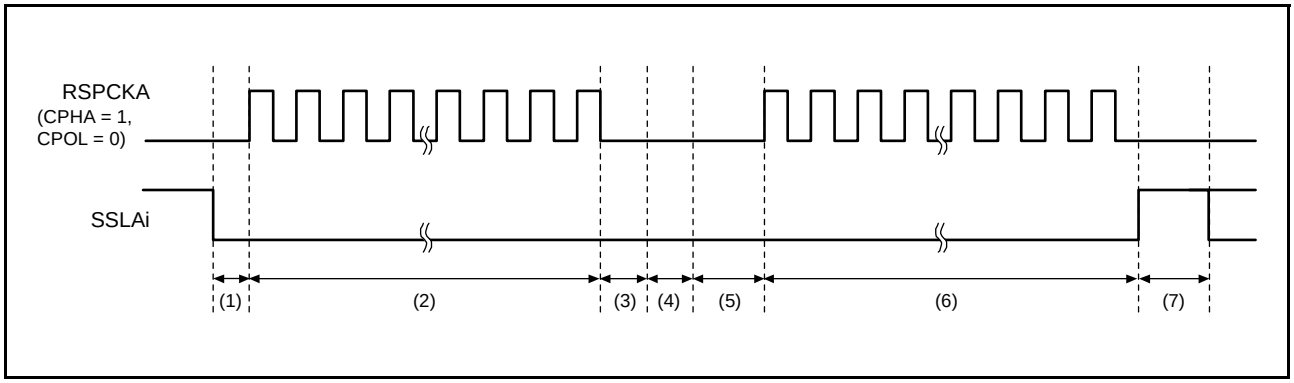


Figure 44.34 Example of Burst Transfer Operation Using SSLKP Bit

(1) Based on SPCMD0, the RSPI asserts the SSLAi signal and inserts RSPCK delays.
(2) The RSPI executes serial transfers according to SPCMD0.
(3) The RSPI inserts SSL negation delays.
(4) Since the SPCMD0.SSLKP bit is 1, the RSPI keeps the SSLAi signal value on SPCMD0. This period is sustained, at the shortest, for a period equal to the next-access delay of SPCMD0. If the shift register is empty after the passage of a minimum period, this period is sustained until the transmit data is stored in the shift register for the next transfer.
(5) Based on SPCMD1, the RSPI asserts the SSLAi signal and inserts RSPCK delays.
(6) The RSPI executes serial transfers according to SPCMD1.
(7) Because the SPCMD1.SSLKP bit is 0, the RSPI negates the SSLAi signal. In addition, a next-access delay is inserted according to SPCMD1.

If the SSLAi signal output settings in the SPCMDm register in which 1 is assigned to the SSLKP bit are different from the SSLAi signal output settings in the SPCMDm register to be used in the next transfer, the RSPI switches the SSLAi signal status to SSLAi signal assertion ((5) in Figure 44.34) corresponding to the command for the next transfer. Note that if such an SSLAi signal switching occurs, the slaves that drive the MISOA signal compete, and collision of signal levels may occur.
The RSPI in master mode references the SSLAi signal operation within the module for the case where the SSLKP bit is not used. Even when the SPCMDm.CPHA bit is 0, the RSPI can accurately start serial transfers by using the SSLAi signal assertion for the next transfer that is detected internally.

(5) RSPCK Delay (t1)

The RSPCK delay value of the RSPI in master mode depends on the SPCMDm.SCKDEN bit setting and the SPCKD register setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines an RSPCK delay value during serial transfer by using the SPCMDm.SCKDEN bit and SPCKD, as listed in Table 44.9. For a definition of RSPCK delay, refer to section 44.3.5, Transfer Format.

Table 44.9 Relationship among SCKDEN Bit, SPCKD, and RSPCK Delay Value

SPCMDm.SCKDEN Bit	SPCKD.SCKDL[2:0] Bits	RSPCK Delay Value
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(6) SSL Negation Delay (t2)

The SSL negation delay value of the RSPI in master mode depends on the SPCMDm.SLNDEN bit setting and the SSLND register setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines an SSL negation delay value during serial transfer by using the SPCMDm.SLNDEN bit and SSLND, as listed in Table 44.10. For a definition of SSL negation delay, refer to section 44.3.5, Transfer Format.

Table 44.10 Relationship among SLNDEN Bit, SSLND, and SSL Negation Delay Value

SPCMDm.SLNDEN Bit	SSLND.SLNDL[2:0] Bits	SSL Negation Delay Value
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(7) Next-Access Delay (t3)

The next-access delay value of the RSPI in master mode depends on the SPCMDm.SPNDEN bit setting and the SPND setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines a next-access delay value during serial transfer by using the SPCMDm.SPNDEN bit and SPND, as listed in Table 44.11. For a definition of next-access delay, refer to section 44.3.5, Transfer Format.

Table 44.11 Relationship among SPNDEN Bit, SPND, and Next-Access Delay Value

SPCMDm.SPNDEN Bit	SPND.SPNDL[2:0] Bits	Next-Access Delay Value
0	000b to 111b	1 RSPCK + 2 PCLK
1	000b	1 RSPCK + 2 PCLK
	001b	2 RSPCK + 2 PCLK
	010b	3 RSPCK + 2 PCLK
	011b	4 RSPCK + 2 PCLK
	100b	5 RSPCK + 2 PCLK
	101b	6 RSPCK + 2 PCLK
	110b	7 RSPCK + 2 PCLK
	111b	8 RSPCK + 2 PCLK

(8) Initialization Flowchart

Figure 44.35 is a flowchart illustrating an example of initialization in SPI operation when the RSPI is used in master mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

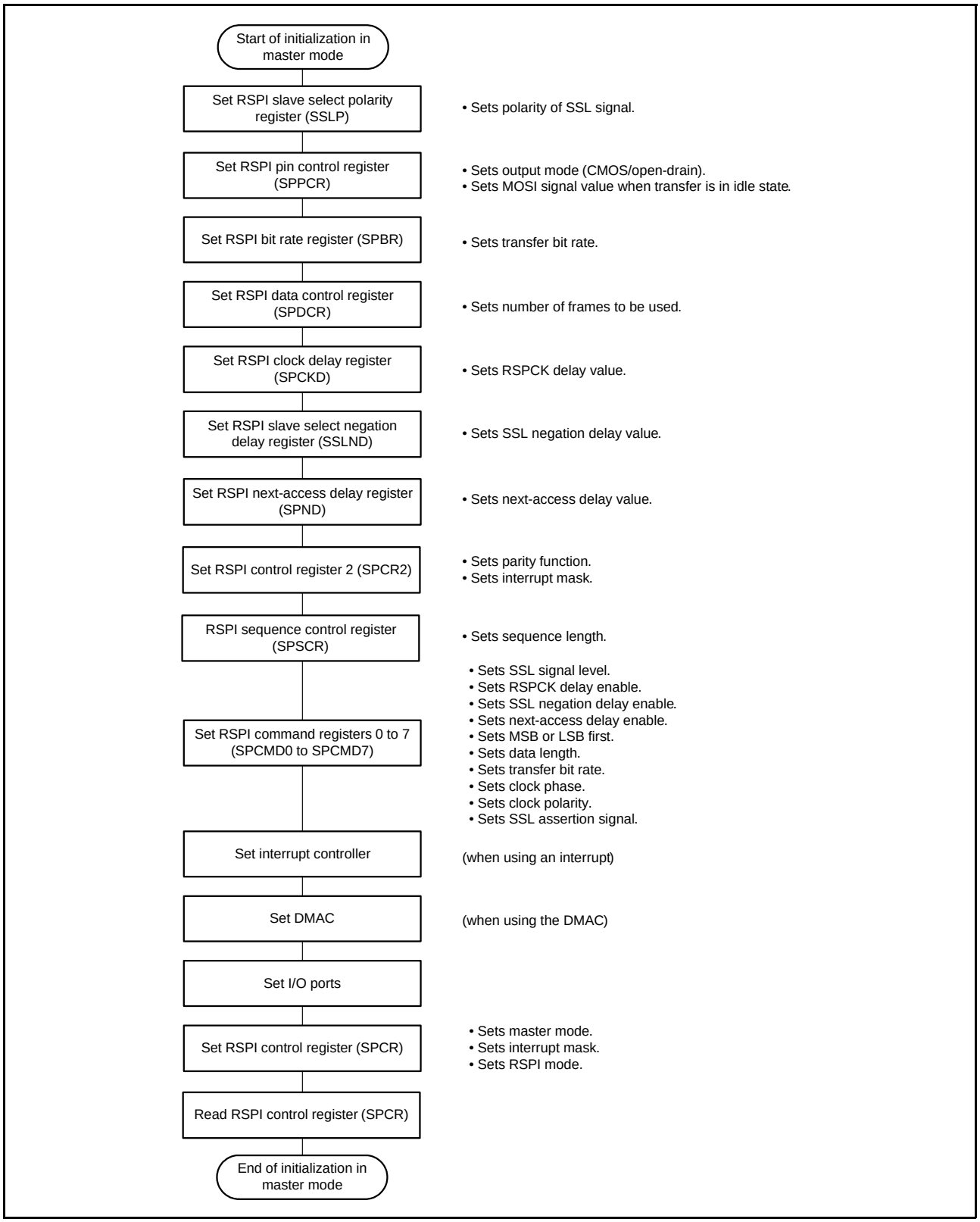


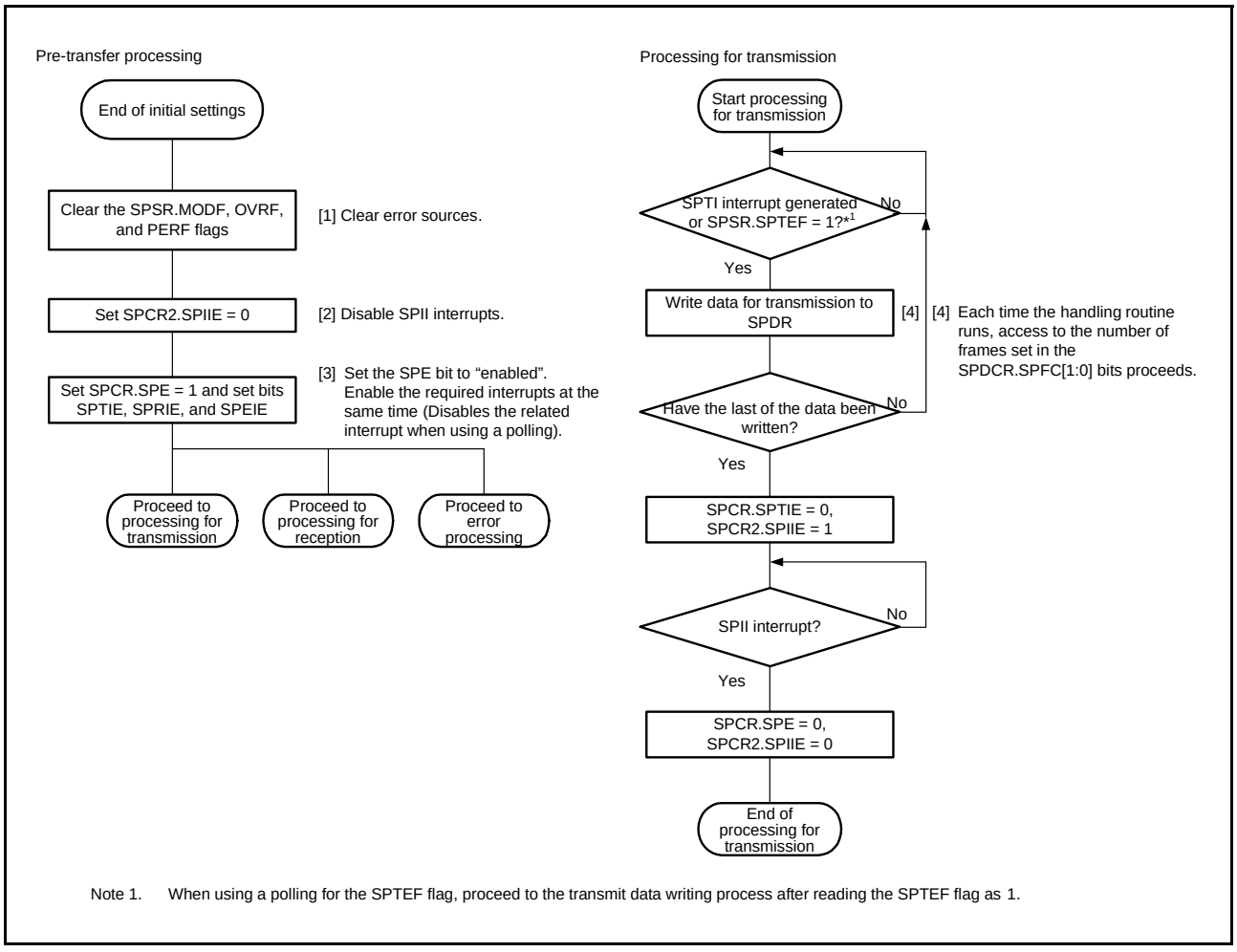
Figure 44.35 Example of Initialization Flowchart in Master Mode (SPI Operation)

(9) Software Processing Flow

Figure 44.36 to Figure 44.38 show examples of the flow of software processing.

(a) Transmit Processing Flow

When transmitting data, the CPU will be notified of the completion of data transmission by enabling the SPI interrupt after the last writing of data for transmission.
The completion of data transmission can also be checked by polling to see if the SPSR.IDLNF flag has become 0, instead of using the SPII interrupt. However, one cycle of PCLK is required for the time from when data for transmission is written in the SPDR register to when the IDLNF flag becomes 1. After the last data is written in the SPDR register, discard the value of the SPSR register once not to judge the condition with the IDLNF flag which has not yet become 1, and read and use the value of the SPSR.IDLNF flag to confirm the completion of data transmission.



Note 1. When using a polling for the SPTEF flag, proceed to the transmit data writing process after reading the SPTEF flag as 1.

Figure 44.36 Flowchart in Master Mode (Transmission)

(b) Receive Processing Flow

The RSPI does not handle receive-only operation, so processing for transmission is required.

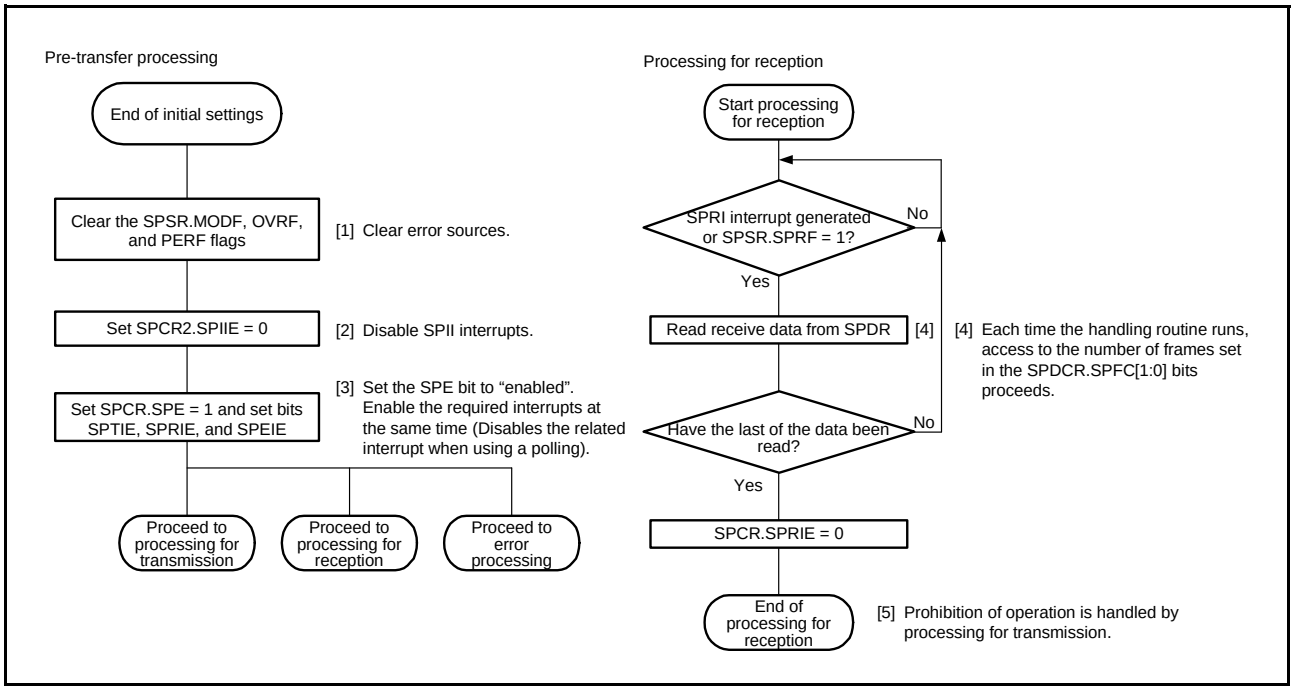


Figure 44.37 Flowchart in Master Mode (Reception)

(c) Flow of Error Processing

The RSPI has three types of error. When a mode fault error is generated, the SPCR.SPE bit is automatically cleared, stopping operations for transmission and reception. For errors from other sources, however, the SPCR.SPE bit is not cleared and operations for transmission and reception continue; accordingly, we recommend clearing of the SPCR.SPE bit to stop operations in the case of errors other than mode fault errors. Not doing so will lead to updating of the SPSSR.SPECM[2:0] bits.

When interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

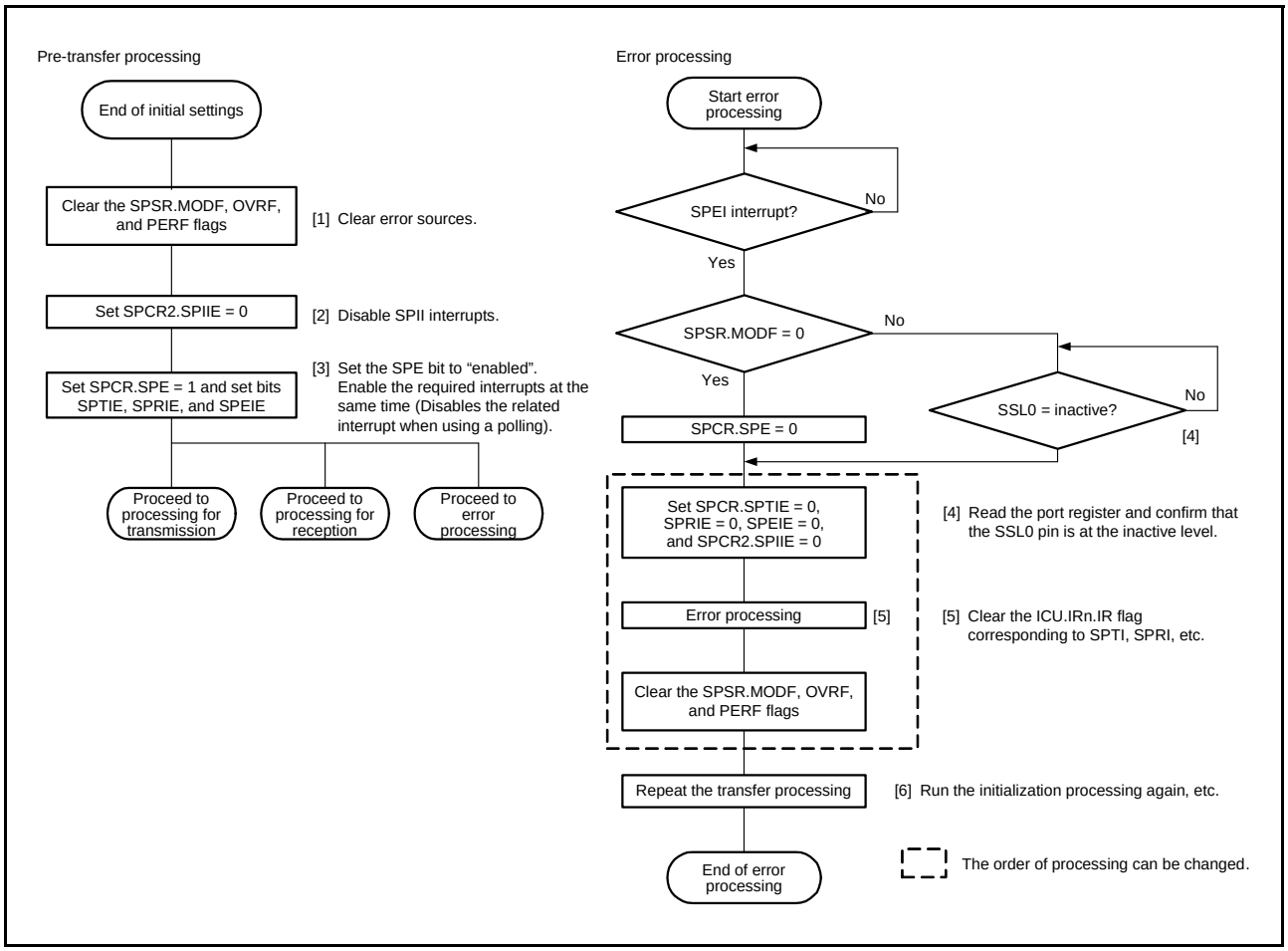


Figure 44.38 Flowchart for Master Mode (Error Processing)

44.3.10.2 Slave Mode Operation

(1) Starting a Serial Transfer

If the SPCMD0.CPHA bit is 0, when detecting an SSLA0 input signal assertion, the RSPI needs to start driving valid data to the MISOA output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLA0 input signal triggers the start of a serial transfer.

If the CPHA bit is 1, when detecting the first RSPCKA edge in an SSLA0 signal asserted condition, the RSPI needs to start driving valid data to the MISOA output signal. For this reason, when the CPHA bit is 1, the first RSPCKA edge in an SSLA0 signal asserted condition triggers the start of a serial transfer.

When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to “full”, so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI leaves the status of the shift register unchanged, in the full state.

Irrespective of the CPHA bit setting, the timing at which the RSPI starts driving of the MISOA output signal is the SSLA0 signal assertion timing. The data which is output by the RSPI is either valid or invalid, depending on the CPHA bit setting.

For details on the RSPI transfer format, refer to section 44.3.5, Transfer Format. The polarity of the SSLA0 input signal depends on the setting of the SSLP.SSL0P bit.

(2) Terminating a Serial Transfer

Irrespective of the SPCMD0.CPHA bit, the RSPI terminates the serial transfer after detecting an RSPCKA edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPRF flag is 0), upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty”, regardless of the receive buffer state. A mode fault error occurs if the RSPI detects an SSLA0 input signal negation from the beginning of serial transfer to the end of serial transfer (refer to section 44.3.8, Error Detection).

The final sampling timing changes depending on the bit length of transfer data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[3:0] bit setting. The polarity of the SSLA0 input signal depends on the SSLP.SSL0P bit setting.

For details on the RSPI transfer format, refer to section 44.3.5, Transfer Format.

(3) Notes on Single-Slave Operations

If the SPCMD0.CPHA bit is 0, the RSPI starts serial transfers when it detects the assertion edge for an SSLA0 input signal. In the type of configuration shown in Figure 44.7 as an example, if the RSPI is used in single-slave mode, the SSLA0 signal is fixed at the active state. Therefore, when the CPHA bit is set to 0, the RSPI cannot correctly start a serial transfer. To correctly execute transmit/receive operations by the RSPI in slave mode in a configuration in which the SSLA0 input signal is fixed at the active state, the CPHA bit should be set to 1. If there is a need for setting the CPHA bit to 0, the SSLA0 input signal should not be fixed.

(4) Burst Transfer

If the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSLA0 input signal. If the CPHA bit is 1, the period from the first RSPCKA edge to the sampling timing for the reception of the final bit in an SSLA0 signal active state corresponds to a serial transfer period. Even when the SSLA0 input signal remains at the active level, the RSPI can accommodate burst transfers because it can detect the start of an access.
If the CPHA bit is 0, the second and subsequent serial transfers during burst transfer cannot be executed correctly.

(5) Initialization Flowchart

Figure 44.39 is a flowchart illustrating an example of initialization in SPI operation when the RSPI is used in slave mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

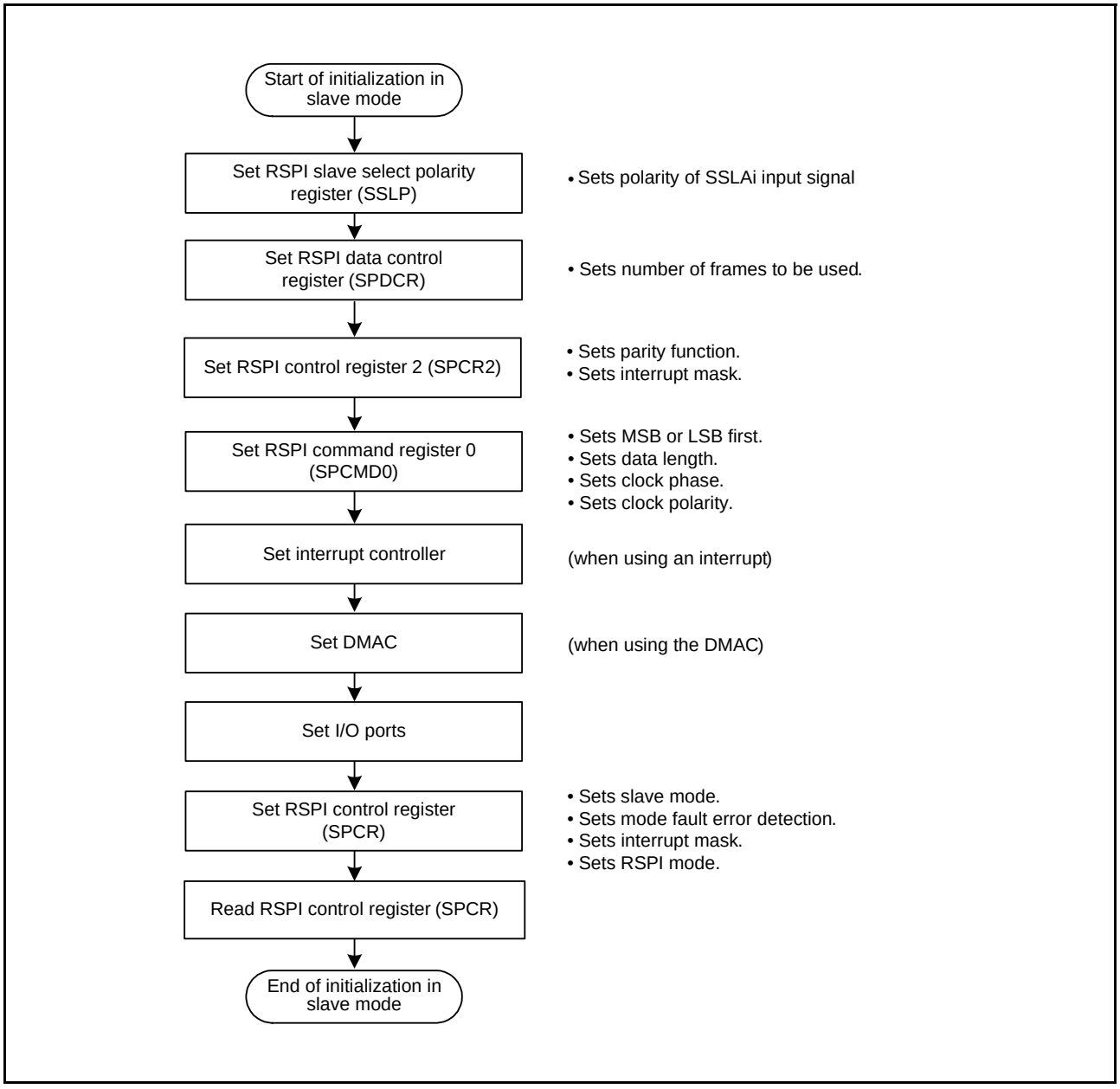


Figure 44.39 Example of Initialization Flowchart in Slave Mode (SPI Operation)

(6) Software Processing Flow

Figure 44.40 to Figure 44.42 show examples of the flow of software processing.

(a) Transmit Processing Flow

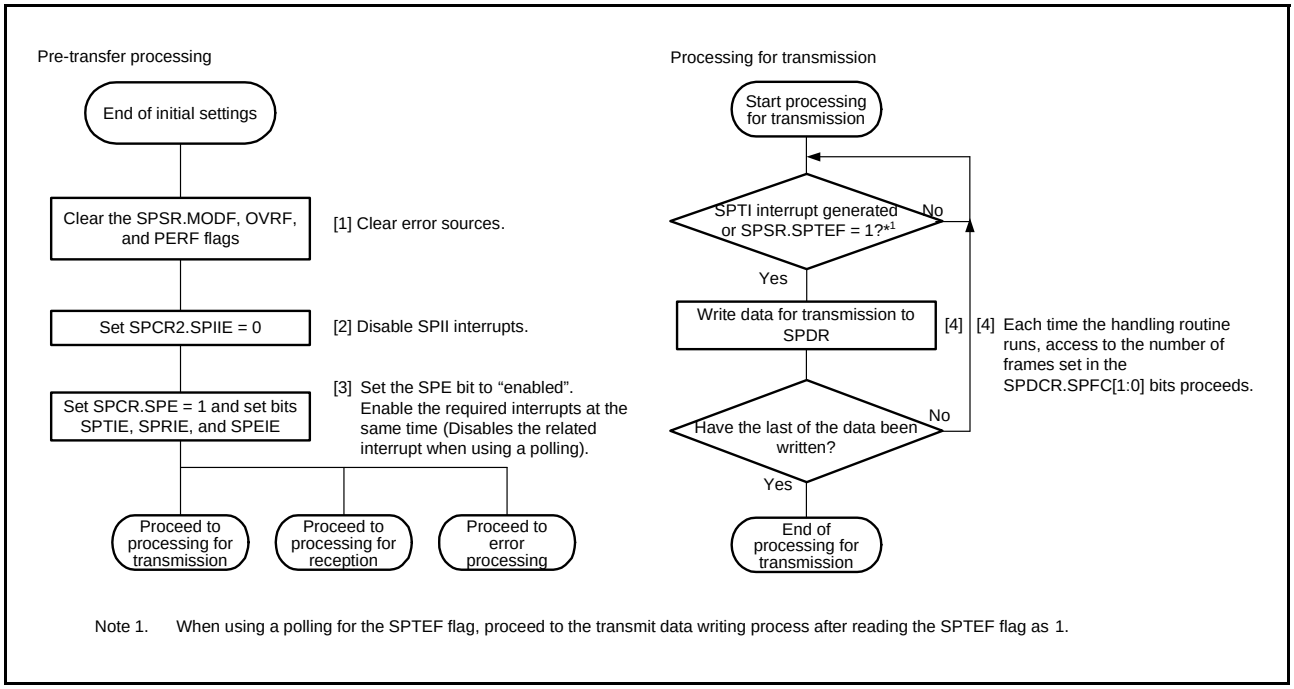


Figure 44.40 Flowchart in Slave Mode (Transmission)

(b) Receive Processing Flow

The RSPI does not handle receive-only operation, so processing for transmission is required.

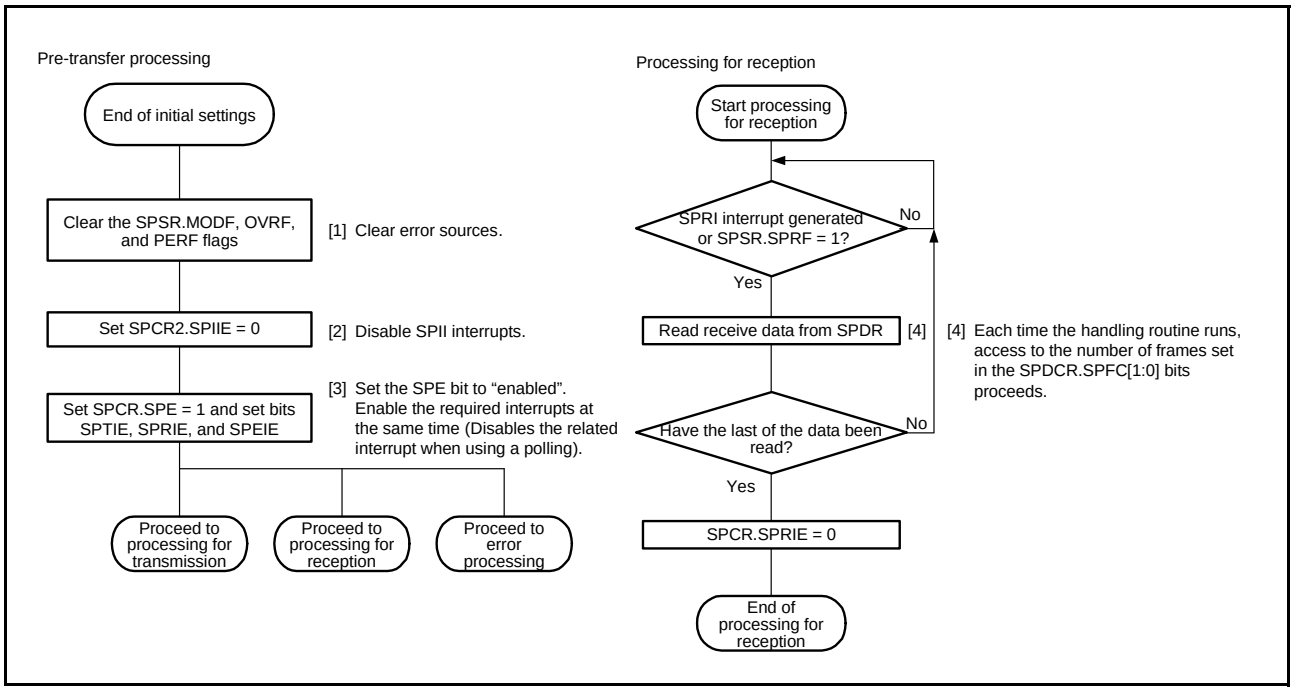


Figure 44.41 Flowchart in Slave Mode (Reception)

(c) Flow of Error Processing

In slave operation, even when a mode fault error is generated, the SPSR.MODF flag can be cleared regardless of the status of the SSLA0 pin.

When interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

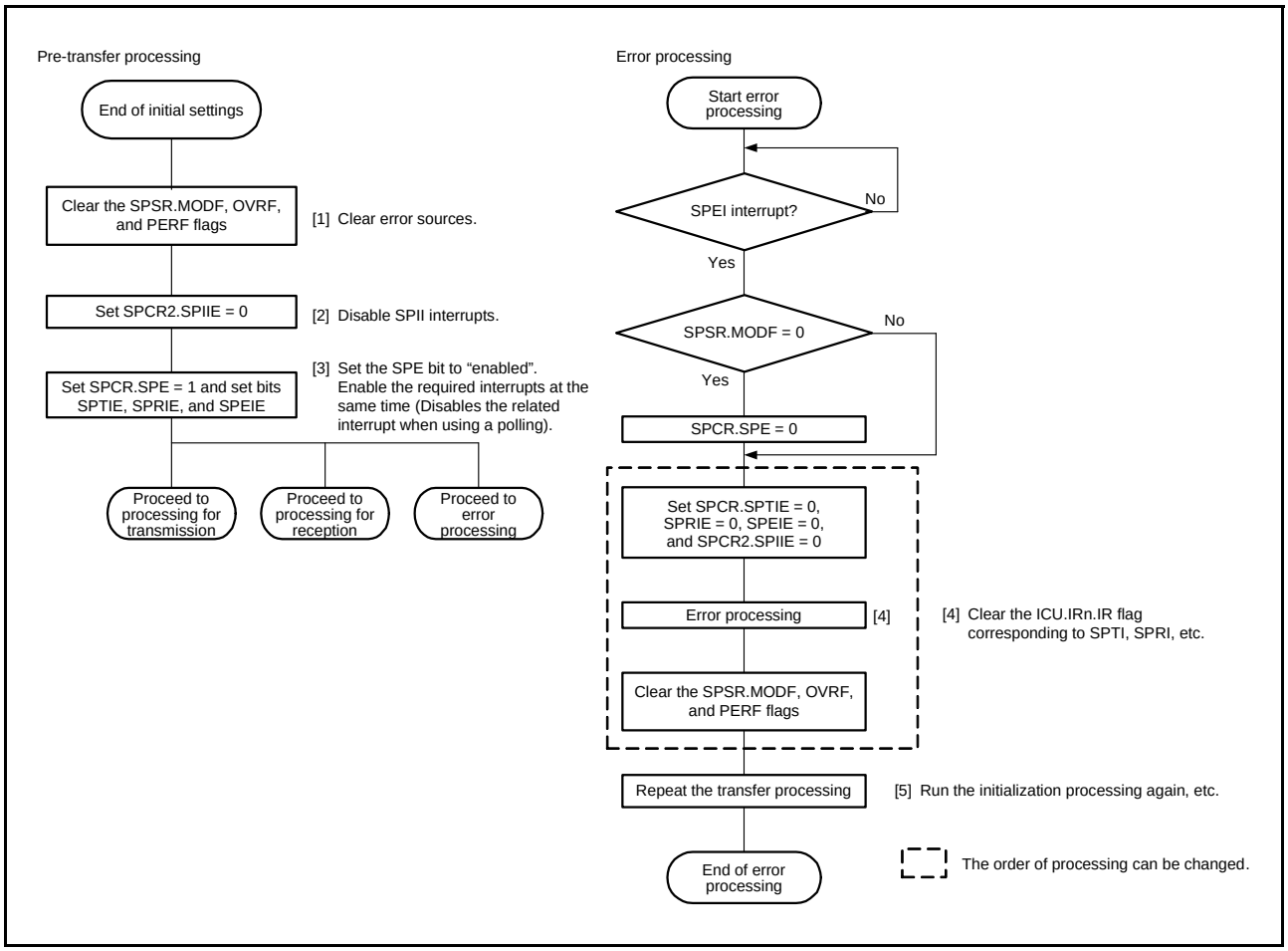


Figure 44.42 Flowchart for Slave Mode (Error Processing)

44.3.11 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the RSPI. In clock synchronous operation, the SSLAi pin is not used, and the three pins of RSPCKA, MOSIA, and MISOA handle communications. The SSLAi pin is available as I/O port pins.
Although clock synchronous operation does not require use of the SSLAi pin, operation of the module is the same as in SPI operation. That is, in both master and slave operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected because the SSLAi pin is not used.
Furthermore, operation should not be performed if clock synchronous operation proceeds when the SPCMDm.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0).

44.3.11.1 Master Mode Operation

(1) Starting a Serial Transfer

The RSPI updates the data in the transmit buffer (SPTX) of SPDR when data is written to the SPDR register with the transmit buffer being empty (the SPTEF flag is 1 and data for the next transfer is not set). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR, the RSPI copies data from the transmit buffer to the shift register and starts serial transmission. Upon copying transmit data to the shift register, the RSPI changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.
For details on the RSPI transfer format, refer to section 44.3.5, Transfer Format.
However, transfer in clock synchronous operation is conducted without the SSLA0 output signal.

(2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after transmitting an RSPCKA edge corresponding to the sampling timing. If free space is available in the receive buffer (SPRX) (the SPRF flag is 0), upon termination of serial transfer, the RSPI copies data from the shift register to the receive buffer of the RSPI data register (SPDR).
It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the SPCMDm.SPB[3:0] bit setting.
For details on the RSPI transfer format, refer to section 44.3.5, Transfer Format.
However, transfer in clock synchronous operation is conducted without the SSLA0 output signal.

(3) Sequence Control

The transfer format employed in master mode is determined by SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers. Although the SSLAi signals are not output in clock synchronous operation, these settings are valid.
SPSCR is a register used to determine the sequence configuration for serial transfers that are executed by the RSPI in master mode. The following items are set in SPCMDm register: SSLAi output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCKA polarity/phase, whether SPCKD is to be referenced, whether SSLND is to be referenced, and whether SPND is to be referenced. SPBR holds some of the bit rate settings; SPCKD, an RSPI clock delay value; SSLND, an SSL negation delay; and SPND, a next-access delay value.
According to the sequence length that is assigned to SPSCR, the RSPI makes up a sequence comprised of a part or all of SPCMDm register. The RSPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD0 register, and incorporates the SPCMD0 register setting into the transfer format at the beginning of serial transfer. The RSPI increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, the RSPI sets the pointer in SPCMD0 register, and in this manner the sequence is executed repeatedly.

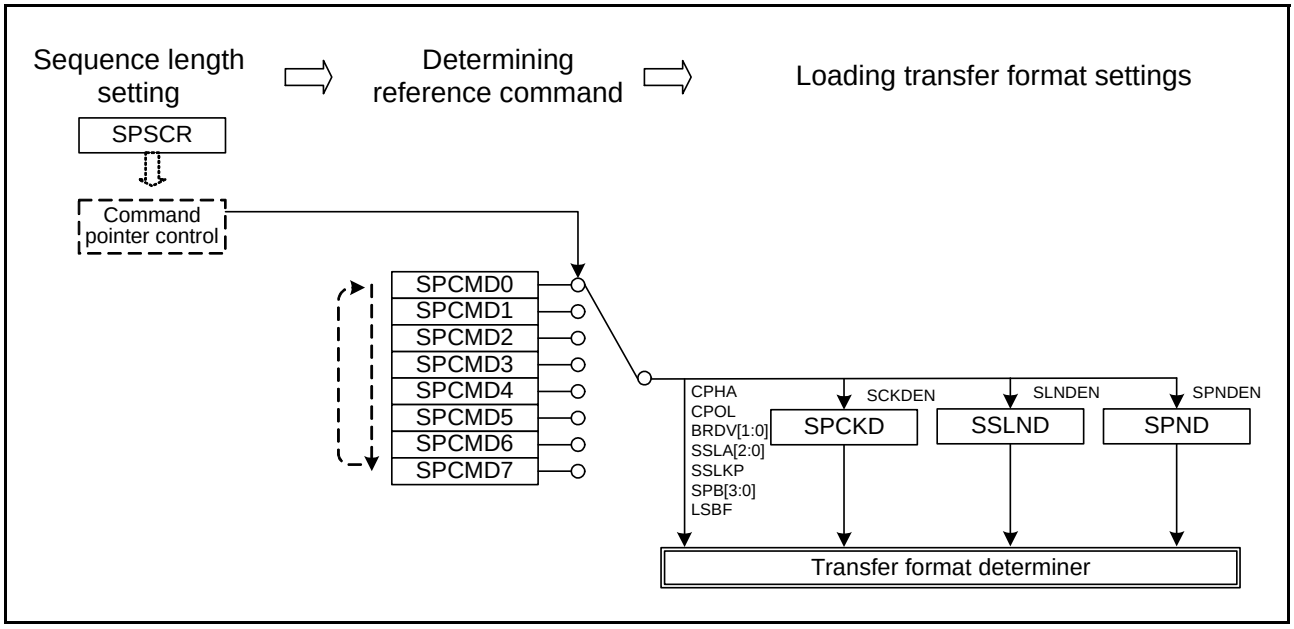


Figure 44.43 Procedure for Determining the Form of Serial Transmission in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

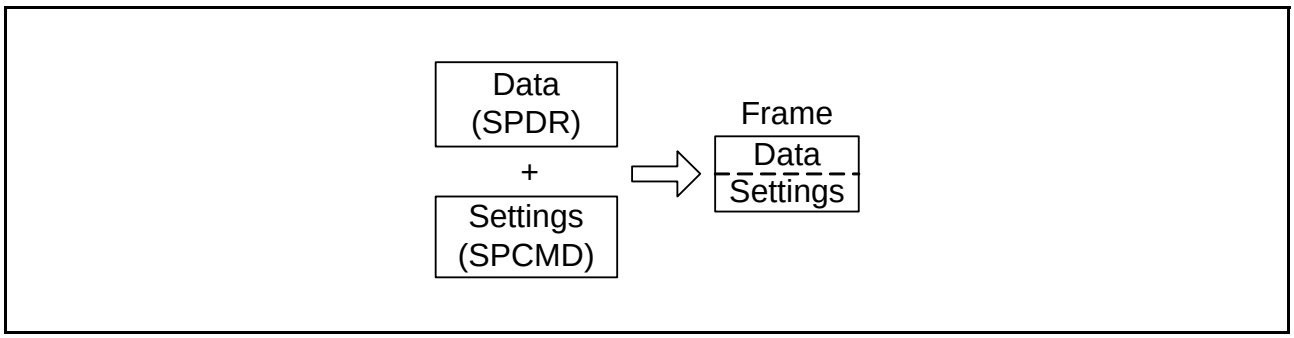


Figure 44.44 Concept of a Frame

Figure 44.45 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 44.4.

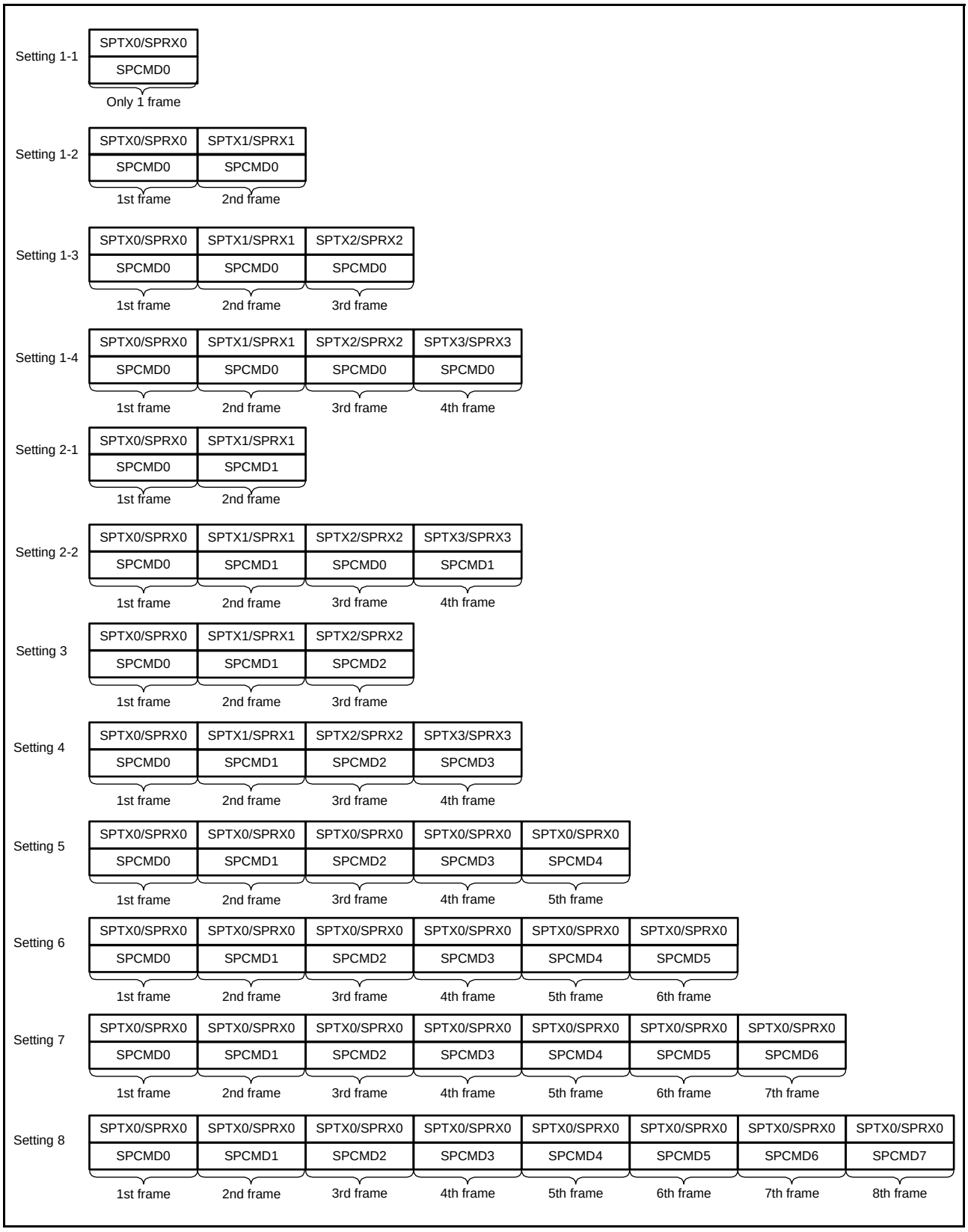


Figure 44.45 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(4) Initialization Flowchart

Figure 44.46 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPI is used in master mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

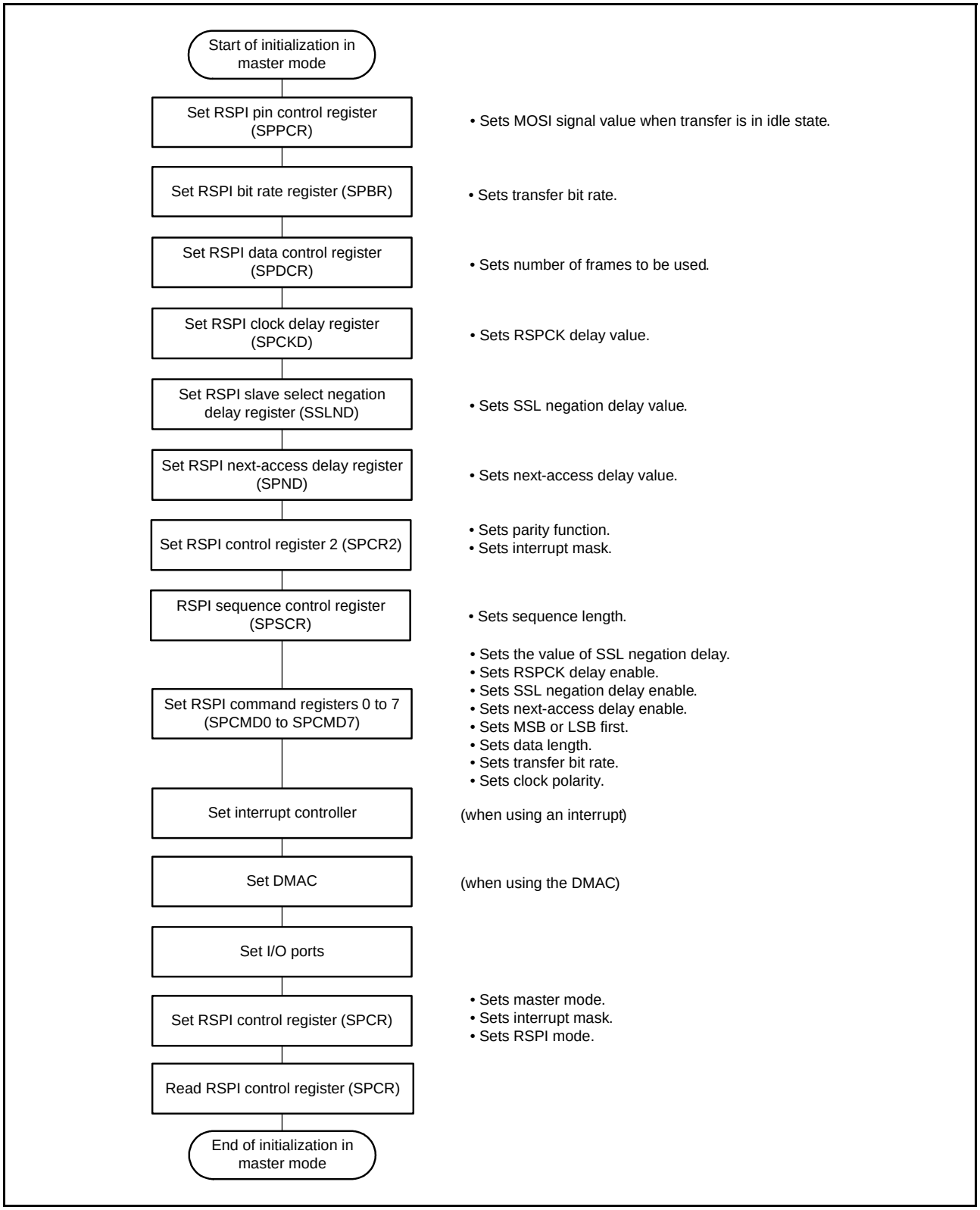


Figure 44.46 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)

(5) Flow of Software Processing

Software processing during clock-synchronous master operation is the same as that for SPI master operation. For details, refer to section 44.3.10.1, (9) Software Processing Flow. Note that mode fault errors will not occur.

44.3.11.2 Slave Mode Operation

(1) Starting a Serial Transfer

When the SPCR.SPMS bit is 1, the first RSPCKA edge triggers the start of a serial transfer in the RSPI.
When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to “full”, so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI keeps the status of the shift register unchanged, in the full state.
When the SPMS bit is 1, the RSPI drives the MISOA output signal.
For details on the RSPI transfer format, refer to section 44.3.5, Transfer Format.
It should be noted that the SSLA0 input signal is not used in clock synchronous operation.

(2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after detecting an RSPCKA edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPRF flag is 0), upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty” regardless of the receive buffer status. The final sampling timing changes depending on the bit length of transfer data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[3:0] bit setting.
For details on the RSPI transfer format, refer to section 44.3.5, Transfer Format.

(3) Initialization Flowchart

Figure 44.47 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPI is used in slave mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

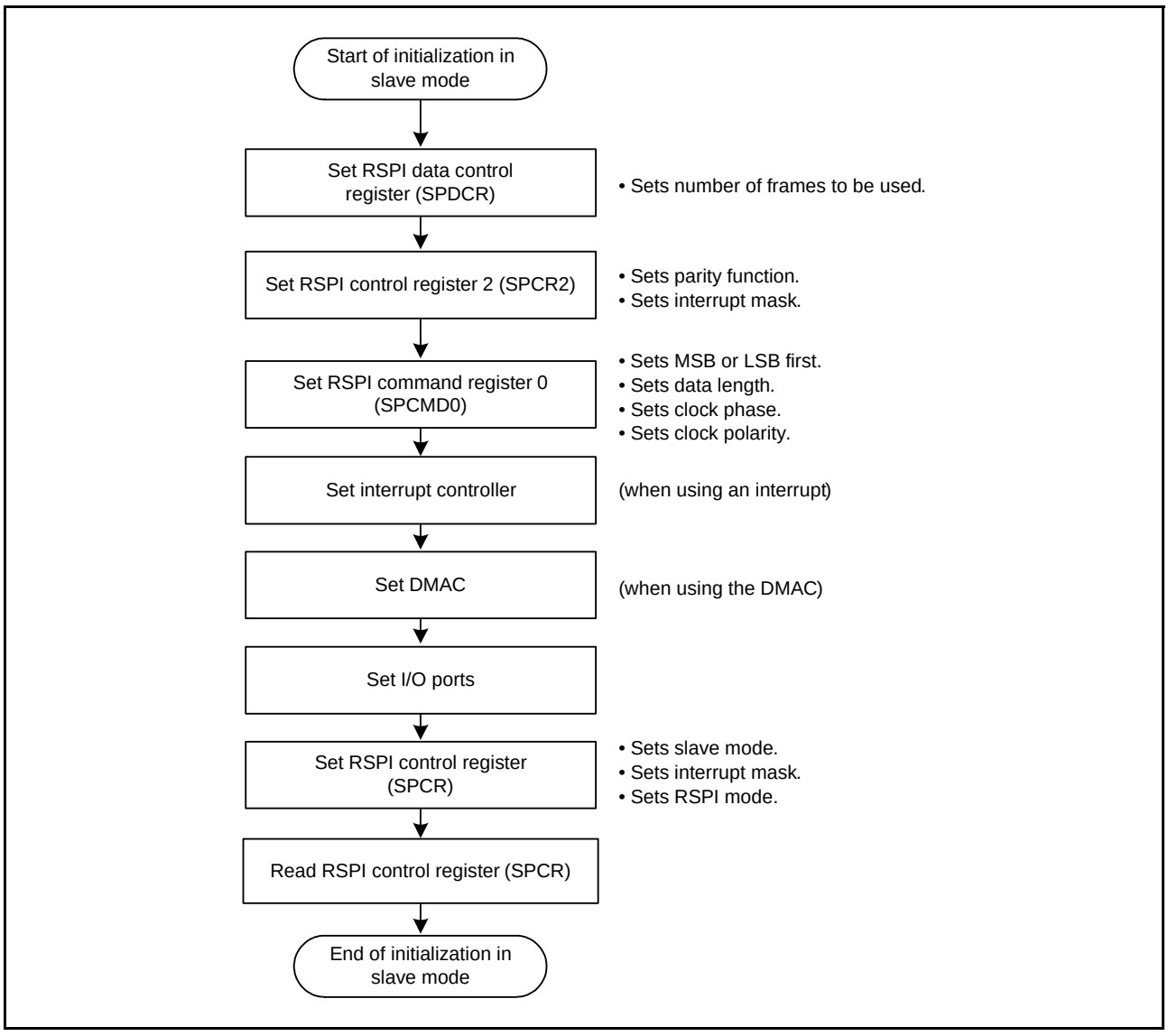


Figure 44.47 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation)

(4) Flow of Software Processing

Software processing during clock-synchronous slave operation is the same as that for SPI slave operation. For details, refer to section 44.3.10.2, (6) Software Processing Flow. Note that mode fault errors will not occur.

44.3.12 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the RSPI shuts off the path between the MISOA pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIA pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path of the shift register. The RSPI does not shut off the path between the MOSIA pin and the shift register if the SPCR.MSTR bit is 1, and between the MISOA pin and the shift register if the SPCR.MSTR bit is 0. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the RSPI or the reversed transmit data becomes the received data for the RSPI.
Table 44.12 lists the relationship among the SPLP2 and SPLP bits and the received data. Figure 44.48 shows the configuration of the shift register I/O paths for the case where the RSPI in master mode is set in loopback mode (SPPCR.SPLP2 = 0, SPPCR.SPLP = 1).

Table 44.12 SPLP2 and SPLP Bit Settings and Received Data

SPPCR.SPLP2 Bit	SPPCR.SPLP Bit	Received Data
0	0	Input data from the MOSIA pin or MISOA pin
0	1	Inverted transmit data
1	0	Transmit data
1	1	Transmit data

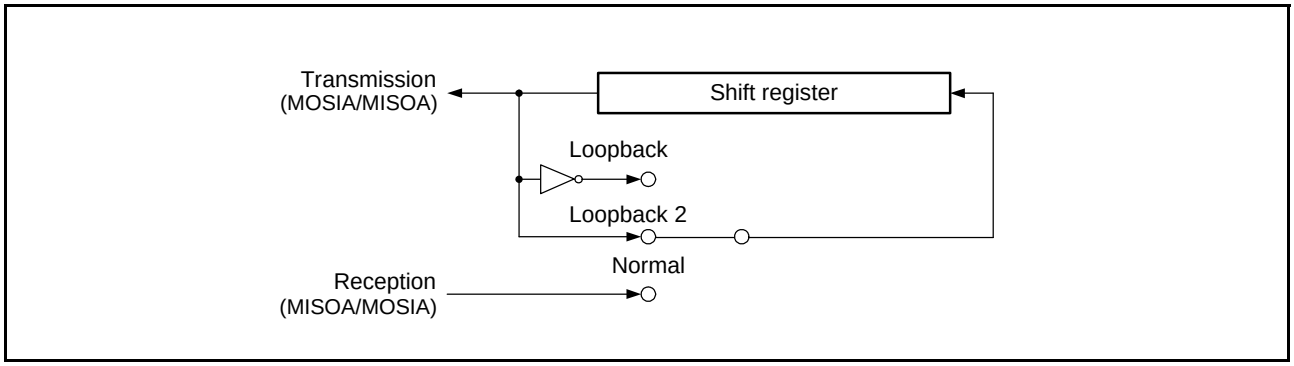


Figure 44.48 Configuration of Shift Register I/O Paths in Loopback Mode (Master Mode)

44.3.13 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for transmit data and an error detecting unit used for received data. In order to detect defects in the parity bit adding unit and error detecting unit of the parity circuit, self-diagnosis is executed for the parity circuit following the flowchart shown in Figure 44.49.

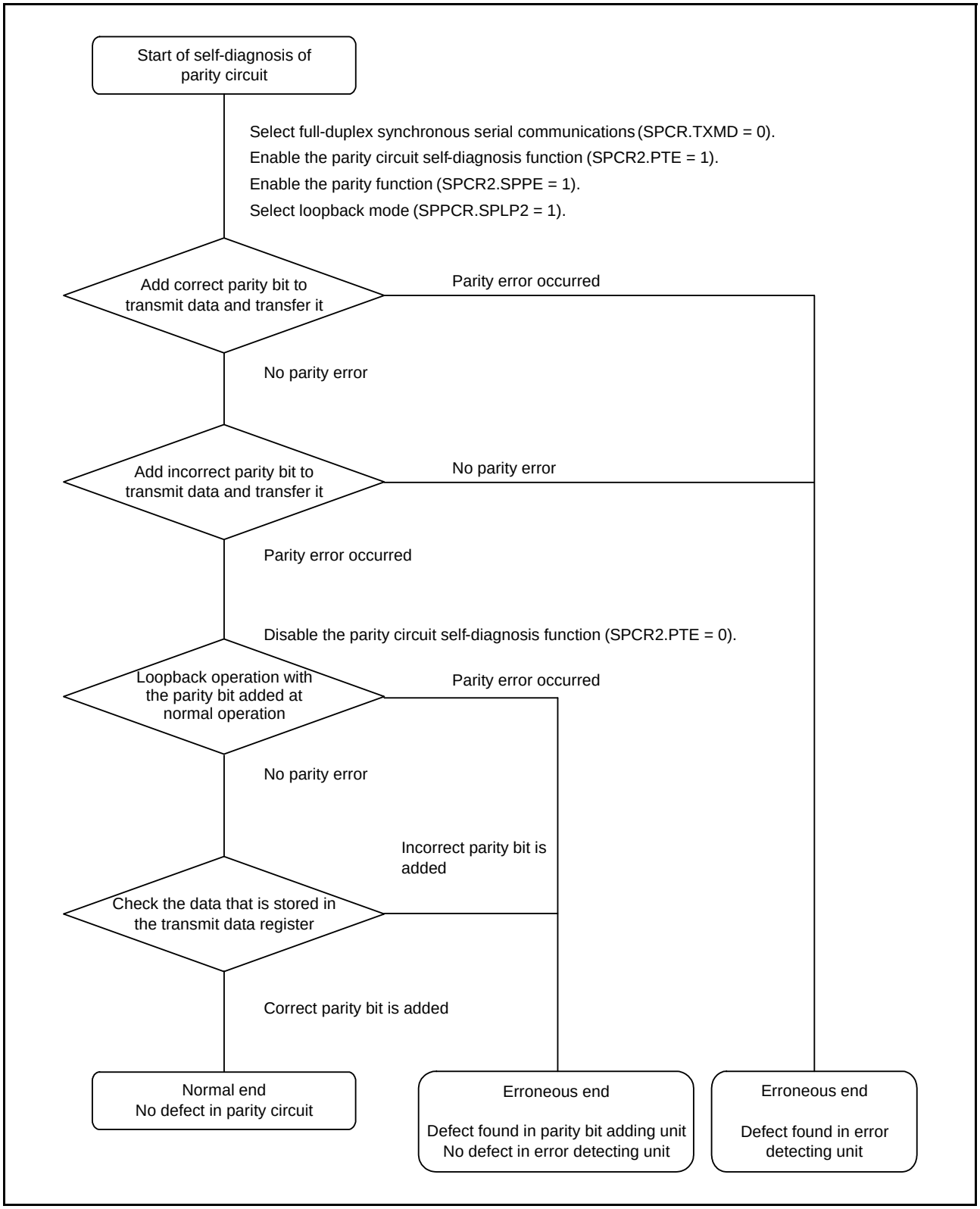


Figure 44.49 Flowchart for Self-Diagnosis of Parity Circuit

44.3.14 Interrupt Sources

The RSPI has interrupt sources of receive buffer full, transmit buffer empty, mode fault, overrun, parity error, and RSPI idle. In addition, the DTC or DMAC can be activated by the receive buffer full or transmit buffer empty interrupt to perform data transfer.

Since the vector address for SPEI is allocated to interrupt requests due to mode fault, overrun, and parity errors, the actual interrupt source must be determined from the flags. Interrupt sources for the RSPI are listed in Table 44.13. An interrupt is generated on satisfaction of an interrupt condition in Table 44.13. Clear the receive buffer full and transmit buffer empty sources through data transfer.

When using the DTC or DMAC to perform data transmission/reception, the DTC or DMAC must be set up first to be in a status in which transfer is enabled before making the RSPI settings. For the method for setting the DTC or DMAC, refer to section 18, DMA Controller (DMACAa), or section 20, Data Transfer Controller (DTCa).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt are generated while the ICU.IRn.IR flag is 1, the interrupt is not output as a request for ICU but is retained internally (the capacity for retention is one request per source). A retained interrupt request is output when the ICU.IRn.IR flag becomes 0. A retained interrupt request is automatically discarded once it is output as an actual interrupt request. The interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) for an internally retained interrupt request can also be cleared to 0.

Table 44.13 Interrupt Sources of RSPI

Interrupt Source	Symbol	Interrupt Condition	DMAC/DTC Activation
Receive buffer full	SPRI	The receive buffer becomes full (the SPRF flag becomes 1) while the SPCR.SPRIE bit is 1.	Possible
Transmit buffer empty	SPTI	The transmit buffer becomes empty (the SPTEF flag becomes 1) while the SPCR.SPTIE bit is 1.	Possible
RSPI errors (mode fault, overrun and parity error)	SPEI	The SPSR.MODF, OVRF, or PERF flag is set to 1 while the SPCR.SPEIE bit is 1.	Impossible
RSPI idle	SPII	The SPSR.IDLNF flag is set to 0 while the SPCR2.SPIIE bit is 1.	Impossible

44.4 Link Operation by Event Linking

The RSPI0 supports the following event output for the event link controller (ELC). The event link output signal is output regardless of the interrupt enable bit setting.

44.4.1 Receive Buffer Full Event Output

This event signal is output when received data have been transferred from the shift register to the SPDR on completion of serial transfer.

44.4.2 Transmit Buffer Empty Event Output

This event signal is output when data for transmission have been transferred from the transmit buffer to the shift register and when the value of the SPE bit has changed from 0 to 1.

44.4.3 Mode Fault, Overrun, or Parity Error Event Output

(1) Mode Fault

Table 44.14 lists the occurrence conditions of a mode fault event.

Table 44.14 Occurrence Conditions of Mode Fault Event

	SPCR.MODFEN Bit	SSLA0 Pin	Remarks
Master (SPCR.MSTR bit = 1)	1	Active	Under the condition (the SPCR.MSTR bit is 1 and the MODFEN bit is 1), if the SPCR.SPMS bit is 0, mode fault error, overrun error, and parity error event output cannot be used. Do not set the ELSRn register to 52h.
Slave (SPCR.MSTR bit = 0)	1	Not active	Event is output only when the pin is deactivated during transmission.

(2) Overrun

The condition for this event signal being output in response to an overrun is completion of serial transfer while the receive buffer contains data that have not been read and the value of the SPCR.TXMD bit is 0, in which case the OVRF flag is set to 1.

(3) Parity Error

The condition for this event signal being output in response to a parity error is detection of a parity error on completion of serial transfer while the value of the TXMD bit in SPCR is 0 and the value of the SPPE bit in SPCR2 is 1.

44.4.4 RSPI Idle Event Output

(1) In Master Mode

In master mode, an event is output when the condition for setting the IDLNF flag (RSPI idle flag) to 0 is satisfied.

(2) In Slave Mode

In slave mode, an event is output when the SPCR.SPE bit is set to 0 (RSPI is initialized).

44.4.5 Transmission-Completed Event Output

During both SPI operation and clock synchronous operation in master mode, an event is output under the condition for setting the IDLNF flag (RSPI idle flag) from 1 to 0.

Table 44.15 Conditions for Generation of a Transmission-Completed Event (Slave)

	Transmit Buffer State	Shift Register State	Others
SPI operation (SPMS = 0)	Empty	Empty	Negation of SSLA0 input
Clock synchronous operation (SPMS = 1)	Empty	Empty	Edge detection of the last RSPCKA

Whether the operation is in master mode or slave mode, an event is not output if 0 is written to the SPCR.SPE bit in transmission or the SPCR.SPE bit is cleared by the mode fault error.

44.5 Usage Notes

44.5.1 Setting Module Stop Function

Module stop control register B (MSTPCRB) can be used to enable or disable the RSPI. Immediately after a reset, operation of the RSPI is disabled. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

44.5.2 Note on Low Power Consumption Functions

When using the module stop function and entering a low power consumption mode other than sleep mode, set the SPCR.SPE bit to 0 before completing communication.

44.5.3 Notes on Starting Transfer

If the ICU.IRn.IR flag is 1 at the time transfer is to be started, an interrupt request is internally retained after transfer starts, and this can lead to unanticipated behavior of the ICU.IRn.IR flag.
When the ICU.IRn.IR flag is 1 at the time transfer is to start, follow the procedure below to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1).

1. Confirm that transfer has stopped (i.e. that the SPCR.SPE bit is 0).
2. Set the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) to 0.
3. Read the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) and confirm that its value is 0.
4. Set the ICU.IRn.IR flag to 0.

44.5.4 Notes on the SPRF and SPTEF flags

When polling the SPSR.SPRF flag and/or SPSR.SPTEF flag, set the SPCR.SPRIE bit and/or SPCR.SPTIE bit to 0.

45. Quad Serial Peripheral Interface (QSPI)

This MCU includes a one-channel quad serial peripheral interface.
In this section, “PCLK” is used to refer to PCLKB.

45.1 Overview

Table 45.1 lists the specifications of the QSPI, Figure 45.1 shows a block diagram of the QSPI, and Table 45.2 lists the I/O pins of the QSPI.

Table 45.1 QSPI Specifications

Item	Description
Number of channels	1 channel
QSPI transfer function	Capable of communications to the serial flash memory through single-/dual-/quad-SPI operation • Single-SPI operation (full-duplex communications) Use of QMO (master out), QMI (master in), QSSL (slave select), and QSPCLK (SPI clock) signals allow for communications to the serial flash memory through SPI operation (four-wire method). QMO output pin and QMI input pin QSSL and QSPCLK serve as output pins. • Dual-SPI operation (half-duplex communications) Use of QIO1, QIO0, QSSL, and QSPCLK signals allow for communications to the serial flash memory through SPI operation (four-wire method). QSSL and QSPCLK serve as output pins. Bidirectional QIO1 and QIO0 pins • Quad-SPI operation (half-duplex communications) Use of QIO3 to QIO0, QSSL, and QSPCLK signals allow for communications to the serial flash memory through SPI operation (six-wire method). QSSL and QSPCLK serve as output pins. Bidirectional QIO3 to QIO0 pins
Data format	• Transfer data length is selectable from 8 bits to 128 Gbits. Data is continuously transferred 1 through 4,294,967,296 times in 8-, 16-, or 32-bit units.
Bit rate	• QSPCLK can be divided by 2 to 4080 (max.). • QSPCLK can be generated by dividing PCLK by the on-chip baud rate generator.
Buffer configuration	• 32 bytes for transmission • 32 bytes for reception
Shift registers	• 32 bits each for transmission and reception
QSSL pin control	• Controllable delay from QSSL signal assertion to QSPCLK operation (clock delay) Range: 0 and 1.5 to 8.5 QSPCLK cycles (set in QSPCLK-cycle units) • Controllable delay from QSPCLK stoppage to QSSL output negation (QSSL negation delay) Range: 0 to 8 QSPCLK cycles (set in QSPCLK-cycle units) • Controllable wait for next-access QSSL output assertion (next-access delay) Range: 0 to 8 QSPCLK cycles (set in QSPCLK-cycle units) • Capable of holding QSSL output value from transfer end to next access • QSSL polarity can be changed.
Transfer control	• A transfer of up to four commands can be executed sequentially in looped execution. • Single-SPI transfer start condition: The specified length of data is in the transmit buffer and there is a space for the specified length of data in the receive buffer. However, when the SPDCR.TXDMY bit is set, receive operation can be started even when the specified length of data is not in the transmit buffer. • Dual-/quad-SPI write start condition: The specified length of data is in the transmit buffer. • Dual-/quad-SPI read start condition: There is a space for the specified length of data in the receive buffer. • QIO3 to QIO0 and QMO output values during the QSSL negation period can be specified. • QIO3 and QIO2 can be specified to output fixed values in single-/dual-SPI modes.
Interrupt sources	• Receive buffer full interrupt • Transmit buffer empty interrupt • QSSL negation interrupt
Others	• QSPI initialization function • Loop back mode
Low power consumption function	Module stop state can be set.

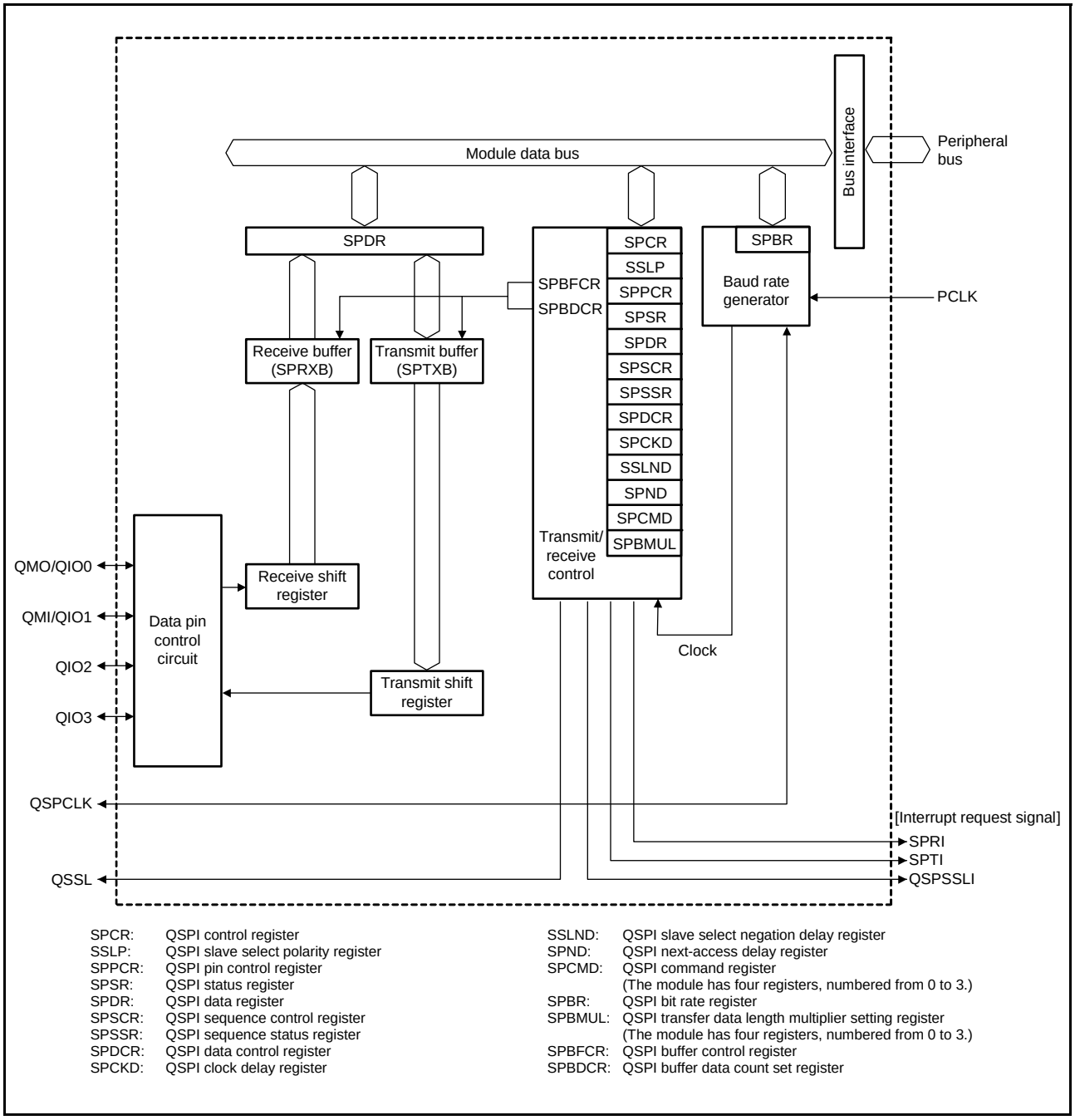


Figure 45.1 QSPI Block Diagram

Table 45.2 QSPI I/O Pins

Pin Name	I/O	Function
QSPCLK	Output	Clock output
QMO/QIO0*1	I/O	Master output data/data 0
QMI/QIO1*1	I/O	Master input data/data 1
QIO2*2	I/O	Data 2
QIO3*2	I/O	Data 3
QSSL	Output	Slave selection

Note 1. In single-SPI mode, QMO and QMI are enabled; QIO0 and QIO1 in dual-/quad-SPI modes.
Note 2. In single-/dual-SPI modes, fixed value according to register setting is output; QIO2 and QIO3 in quad-SPI mode.

45.2 Register Descriptions

45.2.1 QSPI Control Register (SPCR)

Address(es): QSPI.SPCR 0008 9E00h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRIE	SPE	SPTIE	—	MSTR	—	SPSSLIE	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	SPSSLIE	QSSL Negation Interrupt Enable	0: Disables interrupt requests. 1: Enables interrupt requests.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	MSTR	QSPI Master/Slave Mode Select *1	0: Slave mode 1: Master mode (output from QSPCLK and QSSL pins enabled)	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	SPTIE	Transmit Interrupt Enable	0: Disables interrupt requests. 1: Enables interrupt requests.	R/W
b6	SPE	SPI Function Enable	0: Disables the function. 1: Enables the function.	R/W
b7	SPRIE	Receive Interrupt Enable	0: Disables interrupt requests. 1: Enables interrupt requests.	R/W

Note 1. This MCU does not support slave mode. When using this bit, set it to 1.

The SPCR register sets the operating mode for the QSPI. If the set value of the MSTR bit is changed while the SPE bit is 1, subsequent operation cannot be guaranteed.

SPSSLIE Bit (QSSL Negation Interrupt Enable)

This bit enables or disables generation of QSSL negation interrupt requests when the SPSR.SPSSLF flag is set to 1.

MSTR Bit (QSPI Master/Slave Mode Select)

This bit selects master or slave mode. Setting this bit to 1 allows the QSPCLK and QSSL pins to be used as output pins. When setting the MSTR bit, specify the following while the SPE bit is 0, and then set the MSTR bit to 1.

- SSLP.SSLP bit setting (QSSL pin level setting)
- SPCMDn.CPOL bit (n = 0 to 3) setting (QSPCLK pin level setting)

Note that this MCU does not support slave mode.

SPTIE Bit (Transmit Interrupt Enable)

This bit enables or disables generation of transmit interrupt requests when the number of transmit data units in the transmit buffer is equal to or less than the specified transmit buffer data triggering number leading the corresponding flag in the SPSR register being set to 1.

SPE Bit (SPI Function Enable)

Setting this bit to 1 enables the module function.
Setting this bit to 0 disables and partially initializes the module function.

SPRIE Bit (Receive Interrupt Enable)

This bit enables or disables generation of receive interrupt requests when the number of receive data units in the receive buffer is equal to or greater than the specified receive buffer data triggering number leading the corresponding flag in the status register being set to 1.

45.2.2 QSPI Slave Select Polarity Register (SSLP)

Address(es): QSPI.SSLP 0008 9E01h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SSLP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SSLP	QSSL Signal Polarity Setting	0: QSSL signal is active-low. 1: QSSL signal is active-high.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SSLP register sets the polarity of the QSSL signal. If this register is modified while the SPCR.SPE bit is 1, subsequent operation cannot be guaranteed.

SSLP Bit (QSSL Signal Polarity Setting)

This bit sets the polarity of the QSSL signal. The polarity should be same as that of the device to be communicated with.

45.2.3 QSPI Pin Control Register (SPPCR)

Address(es): QSPI.SPPCR 0008 9E02h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MOIFE	MOIFV	—	IO3FV	IO2FV	SPLP
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPLP	Loopback mode*1	0: Normal mode 1: Loopback mode	R/W
b1	IO2FV	Single-/Dual-SPI Mode QIO2 Output Fixed Value	0: QIO2 output fixed value equals 0. 1: QIO2 output fixed value equals 1.	R/W
b2	IO3FV	Single-/Dual-SPI Mode QIO3 Output Fixed Value	0: QIO3 output fixed value equals 0. 1: QIO3 output fixed value equals 1.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	MOIFV	Idle State Output Data Fixed Value	0: Data output value in the idle state (during the QSSL negation period) is 0. 1: Data output value in the idle state (during the QSSL negation period) is 1.	R/W
b5	MOIFE	Idle State Output Data Fixing Enable*2	0: Output value equals final data from previous transfer. 1: Output value equals the value set in the MOIFV bit.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the loopback mode is specified in dual-/quad-SPI modes, the SPCMDn.SPRW bit (n = 0 to 3) should be set to 0 (write operation).

Note 2. In dual-/quad-SPI modes, QIO1 and QIO0/QIO3 to QIO0 are driven to the Hi-Z state after read transfer regardless of this bit setting (see section 45.3.2, Pin Control).

The SPPCR register sets the modes of the pins. If this register is modified while the SPCR.SPE bit is 1, subsequent operation cannot be guaranteed.

SPLP Bit (Loopback mode)

When the SPLP bit is set to 1, this module shuts off the path between the data I/O pin and the transmit/receive shift register, and connects the input path and the output path for the transmit/receive shift register.

IO2FV Bit (Single-/Dual-SPI Mode QIO2 Output Fixed Value)

This bit sets the output value of the QIO2 pin in single-/dual-SPI modes. This bit is valid only in single-/dual-SPI modes, and is not affected by the MOIFE or MOIFV bit values.

IO3FV Bit (Single-/Dual-SPI Mode QIO3 Output Fixed Value)

This bit sets the output value of the QIO3 pin in single-/dual-SPI modes. This bit is valid only in single-/dual-SPI modes, and is not affected by the MOIFE or MOIFV bit values.

MOIFV Bit (Idle State Output Data Fixed Value)

If the MOIFE bit is 1, the pin output value during the QSSL negation period is determined according to this bit setting.

MOIFE Bit (Idle State Output Data Fixing Enable)

This bit fixes the pin output value in a QSSL negation period or the QSSL keeping period during a burst transfer. In single-SPI mode, this bit setting applies to QMO. In dual-SPI mode, this bit setting applies to QIO1 and QIO0. In quad-SPI mode, this bit setting applies to QIO3 through QIO0.

45.2.4 QSPI Status Register (SPSR)

Address(es): QSPI.SPSR 0008 9E03h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRFF	TREND	SPTEF	SPSSLF	—	—	—	—
Value after reset:	0	1	1	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SPSSLF	QSSL Negation Flag	0: QSSL negation is not detected. 1: QSSL negation is detected.	R/(W) *1
b5	SPTEF	Transmit Buffer Empty Flag	0: The number of transmit data units in the transmit buffer is greater than the transmit buffer data triggering number. 1: The number of transmit data units in the transmit buffer is equal to or less than the specified transmit buffer data triggering number.	R/(W) *1
b6	TREND	Transmit End Flag	0: Transmission is not completed. 1: Transmission is completed.	R
b7	SPRFF	Receive Buffer Full Flag	0: The number of receive data units in the receive buffer is less than the receive buffer data triggering number. 1: The number of receive data units in the receive buffer is equal to or greater than the receive buffer data triggering number.	R/(W) *1

Note 1. Only 0 can be written to clear the flag after reading 1.

The SPSR register holds flags indicating the operating state of this module.

SPSSLF Flag (QSSL Negation Flag)

This flag indicates the state of QSSL. This flag should be read after the receive buffer (SPRXB) is read.
[Setting conditions]

- QSSL output is changed from 0 to 1 when the SSLP.SSLP bi is 0.
- QSSL output is changed from 1 to 0 when the SSLP.SSLP bi is 1.

[Clearing conditions]

- After reading the SPSSLF flag = 1, 0 is written to it.
- Reset

SPTEF Flag (Transmit Buffer Empty Flag)

This flag indicates that the number of transmit data units in the transmit buffer is equal to or less than the transmit buffer data triggering number specified in the buffer control register. A flag-clearing operation takes precedence over a flag-setting operation.
[Setting conditions]

- When the number of transmit data units in the transmit buffer is equal to or less than the specified transmit buffer data triggering number.
- When transmit buffer reset is enabled.
- Reset

[Clearing conditions]

- When 0 is written to the SPTEF flag after reading SPTEF = 1.
- When data is written to the transmit buffer (SPTXB) using DTC or DMAC transfer (transfer of the last block in block transfer).

Note: Do not clear the SPTEF flag to 0 during DTC or DMAC transfer.

TREND Flag (Transmit End Flag)

This flag is set to 1 when transmission is completed, and this flag is 0 when transmission is not completed. When the value of the SPCR.SPE bit is changed from 1 to 0, read this flag after waiting for the division ratio setting cycles plus four PCLK cycles.

For division ratio settings, see Table 45.3, SPBR Register and SPCMDn.BRDV[1:0] Bit Settings and Bit Rates.

[Setting condition]

- When the number of data units stored in the transmit buffer is zero when a serial transmission is completed except when the SPDCR.TXDMY bit is set to 1.

[Clearing condition]

- When transmit data is transferred from the transmit buffer to the transmit shift register.

SPRFF Flag (Receive Buffer Full Flag)

This flag indicates that the number of receive data units in the receive buffer is equal to or greater than the receive buffer data triggering number specified in the buffer control register. A flag-clearing operation takes precedence over a flag-setting operation.

[Setting condition]

- The number of data units in the receive buffer is equal to or greater than the specified receive buffer data triggering number.

[Clearing conditions]

- When 0 is written to the SPRFF flag after reading SPRFF = 1.
- When data is read from the receive buffer (SPRXB) using DTC or DMAC transfer (transfer of the last block in block transfer). Do not clear the SPRFF flag to 0 during DTC or DMAC transfer.

45.2.5 QSPI Data Register (SPDR)

- When accessing in longword size

Address(es): QSPI.SPDR 0008 9E04h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

- When accessing in word size

Address(es): QSPI.SPDR.H 0008 9E04h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	H															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

- When accessing in byte size

Address(es): QSPI.SPDR.HH 0008 9E04h

	b7	b6	b5	b4	b3	b2	b1	b0
	HH							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The SPDR register accesses transmit/receive data buffer.
The transmit buffer (SPTXB) and receive buffer (SPRXB) are independent and are mapped to the SPDR register.
When data is written to this register, the data will be written to the transmit buffer.
When data is read from this register, the data will be read from the receive buffer.
This register should be read or written to in byte, word, or longword units.
This register should be accessed in longword, word, or byte size.

45.2.6 QSPI Sequence Control Register (SPSCR)

Address(es): QSPI.SPSCR 0008 9E08h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SPSC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SPSC[1:0]	Sequence Control Specification	Number of SPCMDn register (n = 0 to 3) to be referenced b1 b0 0 0: 0 → 0 → ... 0 1: 0 → 1 → 0 → ... 1 0: 0 → 1 → 2 → 0 → ... 1 1: 0 → 1 → 2 → 3 → 0 → ...	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPSCR register sets the sequence controlled method. If this register is modified while the SPCR.SPE bit is 1, subsequent operation cannot be guaranteed.

SPSC[1:0] Bits (Sequence Control Specification)

These bits specify sequential operations.

This module references the SPCMDn register (n = 0 to 3) in the order according to these bit settings.

45.2.7 QSPI Sequence Status Register (SPSSR)

Address(es): QSPI.SPSSR 0008 9E09h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SPSS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SPSS[1:0]	Sequence Status	b1 b0 0 0: SPCMD0 register 0 1: SPCMD1 register 1 0: SPCMD2 register 1 1: SPCMD3 register	R
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPSSR register indicates the sequence control status.

SPSS[1:0] Bits (Sequence Status)

During sequence control, these bits indicate the SPCMDn register (n = 0 to 3) that is currently referenced.

45.2.8 QSPI Bit Rate Register (SPBR)

Address(es): QSPI.SPBR 0008 9E0Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	SPBR7	SPBR6	SPBR5	SPBR4	SPBR3	SPBR2	SPBR1	SPBR0
Value after reset:	1	1	1	1	1	1	1	1

The SPBR register sets the bit rate that is used as the base. If this register is modified while the SPCR.SPE bit is 1, subsequent operation cannot be guaranteed.
The bit rate is determined by combinations of SPBR register settings and the bit settings in the SPCMDn.BRDV[1:0] bits (n = 0 to 3).
Setting this register to 0 is prohibited.
The equation for calculating the bit rate when SPBR is not 0 is given below. In the equation, m denotes an SPBR register setting (1 to 255), and N denotes bit settings (0 to 3) in the SPCMDn.BRDV[1:0] bits, and 2 × m × 2N denotes a division ratio.

$$\text{Bit rate} = \frac{f(PCLK)}{2 \times m \times 2^{N}}$$

Table 45.3 shows the relationship between SPBR register and SPCMDn.BRDV[1:0] bit settings and bit rates.

Table 45.3 SPBR Register and SPCMDn.BRDV[1:0] Bit Settings and Bit Rates

SPBR Register (m)	SPCMDn.BRDV[1:0] Bits (N)	Division Ratio	Bit Rate PCLK = 60 MHz
1	0	2	30.00 Mbps
2	0	4	15.00 Mbps
3	0	6	10.00 Mbps
4	0	8	7.50 Mbps
5	0	10	6.00 Mbps
6	0	12	5.00 Mbps
6	1	24	2.50 Mbps
6	2	48	1.25 Mbps
6	3	96	625 kbps
255	3	4080	14.70 kbps

Note: Set the SPBR register and the SPCMDn.BRDV[1:0] bits so that the division ratio of QSPCLK is 2 or more.

45.2.9 QSPI Data Control Register (SPDCR)

Address(es): QSPI.SPDCR 0008 9E0Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	TXDMY	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TXDMY	Dummy Data Transmission Enable	0: Disables dummy data transmission. 1: Enables dummy data transmission.	R/W

The SPDCR register enables or disables dummy data transmission.

TXDMY Bit (Dummy Data Transmission Enable)

This bit enables or disables dummy data transmission from the QMO pin in single-SPI mode when the transmit buffer is empty.

If the TXDMY bit is set to 1 when the transmit buffer is empty, setting the SPCR.SPE bit = 1 allows 0 to be output from the QMO pin as a dummy data. This enables starting of receive operation without writing transmit data to the transmit buffer.

If this bit is modified while the SPCR.SPE bit is 1, subsequent operation cannot be guaranteed.

This bit is invalid in dual-/quad-SPI operations.

45.2.10 QSPI Clock Delay Register (SPCKD)

Address(es): QSPI.SPCKD 0008 9E0Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SCKDL[2:0]	Clock Delay Setting	b2 b1 b0 0 0 0: 1.5 QSPCLK 0 0 1: 2.5 QSPCLK 0 1 0: 3.5 QSPCLK 0 1 1: 4.5 QSPCLK 1 0 0: 5.5 QSPCLK 1 0 1: 6.5 QSPCLK 1 1 0: 7.5 QSPCLK 1 1 1: 8.5 QSPCLK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPCKD register sets a period (clock delay) from the beginning of QSSL signal assertion to QSPCLK oscillation when the SPCMDn.SCKDEN bit (n = 0 to 3) is 1.
If this register is modified while the SPCR.SPE bit is 1, subsequent operation cannot be guaranteed.

SCKDL[2:0] Bits (Clock Delay Setting)

These bits set a period (clock delay) from the beginning of QSSL signal assertion to QSPCLK oscillation when the SPCMDn.SCKDEN bit is 1.

45.2.11 QSPI Slave Select Negation Delay Register (SSLND)

Address(es): QSPI.SSLND 0008 9E0Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SLNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SLNDL[2:0]	QSSL Negation Delay Setting	b2 b1 b0 0 0 0: 1 QSPCLK 0 0 1: 2 QSPCLK 0 1 0: 3 QSPCLK 0 1 1: 4 QSPCLK 1 0 0: 5 QSPCLK 1 0 1: 6 QSPCLK 1 1 0: 7 QSPCLK 1 1 1: 8 QSPCLK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SSLND register sets a period (QSSL negation delay) from a final QSPCLK edge to the negation of the QSSL signal during a serial transfer. when the SPCMDn.SLNDEN bit (n = 0 to 3) is 1. If this register is modified while the SPCR.SPE bit is 1, subsequent operation cannot be guaranteed.

SLNDL[2:0] Bits (QSSL Negation Delay Setting)

These bits set a period (QSSL negation delay) from a final QSPCLK edge to the negation of the QSSL signal during a serial transfer when the SPCMDn.SLNDEN bit (n = 0 to 3) is 1.

45.2.12 QSPI Next-Access Delay Register (SPND)

Address(es): QSPI.SPND 0008 9E0Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SPNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPNDL[2:0]	Next-Access Delay Setting	b2 b1 b0 0 0 0: 1 QSPCLK 0 0 1: 2 QSPCLK 0 1 0: 3 QSPCLK 0 1 1: 4 QSPCLK 1 0 0: 5 QSPCLK 1 0 1: 6 QSPCLK 1 1 0: 7 QSPCLK 1 1 1: 8 QSPCLK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPND register sets a period (next-access delay) from termination of a serial transfer to the beginning of the next serial transfer when the SPCMDn.SPNDEN bit (n = 0 to 3) is 1. If this register is modified while the SPCR.SPE bit is 1, subsequent operation cannot be guaranteed.

SPNDL[2:0] Bits (Next-Access Delay Setting)

These bits set a period (next-access delay) from termination of a serial transfer to the beginning of the next serial transfer when the SPCMDn.SPNDEN bit is 1.

45.2.13 QSPI Command Register n (SPCMDn) (n = 0 to 3)

Address(es): QSPI.SPCMD0 0008 9E10h, QSPI.SPCMD1 0008 9E12h, QSPI.SPCMD2 0008 9E14h, QSPI.SPCMD3 0008 9E16h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]				SSLKP	SPIMOD[1:0]		SPRW	BRDV[1:0]		CPOL	CPHA
Value after reset:	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	CPHA	QSPCLK Phase Setting*1	0: Data latch on odd edge, data shift on even edge 1: Data shift on odd edge, data latch on even edge	R/W
b1	CPOL	QSPCLK Polarity Setting	0: Positive (QSPCLK = 0 when idle) 1: Negative (QSPCLK = 1 when idle)	R/W
b3, b2	BRDV[1:0]	Bit Rate Frequency Division Setting *2	b3 b2 0 0: Base bit rate 0 1: Base bit rate divided by 2 1 0: Base bit rate divided by 4 1 1: Base bit rate divided by 8	R/W
b4	SPRW	SPI Read/Write Access	0: Write operation (QIO1 and QIO0/QIO3 to QIO0: Output) 1: Read operation (QIO1 and QIO0/QIO3 to QIO0: Input)	R/W
b6, b5	SPIMOD[1:0]	SPI Operating Mode	b6 b5 0 0: Single-SPI 0 1: Dual-SPI 1 0: Quad-SPI 1 1: Setting prohibited	R/W
b7	SSLKP	QSSL Signal Level Keeping	0: Negates all QSSL signals upon completion of transfer. 1: Keeps the QSSL signal level from the end of the transfer to the beginning of the next access.	R/W
b11 to b8	SPB[3:0]	Transfer Data Length Setting	b11 b8 0 0 0 0: 8 bits (1 byte) 0 0 0 1: 16 bits (2 bytes) 0 0 1 0: 32 bits (4 bytes) Settings other than above are prohibited.	R/W
b12	LSBF	LSB First	0: MSB first 1: LSB first	R/W
b13	SPNDEN	Next-Access Delay Enable	0: A next-access delay of 0 QSPCLK cycle 1: A next-access delay equal to SPND register settings.	R/W
b14	SLNDEN	QSSL Negation Delay Setting Enable	0: A QSSL negation delay of 0 QSPCLK cycle 1: A QSSL negation delay equal to SSLND register settings.	R/W
b15	SCKDEN	Clock Delay Setting Enable	0: A clock delay of 0 QSPCLK cycle 1: A clock delay equal to SPCKD register settings.	R/W

Note: When setting any or all of the clock delay period, QSSL negation delay period, and next-access delay period to 0, set the SSLKP bit to 1 to select the continuous access in which QSSL is not negated. Otherwise, operation cannot be guaranteed. For the method of setting the various delay periods for the continuous access in which QSSL is not negated, see below.

Note: For the continuous access in which QSSL is not negated, QSPCLK clock stopping is followed by the QSSL negation delay period, next-access delay period, and next command clock delay period, in this order. There are restrictions on setting the SCKDEN, SLNDEN, and SPNDEN bits at this time. For details, see the SCKDEN bit description.

Note 1. The first QSPCLK edge is treated as the first edge.

Note 2. Set the SPBR register and the SPCMDn.BRDV[1:0] bits (n = 0 to 3) so that the division ratio of QSPCLK is 2 or more.

The SPCMDn register (n = 0 to 3) is used to set a transfer format. This module sequentially references the SPCMDn register according to the settings in the SPSCR register, and executes the serial transfer that is set in the referenced the SPCMDn register.

If the SPCMDn register (n = 0 to 3) currently being referred to is modified while the SPE bit is 1, subsequent operation cannot be guaranteed. The SPCMDn register currently being referred to can be checked by reading the SPSSR register.

CPHA Bit (QSPCLK Phase Setting)

This bit sets a QSPCLK edge for latching and shifting data to be transferred. The phase should be the same as that of the device to be communicated with.
This MCU supports four SPI modes listed in the Table 45.4 in combination of the CPHA bit with the CPOL bit.

CPOL Bit (QSPCLK Polarity Setting)

This bit sets a QSPCLK polarity. The polarity should be same as that of the device to be communicated with.
This MCU supports four SPI modes listed in the Table 45.4 in combination of the CPOL bit with the CPHA bit.

Table 45.4 SPI Modes Definitions

SPI Mode	CPOL	CPHA
Mode 0	0	0
Mode 1	0	1
Mode 2	1	0
Mode 3	1	1

BRDV[1:0] Bits (Bit Rate Frequency Division Setting)

The settings of these bits and of the SPBR register together determine the bit rate. The base bit rate depends on the setting of the SPBR register . The setting of these bits selects division of the base bit rate by 1, 2, 4 or 8.

Note: In transmission, set the SPBR register and these bits so that the division ratio of QSPCLK is 2 or more.
Note: In communication and burst transfer using sequence control, the set value of the BRDV[1:0] bits cannot be changed for each of the commands that configure the sequence.

SPRW Bit (SPI Read/Write Access)

This bit sets an access direction in dual-/quad-SPI modes.
This bit is invalid in single-SPI mode.

SPIMOD[1:0] Bits (SPI Operating Mode)

These bits select the operating mode from single-, dual-, or quad-SPI.

SSLKP Bit (QSSL Signal Level Keeping)

This bit specifies whether the QSSL signal level for the current command is to be kept or not from the end of the transfer for the current command to the beginning of the transfer for the next command. Setting this bit to 1 enables a transition to the next access while the QSSL signal is kept asserted.

SPB[3:0] Bits (Transfer Data Length Setting)

These bits set the basic transfer data length for serial transfer. For LSB first transfer, the transfer data is reversed within the data width specified with these bits. The actual amount of data to be transferred is determined by multiplying the value set with these bits by the value set with the SPBMULn register (n = 0 to 3).

LSBF Bit (LSB First)

This bit sets the data format to MSB first or LSB first.

SPNDEN Bit (Next-Access Delay Enable)

This bit sets the period (next-access delay) from termination of a serial transfer to the beginning of the next serial transfer. If this bit is 0, this module sets the next-access delay to 0 QSPCLK cycle. If this bit is 1, this module starts next

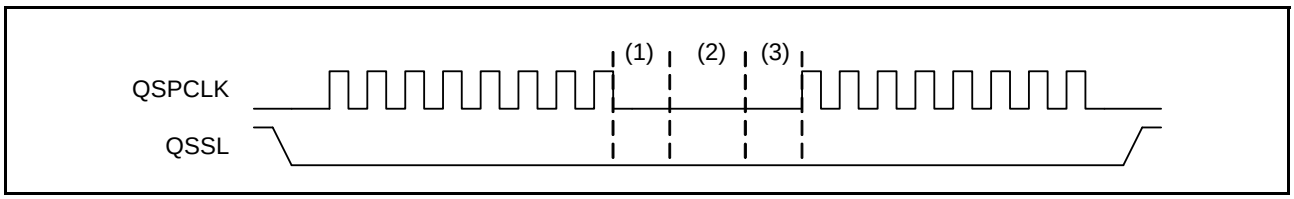
serial transfer in compliance with the SPND register settings. For the continuous access in which QSSL is kept asserted over the multiple commands, this bit can be set to 0 only when the pertinent command is not the last one. Otherwise, this bit should be set to 1.

SLNDEN Bit (QSSL Negation Delay Setting Enable)

This bit sets a period (QSSL negation delay) from QSPCLK oscillation stoppage to QSSL signal negation. If this bit is 0, this module sets the QSSL negation delay to 0 QSPCLK cycle. If this bit is 1, this module negates the QSSL signal in compliance with the SSLND register settings. For the continuous access in which QSSL is kept asserted over the multiplier commands, this bit can be set to 0 only when the pertinent command is not the last one. Otherwise, this bit should be set to 1.

SCKDEN Bit (Clock Delay Setting Enable)

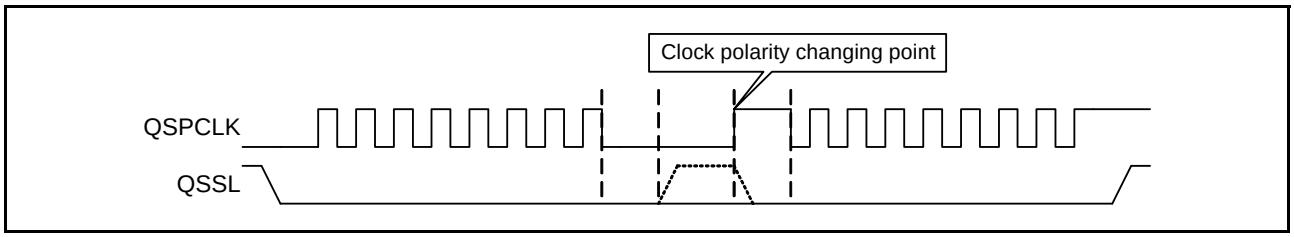
This bit sets a period (clock delay) from the beginning of QSSL signal assertion to QSPCLK oscillation. If this bit is 0, this module sets the clock delay to 0 QSPCLK cycle. If this bit is 1, this module starts QSPCLK oscillation in compliance with the SPCKD register settings. For the continuous access in which QSSL is kept asserted over the multiple commands, this bit can be set to 0 only when the pertinent command is the second or subsequent one. Otherwise, this bit should be set to 1.



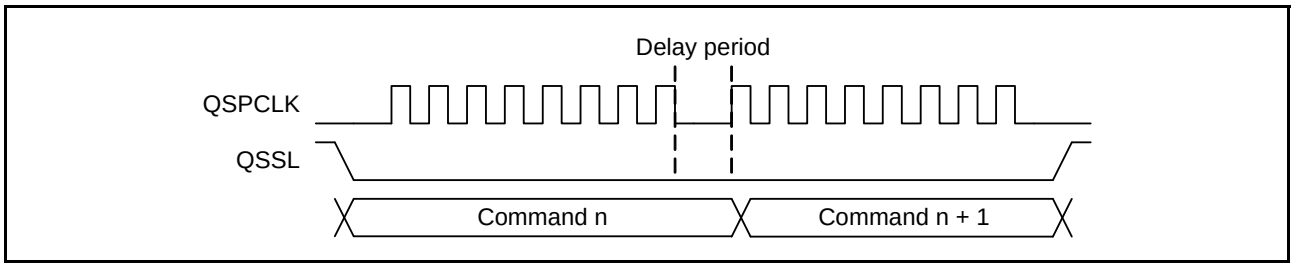
In the above figure, (1), (2), and (3) refer to the QSSL negation delay period, next-access delay period, and next command clock delay period, respectively.
Allowed setting is {(1), (2), (3)} = {1, 1, 1}, {1, 1, 0}, {1, 0, 0}, {0, 0, 0}. If a combination other than these is set, operation cannot be guaranteed.

Note: When changing the BRDV[1:0] bits or CPOL bit for each command for the continuous access in which the QSSL level is held, insert the QSSL negation delay period, next-access delay period, and clock delay period between commands. Otherwise, operation cannot be guaranteed.

Note: A clock polarity changing point may be detected as a clock edge if the CPOL bit is changed with the QSSL level held.



Note: When changing the SPIMOD[1:0] bits or CPHA bit for each command for the continuous access in which the QSSL level is held, insert one cycle or more between commands. Otherwise, operation cannot be guaranteed. (This also applies to write-to-read or read-to-write switching in dual-/quad-SPI modes.)
In the figure below, the data line is driven during the command n period if command n is for dual-/quad-SPI write access.



45.2.14 QSPI Buffer Control Register (SPBFCR)

Address(es): QSPI.SPBFCR 0008 9E18h

	b7	b6	b5	b4	b3	b2	b1	b0
	TXRST	RXRST	TXTRG[1:0]		TXTRGEX	RXTRG[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RXTRG[2:0]	Receive Buffer Data Triggering Number	b2 b0 0 0 0: 1 byte (31 bytes available) 0 0 1: 2 bytes (30 bytes available) 0 1 0: 4 bytes (28 bytes available) 0 1 1: 5 bytes (27 bytes available) 1 0 0: 8 bytes (24 bytes available) 1 0 1: 16 bytes (16 bytes available) 1 1 0: 24 bytes (8 bytes available) 1 1 1: 32 bytes (0 byte available)	R/W
b3	TXTRGEX	Transmit Buffer Data Triggering Number	b5 b3 0 0 0: 31 bytes (1 byte available) 0 1 0: 30 bytes (2 bytes available) 1 0 0: 28 bytes (4 bytes available) 1 1 0: 0 byte (32 bytes available)	R/W
b5, b4	TXTRG[1:0]		0 0 1: 24 bytes (8 bytes available) 0 1 1: 16 bytes (16 bytes available) 1 0 1: 8 bytes (24 bytes available) 1 1 1: 4 bytes (28 bytes available)	
b6	RXRST	Receive Buffer Data Reset	0: Allows the receive buffer normal operation. 1: Resets the receive buffer.	R/W
b7	TXRST	Transmit Buffer Data Reset	0: Allows the transmit buffer normal operation. 1: Resets the transmit buffer.	R/W

The SPBFCR register resets the number of data units in the transmit buffer (SPTXB) or receive buffer (SPRXB) and sets the number of triggering data units. If this register is modified, perform a dummy read after data is written, and then execute the subsequent instruction. In addition, if this register is modified while the SPCR.SPE bit is 1, subsequent operation cannot be guaranteed.

RXTRG[2:0] Bits (Receive Buffer Data Triggering Number)

These bits specify the timing at which the receive buffer full state is determined, that is when the SPSR.SPRFF flag in the status register is set. When the number of bytes of data in the receive buffer (SPRXB) is equal to or greater than the specified triggering number, the SPSR.SPRFF flag is set to 1. If these bits are modified, perform a dummy read after data is written, and then execute the subsequent instruction. These bits should be modified when the SPCR.SPE bit is 0.

TXTRG[1:0] Bits and TXTRGEX Bit (Transmit Buffer Data Triggering Number)

These bits specify the timing at which the transmit buffer empty state is determined, that is when the SPSR.SPRFF flag in the status register is set. When the number of bytes of data in the transmit buffer (SPTXB) is equal to or less than the specified triggering number, the SPSR.SPTEF flag is set to 1. If these bits are modified, perform a dummy read after data is written, and then execute the subsequent instruction. These bits should be modified when the SPCR.SPE bit is 0.

RXRST Bit (Receive Buffer Data Reset)

This bit invalidates receive data in the receive buffer and resets the receive buffer to an empty state. If this bit is modified, perform a dummy read after data is written, and then execute the subsequent instruction. This bit should be modified when the SPCR.SPE bit is 0.

TXRST Bit (Transmit Buffer Data Reset)

This bit invalidates transmit data in the transmit buffer and resets the transmit buffer to an empty state. If this bit is modified, perform a dummy read after data is written, and then execute the subsequent instruction. This bit should be modified when the SPCR.SPE bit is 0.

45.2.15 QSPI Buffer Data Count Set Register (SPBDCR)

Address(es): QSPI.SPBDCR 0008 9E1Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	TXBC[5:0]						—	—	RXBC[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	RXBC[5:0]	Receive Data Byte Counter	000000b indicates that SPRXB is empty. 100000b indicates that SPRXB is full.	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	TXBC[5:0]	Transmit Data Byte Counter	000000b indicates that SPTXB is empty. 100000b indicates that SPTXB is full.	R
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPBDCR register indicates the number of data units stored in the transmit buffer (SPTXB) and receive buffer (SPRXB). The upper eight bits indicate the number of transmit data units in the transmit buffer and the lower 8 bits indicate the number of receive data units in the receive buffer.

RXBC[5:0] Bits (Receive Data Byte Counter)

These bits indicate the number of receive data bytes in the receive data buffer (SPRXB).

TXBC[5:0] Bits (Transmit Data Byte Counter)

These bits indicate the number of transmit data bytes in the transmit data buffer (SPTXB).

45.2.16 QSPI Transfer Data Length Multiplier Setting Register n (SPBMULn) (n = 0 to 3)

Address(es): QSPI.SPBMUL0 0008 9E1Ch, QSPI.SPBMUL1 0008 9E20h, QSPI.SPBMUL2 0008 9E24h, QSPI.SPBMUL3 0008 9E28h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SPBMUL[31:24]								SPBMUL[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SPBMUL[15:8]								SPBMUL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	SPBMUL[31:0]	Transfer Data Length Multiplier Setting	These bits set the multiplier for the transfer data length.	R/W

The SPBMULn register (n = 0 to 3) sets the number of times to transfer the specific length of data defined by the SPCMDn.SPB[3:0] bits (n = 0 to 3). The SPBMUL0 to SPBMUL3 registers correspond to the SPCMD0 to SPCMD3 registers, respectively.

If the SPBMULn register corresponding to the SPCMDn register currently being referred to is modified while the SPCR.SPE bit is 1, subsequent operation cannot be guaranteed. The SPCMDn register currently being referred to can be checked by reading the SPSSR register.

SPBMUL[31:0] Bits (Transfer Data Length Multiplier Setting)

These bits set the multiplier for transfer data; that is, the number of times to transfer the specific length of data defined by the SPCMDn.SPB[3:0] bits (n = 0 to 3).

The actual amount of data to be transferred is determined by the SPCMDn.SPB[3:0] bits × the SPBMUL[31:0] bits.

Setting these bits to 0000 0000h allows the defined size of data to be transferred 4,294,967,296 times.

45.3 Operation

In this section, the serial transfer period means a period from the beginning of driving valid data to the fetching of the final valid data, and the idle period means the QSSL negation period.

45.3.1 Overview of Operations

This module is capable of serial transfers in single-/dual-/quad-SPI modes. Table 45.5 shows the features of each SPI mode.

Table 45.5 Features of Each SPI Mode

	Single-SPI	Dual-SPI	Quad-SPI
Number of data lines	One input line and one output line	Two I/O lines	Four I/O lines
Data line direction	Single-directional	Bidirectional	Bidirectional
Simultaneous transmission/ reception	Supported	Not supported	Not supported

Table 45.6 shows the overview of operation.

Table 45.6 Overview of Operation

Items	Specification
QSPCLK signal	Output
QMO signal (single-SPI)	Output
QMI signal (single-SPI)	Input
QIO1 and QIO0 (dual-SPI)/QIO3 to QIO0 (quad-SPI)	Input/output
QSSL signal	Output
Switching QSSL polarity	Supported
Transfer bit rate	Up to PCLK
Clock source	On-chip baud rate generator
Clock polarity	Positive/negative
Clock phase	Data latch at odd edge, data shift at even edge/ data shift at odd edge, data latch at even edge
Transfer bit order	MSB first/LSB first
Transfer data length	(8/16/32) × (1 to 4,294,967,296) bits
Burst transfer	Supported
QSPCLK delay control	Supported
QSSL negation delay control	Supported
Next-access delay control	Supported
Transfer start method	The specified length of data is in the transmit buffer when the SPCR.SPE bit = 1 There is a space for the length of data in the receive buffer when the SPCR.SPE bit = 1*1
Sequence control	Supported
Transmit buffer empty detection	Supported
Receive buffer full detection	Supported
QSSL negation detection	Supported

Note 1. In single-SPI operation, a transfer is started when the specified length of data is in the transmit buffer and there is a space for the specified length of data in the receive buffer. In dual-/quad-SPI operation, a transfer is started when the specified length of data is in the transmit buffer during transmission, and there is a space for the specified length of data in the receive buffer during reception.

45.3.2 Pin Control

This module automatically switches the pin states according to the status after write/read transfer in single-/dual-/quad-SPI mode. The status of the data pins (QMO/QMI/QIO[3:0]) in the idle state depends on the SPPCR.MOIFE, MOIFV, IO3FV, and IO2FV bit settings. Table 45.7 shows the pin states in single-SPI mode. Table 45.8 shows the pin states in dual-/quad-SPI modes.

Table 45.7 Pin States in Single-SPI Mode

Items	Single-SPI
QSSL	Output
QSPCLK	Output
QMO	Output
QMI	Input
QMO in the idle state	SPPCR.MOIFE bit = 0: Final output value SPPCR.MOIFE bit = 1: SPPCR.MOIFV bit setting value
QMI in the idle state	—
QIO2	SPPCR.IO2FV bit setting value output or not used
QIO3	SPPCR.IO3FV bit setting value output or not used

Table 45.8 Pin States in Dual-/Quad-SPI Mode

Items	Dual-SPI	Quad-SPI
QSSL	Output	Output
QSPCLK	Output	Output
QIO0	Input/output	Input/output
QIO1	Input/output	Input/output
QIO2	SPPCR.IO2FV bit setting value output or not used	Input/output
QIO3	SPPCR.IO3FV bit setting value output or not used	Input/output
QIO0 in the idle state	After writing: SPPCR.MOIFE bit = 0: Final output value SPPCR.MOIFE bit = 1: SPPCR.MOIFV bit setting value After reading: Hi-Z	After writing: SPPCR.MOIFE bit = 0: Final output value SPPCR.MOIFE bit = 1: SPPCR.MOIFV bit setting value After reading: Hi-Z
QIO1 in the idle state	After writing: SPPCR.MOIFE bit = 0: Final output value SPPCR.MOIFE bit = 1: SPPCR.MOIFV bit setting value After reading: Hi-Z	After writing: SPPCR.MOIFE bit = 0: Final output value SPPCR.MOIFE bit = 1: SPPCR.MOIFV bit setting value After reading: Hi-Z
QIO2 in the idle state	SPPCR.IO2FV bit setting value output or not used	After writing: SPPCR.MOIFE bit = 0: Final output value SPPCR.MOIFE bit = 1: SPPCR.MOIFV bit setting value After reading: Hi-Z
QIO3 in the idle state	SPPCR.IO3FV bit setting value output or not used	After writing: SPPCR.MOIFE bit = 0: Final output value SPPCR.MOIFE bit = 1: SPPCR.MOIFV bit setting value After reading: Hi-Z

45.3.3 Transfer Format

The SPI has four modes determined by setting the SPCMDn.CPOL and SPCMDn.CPHA bits (n = 0 to 3). Figure 45.2 shows the data latch/shift timing based on each mode in an 8-bit MSB first transfer. In Figure 45.2, L indicates the latch timing and S indicates the shift timing. DATA corresponds to QMI/QMO in single-SPI mode; QIO1 and QIO0 in dual-SPI mode; or QIO3 to QIO0 in quad-SPI mode. t_{ckd} indicates the clock delay period when the SPCMDn.SCKDEN bit is set to 1. Similarly, t_{slnd} indicates the QSSL negation delay period when the SPCMDn.SLNDEN bit is set to 1, and t_{spnd} indicates the next-access delay period when the SPCMDn.SPNDEN bit is set to 1.

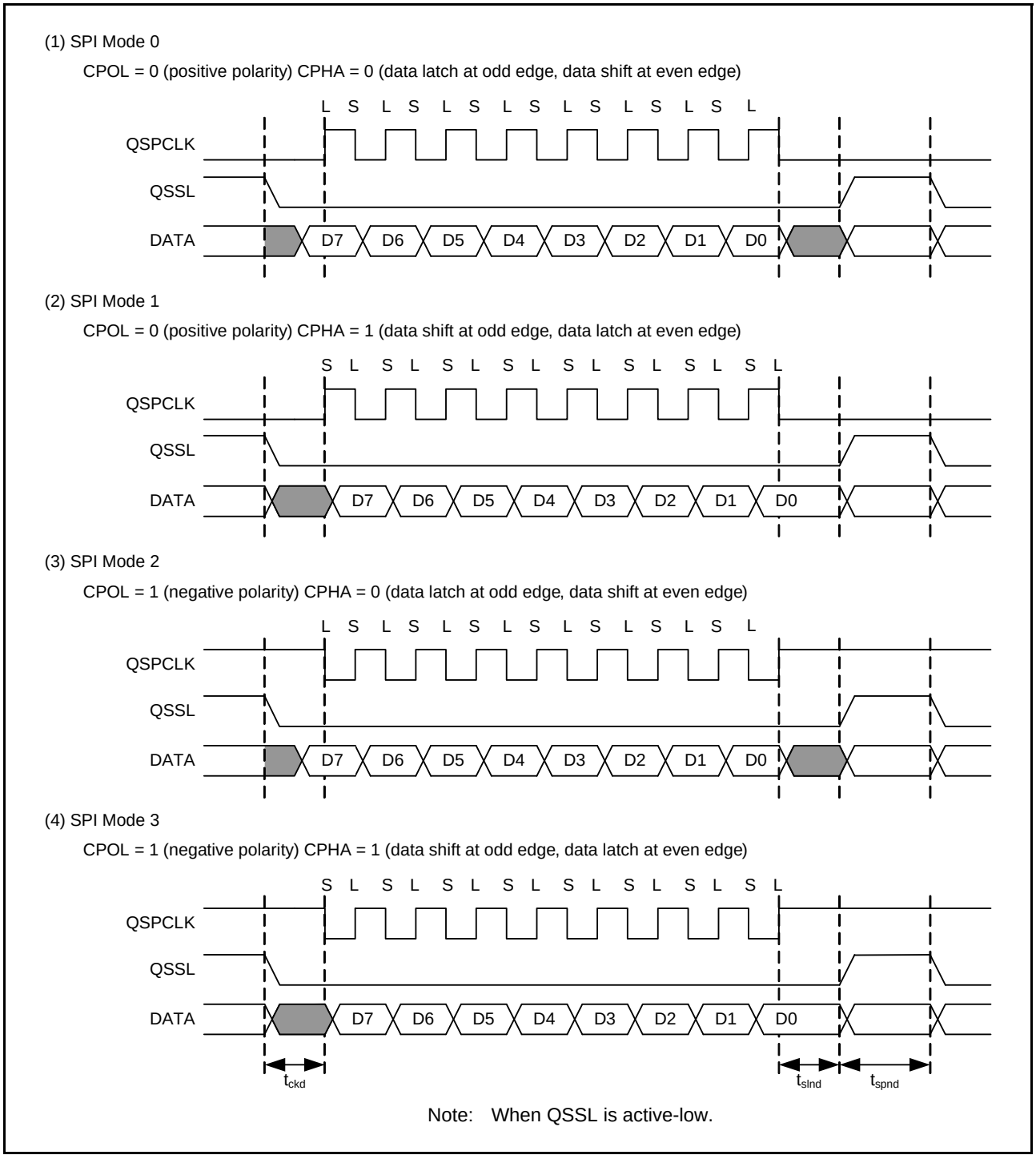


Figure 45.2 Clock Setting and Transfer Timing

Note that when the base bit rate is used, transmission and reception when the SPCMDn.CPHA bit (n = 0 to 3) = 0 is not available.
The following describes 8-bit MSB first transfer in single-/dual-/quad-SPI modes when the SPCMDn.CPOL bit = 0 and the SPCMDn.CPHA bit = 0.

(1) Single-SPI Mode

Figure 45.3 shows the transfer format in single-SPI mode. This mode provides transmission and reception simultaneously. Since one data line is used for serial communication both in transmission and reception, the communication speed is 1 bit per QSPCLK clock cycle. Transfer data is specified using the SPCMDn register (n = 0 to 3). For details of transfer data, see section 45.3.4, Transfer Data.

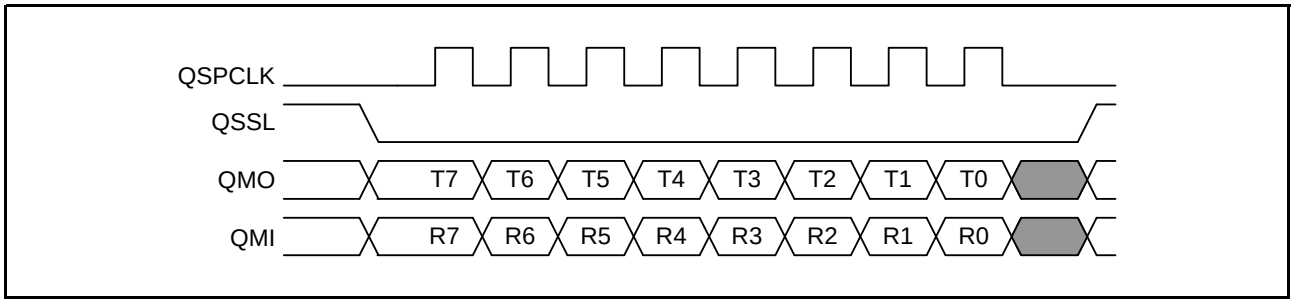


Figure 45.3 Transfer Format in Single-SPI Format

(2) Dual-SPI Mode

Figure 45.4 shows the transfer format in dual-SPI mode. This mode only provides operation of a single direction, that is, either transmission or reception. Transmission or reception can be set using the SPCMDn.SPRW bit (n = 0 to 3). Transmission is carried out by write operation and reception by read operation. The I/O directions of QIO1 and QIO0 are switched accordingly. Since two data lines are used for serial communication both in transmission and reception, the communication speed is 2 bits per QSPCLK clock cycle. The start bit of the transfer data is output from QIO1. Transfer data is specified using the SPCMDn register. For details of transfer data, see section 45.3.4, Transfer Data.

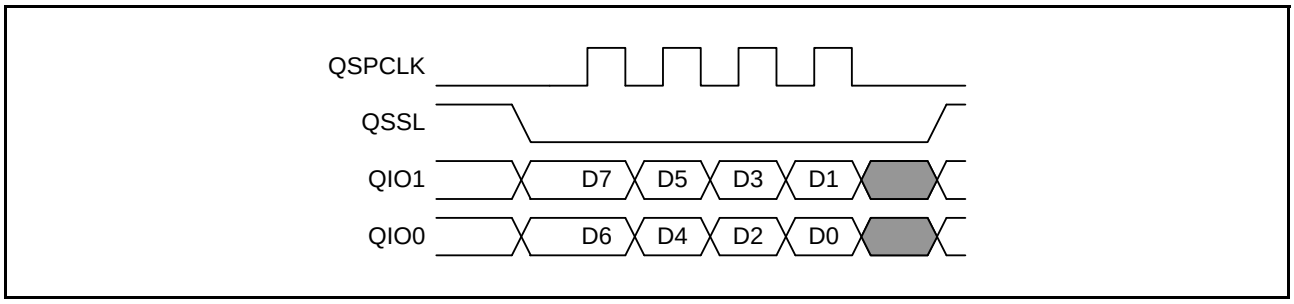


Figure 45.4 Transfer Format in Dual-SPI Format

(3) Quad-SPI Mode

Figure 45.5 shows the transfer format in quad-SPI mode. This mode provides operation of a single direction, that is, either transmission or reception. Transmission or reception can be set using the SPCMDn.SPRW bit (n = 0 to 3). Transmission and reception are carried out by writing and reading, respectively. The I/O directions of QIO3 to QIO0 are switched accordingly. Since four data lines are used for serial communication both in transmission and reception, the communication speed is 4 bits per QSPCLK clock cycle. The start bit of the transfer data is output from QIO3. Transfer data is specified using the SPCMDn register. For details of transfer data, see section 45.3.4, Transfer Data.

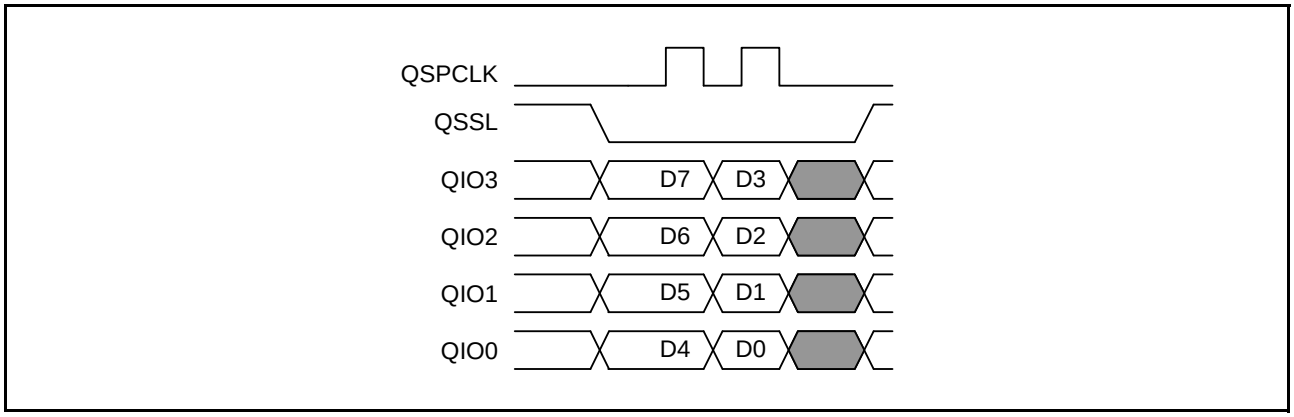


Figure 45.5 Transfer Format in Quad-SPI Format

45.3.4 Transfer Data

The data format is determined by the SPCMDn.SPB[3:0] bits and LSBF bit (n = 0 to 3) and the SPBMULn register (n = 0 to 3). This module treats the specified size of data beginning at the MSB of the transmit shift register as transmit data, and the specified length of data beginning at the LSB of the receive shift register as receive data, regardless of whether the actual arrangement is MSB or LSB first. The following sections describe MSB first and LSB first transfers in 32-bit, 16-bit, and 8-bit data units.

(1) MSB First Transfer (32-Bit Data)

Figure 45.6 shows the operation of the transmit buffer and transmit shift register, and the receive shift register and receive buffer when this module performs a 32-bit MSB-first data transfer.

For data transmission, the CPU, DTC, or DMAC writes 32-bit transmit data to the transmit buffer (SPTXB). If the transmit shift register is empty, this module copies the data in the transmit buffer to the transmit shift register, and fills the transmit shift register. When data transmission is started, this module outputs data beginning at the MSB (bit 31) of the transmit shift register, and when the QSPCLK clock cycle required for the serial transfer of 32 bits has passed, the transmit shift register becomes empty.

For data reception, data received from the data pin is stored in the receive shift register beginning at the LSB (bit 0). When the QSPCLK clock cycle required for the serial transfer of 32 bits has passed, the receive shift register becomes full. If the receive buffer (SPRXB) has a space for 32 bits or more, this module copies the 32-bit data beginning at the LSB from the receive shift register to the receive buffer, and empties the receive shift register. If the receive buffer does not have a space for 32 bits or more, data reception is not carried out. In order to start reception, the specified length of data should be read from the receive buffer to secure the space for 32 bits or more in the receive buffer.

In actual transfer, this operation is repeated for the number of times defined by the SPBMULn register (n = 0 to 3).

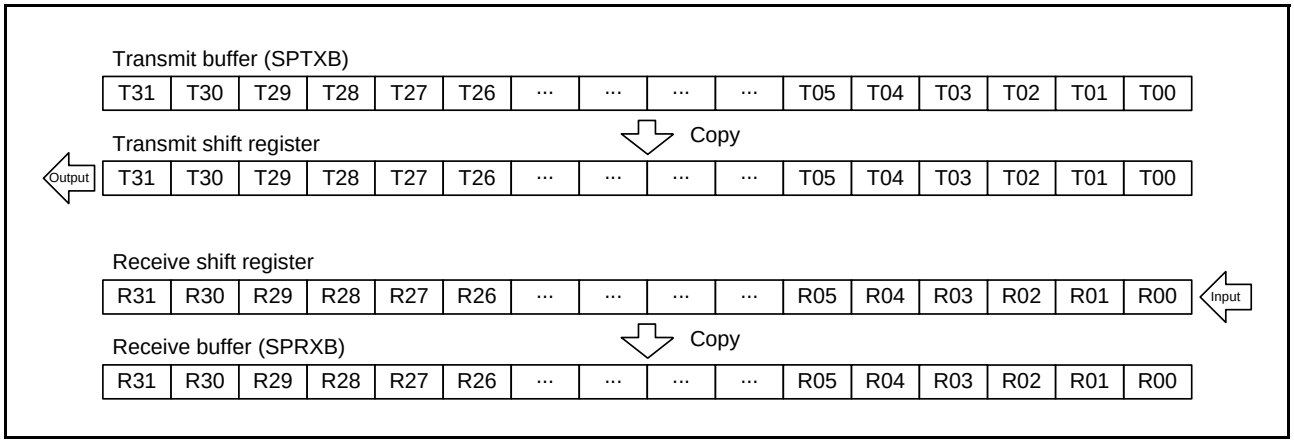


Figure 45.6 MSB First Transfer (32-Bit Data)

(2) MSB First Transfer (16-Bit Data)

Figure 45.7 shows the operation of the transmit buffer and transmit shift register, and the receive shift register and receive buffer when this module performs a 16-bit MSB-first data transfer.

For data transmission, the CPU, DTC, or DMAC writes 16-bit transmit data to the transmit buffer (SPTXB). If the transmit shift register is empty, this module copies the data with MSB-aligned in the transmit buffer to the transmit shift register, and fills the transmit shift register. When data transmission is started, this module outputs data beginning at the MSB (bit 31) of the transmit shift register, and when the QSPCLK clock cycle required for the serial transfer of 16 bits has passed, the transmit shift register becomes empty.

For data reception, data received from the data pin is stored in the receive shift register beginning at the LSB (bit 0). When the QSPCLK clock cycle required for the serial transfer of 16 bits has passed, the receive shift register becomes full. If the receive buffer (SPRXB) has a space for 16 bits or more, this module copies the 16-bit data beginning at the LSB from the receive shift register to the receive buffer, and empties the receive shift register. If the receive buffer does not have a space for 16 bits or more, data reception is not carried out. In order to start reception, the specified length of data should be read from the receive buffer to secure the space for 16 bits or more in the receive buffer.

In actual transfer, this operation is repeated for the number of times defined by the SPBMULn register (n = 0 to 3).

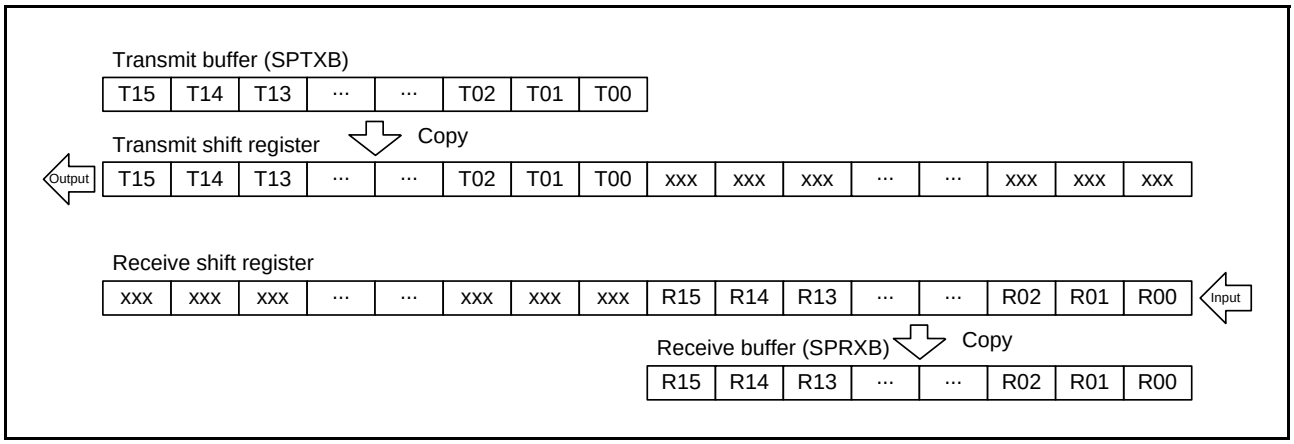


Figure 45.7 MSB First Transfer (16-Bit Data)

(3) MSB First Transfer (8-Bit Data)

Figure 45.8 shows the operation of the transmit buffer and transmit shift register, and the receive shift register and receive buffer when this module performs an 8-bit MSB-first data transfer.

For data transmission, the CPU, DTC, or DMAC writes 8-bit transmit data to the transmit buffer (SPTXB). If the transmit shift register is empty, this module copies the data with MSB-aligned in the transmit buffer to the transmit shift register, and fills the transmit shift register. When data transmission is started, this module outputs data beginning at the MSB (bit 31) of the transmit shift register, and when the QSPCLK clock cycle required for the serial transfer of 8 bits has passed, the transmit shift register becomes empty.

For data reception, data received from the data pin is stored in the receive shift register beginning at the LSB (bit 0). When the QSPCLK clock cycle required for the serial transfer of 8 bits has passed, the receive shift register becomes full. If the receive buffer (SPRXB) has a space for 8 bits or more, this module copies the 8-bit data beginning at the LSB from the receive shift register to the receive buffer, and empties the receive shift register. If the receive buffer does not have a space for 8 bits or more, data reception is not carried out. In order to start reception, data for the specified length of data should be read from the receive buffer to secure the space for 8 bits or more in the receive buffer.

In actual transfer, this operation is repeated for the number of times defined by the SPBMULn register (n = 0 to 3).

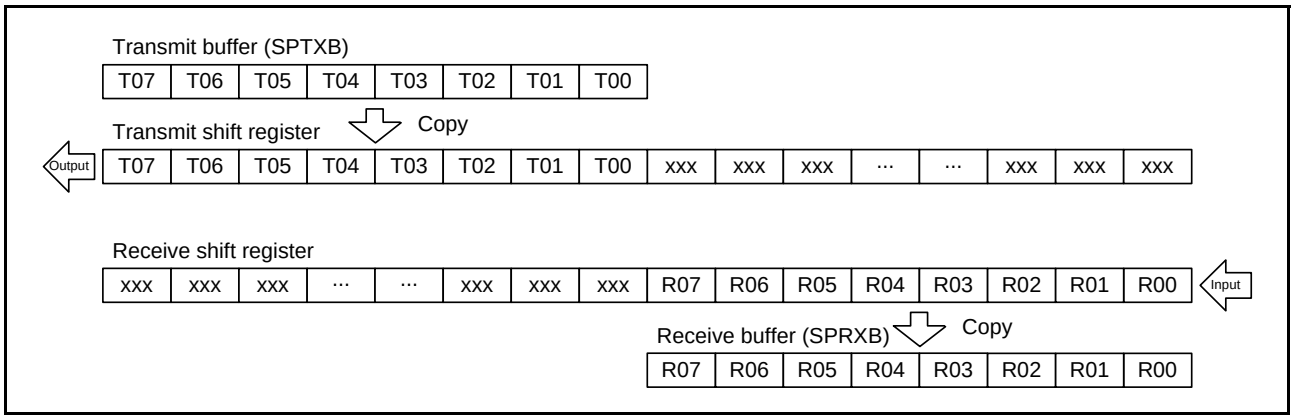


Figure 45.8 MSB First Transfer (8-Bit Data)

(4) LSB First Transfer (32-Bit Data)

Figure 45.9 shows the operation of the transmit buffer and transmit shift register, and the receive shift register and receive buffer when this module performs a 32-bit LSB-first data transfer.

For data transmission, the CPU, DTC, or DMAC writes 32-bit transmit data to the transmit buffer (SPTXB). If the transmit shift register is empty, this module reverses the bit order in the 32-bit transmit data, copies it to the transmit shift register, and fills the transmit shift register. When data transmission is started, this module outputs data beginning at the MSB (bit 31) of the transmit shift register, and when the QSPCLK clock cycle required for the serial transfer of 32 bits has passed, the transmit shift register becomes empty.

For data reception, data received from the data pin is stored in the receive shift register beginning at the LSB (bit 0). When the QSPCLK clock cycle required for the serial transfer of 32 bits has passed, the receive shift register becomes full. If the receive buffer (SPRXB) has a space for 32 bits or more, this module reverses the order of the bits of the 32-bit data, copies it beginning at the LSB from the receive shift register to the receive buffer, and empties the receive shift register. If the receive buffer does not have a space for 32 bits or more, data reception is not carried out. In order to start reception, the specified length of data should be read from the receive buffer to secure the space for 32 bits or more in the receive buffer.

In actual transfer, this operation is repeated for the number of times defined by the SPBMULn register (n = 0 to 3).

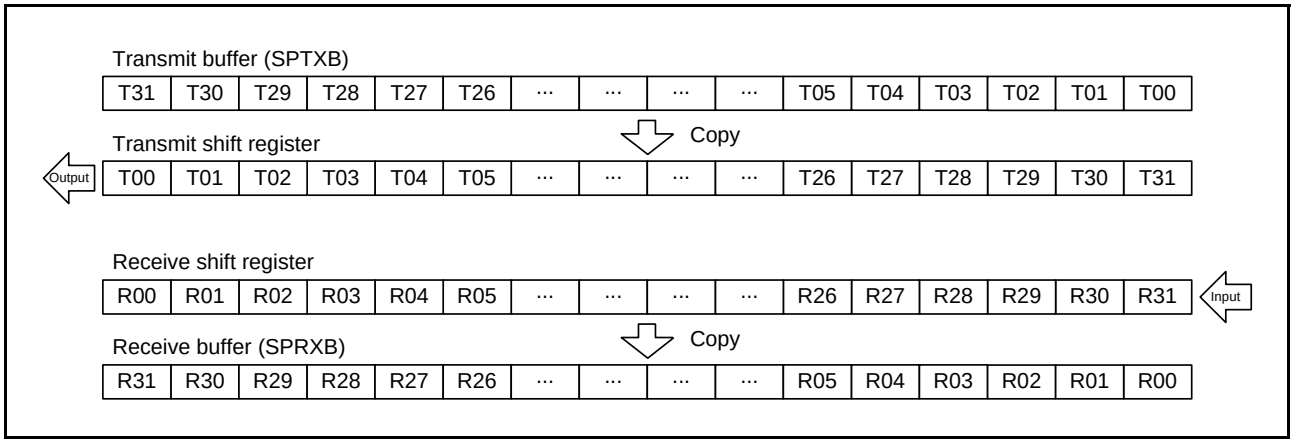


Figure 45.9 LSB First Transfer (32-Bit Data)

(5) LSB First Transfer (16-Bit Data)

Figure 45.10 shows the operation of the transmit buffer and transmit shift register, and the receive shift register and receive buffer when this module performs a 16-bit LSB-first data transfer.

For data transmission, the CPU, DTC, or DMAC writes 16-bit transmit data to the transmit buffer (SPTXB). If the transmit shift register is empty, this module reverses the bit order in the 16-bit transmit data, copies it with MSB-aligned to the transmit shift register, and fills the transmit shift register. When data transmission is started, this module outputs data beginning at the MSB (bit 31) of the transmit shift register, and when the QSPCLK clock cycle required for the serial transfer of 16 bits has passed, the transmit shift register becomes empty.

For data reception, data received from the data pin is stored in the receive shift register beginning at the LSB (bit 0). When the QSPCLK clock cycle required for the serial transfer of 16 bits has passed, the receive shift register becomes full. If the receive buffer (SPRXB) has a space for 16 bits or more, this module reverses the bit order in the 16-bit data, copies it beginning at the LSB from the receive shift register to the receive buffer, and empties the receive shift register. If the receive buffer does not have a space for 16 bits or more, data reception is not carried out. In order to start reception, the specified length of data should be read from the receive buffer to secure the space for 16 bits or more in the receive buffer.

In actual transfer, this operation is repeated for the number of times defined by the SPBMULn register (n = 0 to 3).

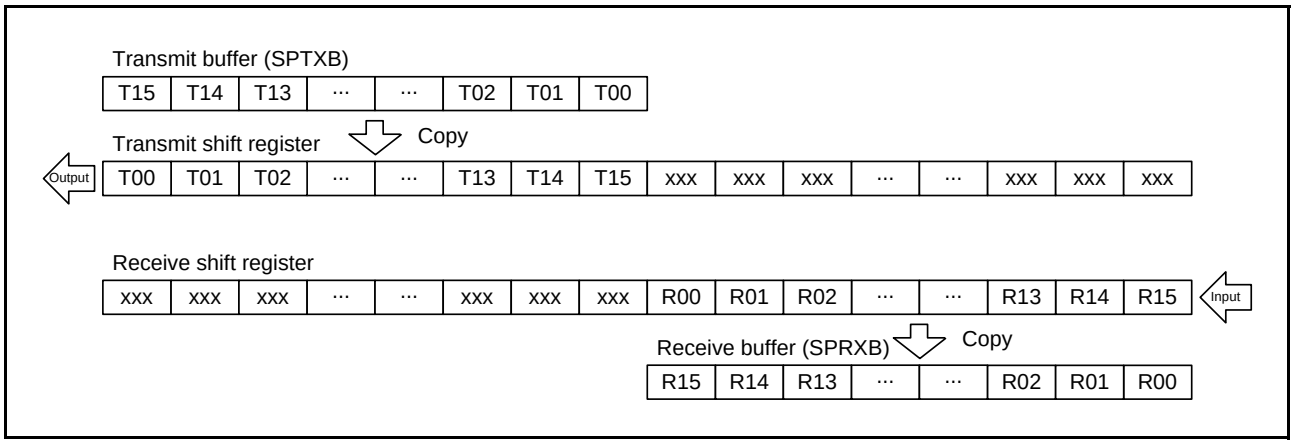


Figure 45.10 LSB First Transfer (16-Bit Data)

(6) LSB First Transfer (8-Bit Data)

Figure 45.11 shows the operation of the transmit buffer and transmit shift register, and the receive shift register and receive buffer when this module performs an 8-bit LSB-first data transfer.
For data transmission, the CPU, DTC, or DMAC writes 8-bit transmit data to the transmit buffer (SPTXB). If the transmit shift register is empty, this module reverses the bit order in the 8-bit transmit data, copies it with MSB-aligned to the transmit shift register, and fills the transmit shift register. When data transmission is started, this module outputs data beginning at the MSB (bit 31) of the transmit shift register, and when the QSPCLK clock cycle required for the serial transfer of 8 bits has passed, the transmit shift register becomes empty.
For data reception, data received from the data pin is stored in the receive shift register beginning at the LSB (bit 0). When the QSPCLK clock cycle required for the serial transfer of 8 bits has passed, the receive shift register becomes full. If the receive buffer (SPRXB) has a space for 8 bits or more, this module reverses the bit order in the 8-bit data, copies it beginning at the LSB from the receive shift register to the receive buffer, and empties the receive shift register. If the receive buffer does not have a space for 8 bits or more, data reception is not carried out. In order to start reception, the specified length of data should be read from the receive buffer to secure the space for 8 bits or more in the receive buffer. In actual transfer, this operation is repeated for the number of times defined by the SPBMULn register (n = 0 to 3).

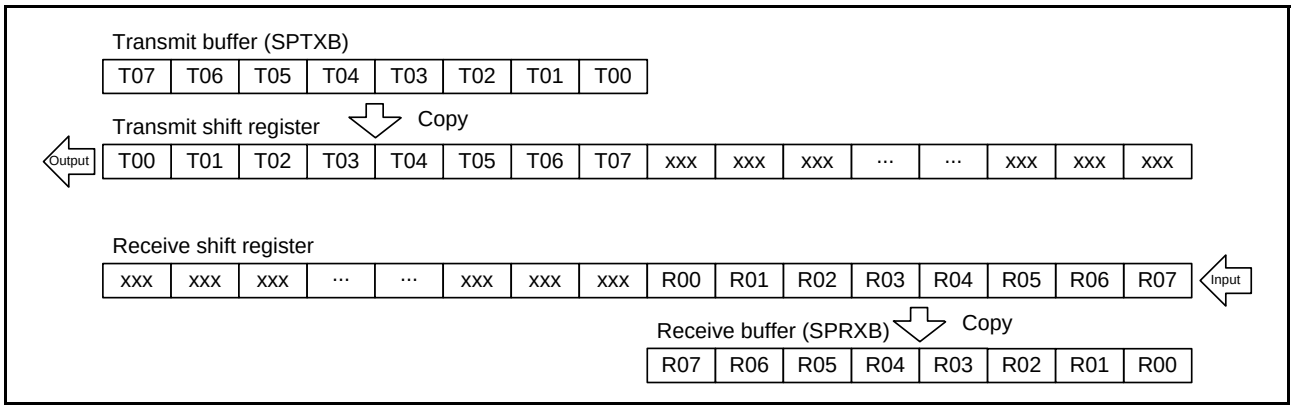


Figure 45.11 LSB First Transfer (8-Bit Data)

45.3.5 Non-Normal Transfer Operations

In the normal serial transfer, the data written from the SPDR register to the transmit buffer is serially transmitted, and the serially received data can be read from the receive buffer of the SPDR register. If access is made to the SPDR register, depending on the status of the transmit buffer/receive buffer, in some cases non-normal transfers can be executed. Table 45.9 shows the relationship between non-normal transfer operations.

Table 45.9 Relationship between Non-Normal Transfer Operations

	Occurrence Condition	Operation
A	The SPDR register is written when the transmit buffer is full.	Missing write data.
B	The SPDR register is read when the receive buffer is empty.	The output data is undefined.

On operation A shown in Table 45.9, whether the SPDR register can be written to or not can be checked using the SPBDCR.TXBC[5:0] bits.
Similarly, on operation B shown in Table 45.9, whether the valid data is stored in the receive buffer or not can be checked by reading the SPBDCR.RXBC[5:0] bits.

45.3.6 Initialization

If 0 is written to the SPCR.SPE bit, this module disables the module function, and initializes a part of the module function. When a power-on reset is generated, this module initializes all of the module function.
When the SPCR.SPE bit is cleared to 0, this module performs the following initialization:

- Suspending any serial transfer that is being executed
- Initializing the transmit shift register and the receive shift register
- Initializing the internal state machine
- Initializing the sequence
- Initializing the SPSR.TREND flag

Initialization by the clearing of the SPCR.SPE bit to 0 does not initialize the control bits of this module and the transmit/receive buffer. For this reason, this module can be started in the same transfer state as prior to the initialization if the SPCR.SPE bit is re-set to 1. However, clearing the SPCR.SPE bit to 0 initializes the transmit shift register and the receive shift register, thus the data that is being transferred will be discarded.

45.3.7 SPI Operation

The operating modes of this module are listed below.

- Single-SPI mode
- Dual-SPI mode/quad-SPI mode

The operation in each mode is described below.

(1) Single-SPI Mode

(a) Starting Serial Transfer

The serial transfer start conditions are: there is the specified length of data in the transmit buffer; and there is space for the specified length of data in the receive buffer.

(b) Terminating Serial Transfer

Irrespective of the clock setting, this module terminates the serial transfer after transmitting an QSPCLK edge corresponding to the final sampling timing. After the serial transfer is completed, receive data is copied from the receive shift register to the receive buffer. If there is not enough space for the specified length of data in the receive buffer after receive data is copied from the receive shift register to the receive buffer, another serial transfer will not be performed.

(c) Sequence Control

In single-SPI mode, according to the sequence length that is assigned to the SPSCR register, this module makes up a sequence comprised of a part or all of the SPCMDn register (n = 0 to 3) and the SPBMULn register (n = 0 to 3). This module contains a pointer to the SPCMDn register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR register.

When the SPCR.SPE bit is set to 1 and the function of this module is enabled, this module loads the pointer to the commands in the SPCMD0 register, and incorporates the SPCMD0 and SPBMUL0 register settings into the transfer format at the beginning of serial transfer. This module increments the pointer each time the next-access delay period for a data transfer that corresponds to the referenced the SPCMDn register ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, this module sets the pointer in the SPCMD0 register, and in this manner the sequence is executed repeatedly.

The following items are set in the SPCMDn register: basic transfer data length, MSB or LSB first, clock settings, some of the bit rate settings, SPI transfer mode and transfer direction (only in dual-/quad-SPI modes), whether QSSL level is held, a clock delay period, an QSSL negation delay period, and a next-access delay period. The total amount of data to be transferred is determined by multiplying the basic length of data to be transferred by the value set with the SPBMULn register.

Figure 45.12 shows an operation example when the SPSCR register is set to 02h, and the sequence is configured based on SPCMD0 to SPCMD2 register settings. In Figure 45.12, shaded areas of QMO/QMI indicate invalid data. Periods (1) to (3) in the figure indicate the followings.

(1) Clock delay period (SPCKD) setting value = 000b (1.5 QSPCLK cycles)
(2) QSSL negation delay period (SSLND) setting value = 000b (1 QSPCLK cycle)
(3) Next-access delay period (SPND) setting value = 000b (1 QSPCLK cycle)

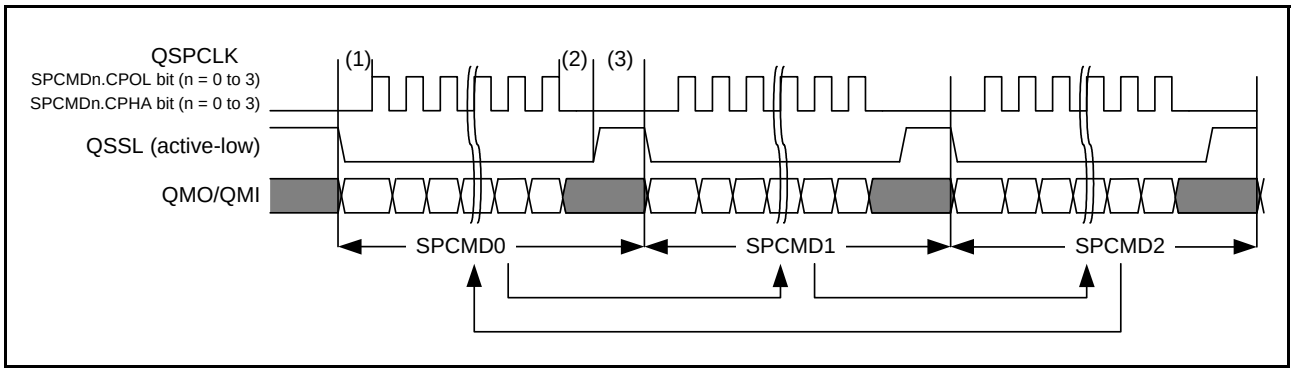


Figure 45.12 Sequence Control Operation

(d) Burst Transfer

This module can execute burst transfer with the following two methods in single-SPI mode.
One method uses the SPCMDn.SPB[3:0] bits (n = 0 to 3) and the SPBMULn register (n = 0 to 3). Setting SPCMDn.SPB[3:0] bits to select 8, 16, or 32 bits and setting the SPBMULn register to select 1 through 4,294,967,296 allows the specified length of data to be continuously transferred for the specified times, where the length is specified by SPCMDn.SPB[3:0] bits and the number of times is specified by the SPBMULn register. However, if the transmit buffer (SPTXB) becomes empty during transfer, or the receive buffer (SPRXB) has no longer a space enough to receive the specified length of data defined by SPCMDn.SPB[3:0] bits, the clock is stopped until transfer is resumed. Figure 45.13 shows a burst transfer example in which SPCMDn.SPB[3:0] bits are set to select 32 bits and the SPBMULn register to select four times thus specifying 128 bits as a total transfer data amount. The following describes operations (1) to (4) in the figure.

(1) First 32-bit data transfer
(2) Second 32-bit data transfer
(3) When the transmit buffer becomes empty or the receive buffer has no longer a space for 32 or more bits, the clock is stopped. Here, the QMO continues outputting the previous value. When data is written to the transmit buffer or an enough space is created in the receive buffer, the clock output is resumed to restart transfer.
(4) Third and fourth 32-bit data transfer

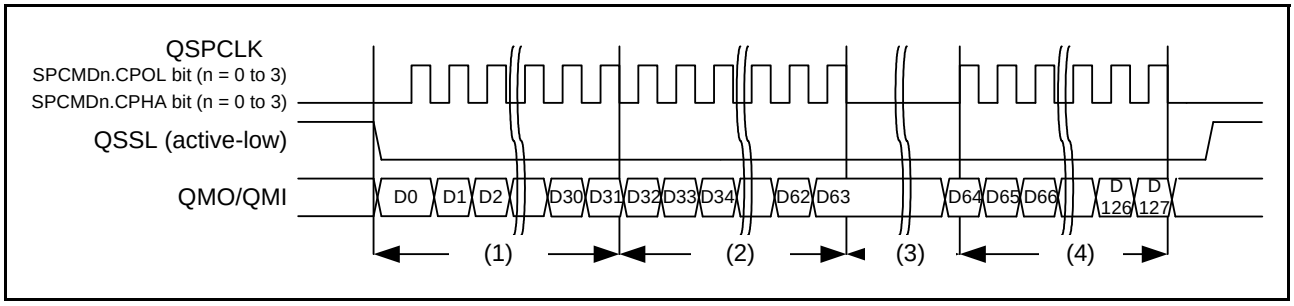


Figure 45.13 Burst Transfer Example in which Total Transfer Data Amount is 128 Bits (Single-SPI Mode)

In the other method, QSSL is kept asserted after a serial transfer is completed until the next serial transfer. Setting the SPCMDn.SSLKP bit (n = 0 to 3) allows the QSSL signal to be kept asserted after the transfer corresponding to the SPCMDn register is completed until the next transfer. Figure 45.14 shows a burst transfer example in which the QSSL signal level keeping function is used. The following describes operations (1) to (6) in the figure.

(1) Clock delay period according to the SPCMD0 register setting. The setting should be made so that the delay period should be at least 1.5 QSPCLK cycles for the first transfer in the burst transfer.
(2) QSSL negation delay period according to the SPCMD0 register setting. Since the SPCMD0.SSLKP bit is set to 1, QSSL is not negated even after QSSL negation delay period is over. The QSSL negation delay period depends on the SPCMD0.SLNDEN bit setting. When the SPCMD0.SLNDEN bit is 1, the QSSL negation delay period is determined by the SSLND register setting, and the delay period is 0 QSPCLK cycle when the SPCMD0.SLNDEN bit is 0.
(3) Next-access delay period according to the SPCMD0 register setting. Since the SPCMD0.SSLKP bit is set to 1, QSSL is not negated even during next-access delay period. The next-access delay period depends on the SPCMD0.SPNDEN bit setting. When the SPCMD0.SPNDEN bit is 1, the next-access delay period is determined by the SPND register setting, and the delay period is 0 QSPCLK cycle when the SPCMD0.SPNDEN bit is 0.
(4) Clock delay period according to the SPCMD1 register setting. The clock delay period depends on the SPCMD1.SCKDEN bit setting. When the SPCMD1.SCKDEN bit is 1, the clock delay period is determined by the SPCKD register setting, and the delay period is 0 QSPCLK cycle when the SPCMD1.SCKDEN bit is 0.
(5) QSSL negation delay period according to the SPCMD1 register setting. The setting should be made so that the delay period should be at least one QSPCLK cycle for the last transfer in the burst transfer. Since the SPCMD1.SSLKP bit is set to 0, QSSL is negated after QSSL negation delay period is over.
(6) Next-access delay period according to the SPCMD1 register setting. The setting should be made so that the delay period should be at least one QSPCLK cycle for the last transfer in the burst transfer. Set the SPCMD1.SSLKP bit to 0 to negate QSSL.

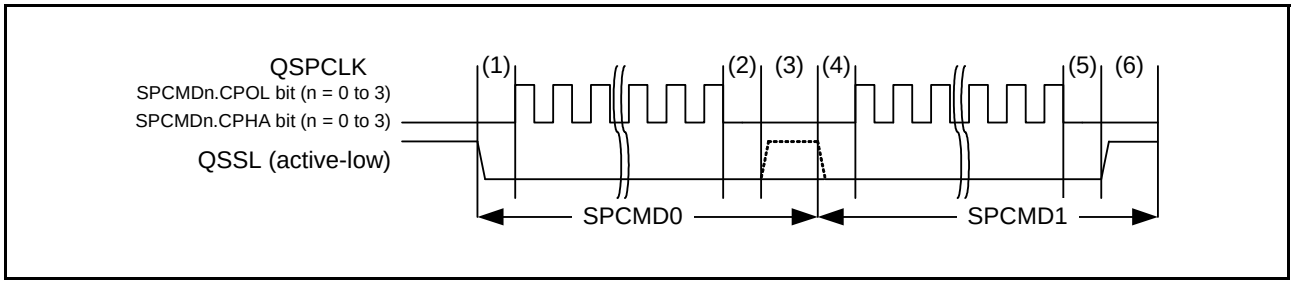


Figure 45.14 Burst Transfer Example in which QSSL Signal Level Keeping Function is Used (Single-SPI Mode)

Note the following when specifying a burst transfer using this method.
Periods (2) to (4) should be inserted without fail when changing the clock polarity through command update.

(e) Initialization Flowchart

Figure 45.15 is a flowchart illustrating an example of initialization in SPI operation when this module is used in single-SPI mode. For a description of how to set up the interrupt controller, DTC, or DMAC, see the descriptions given in the corresponding sections.

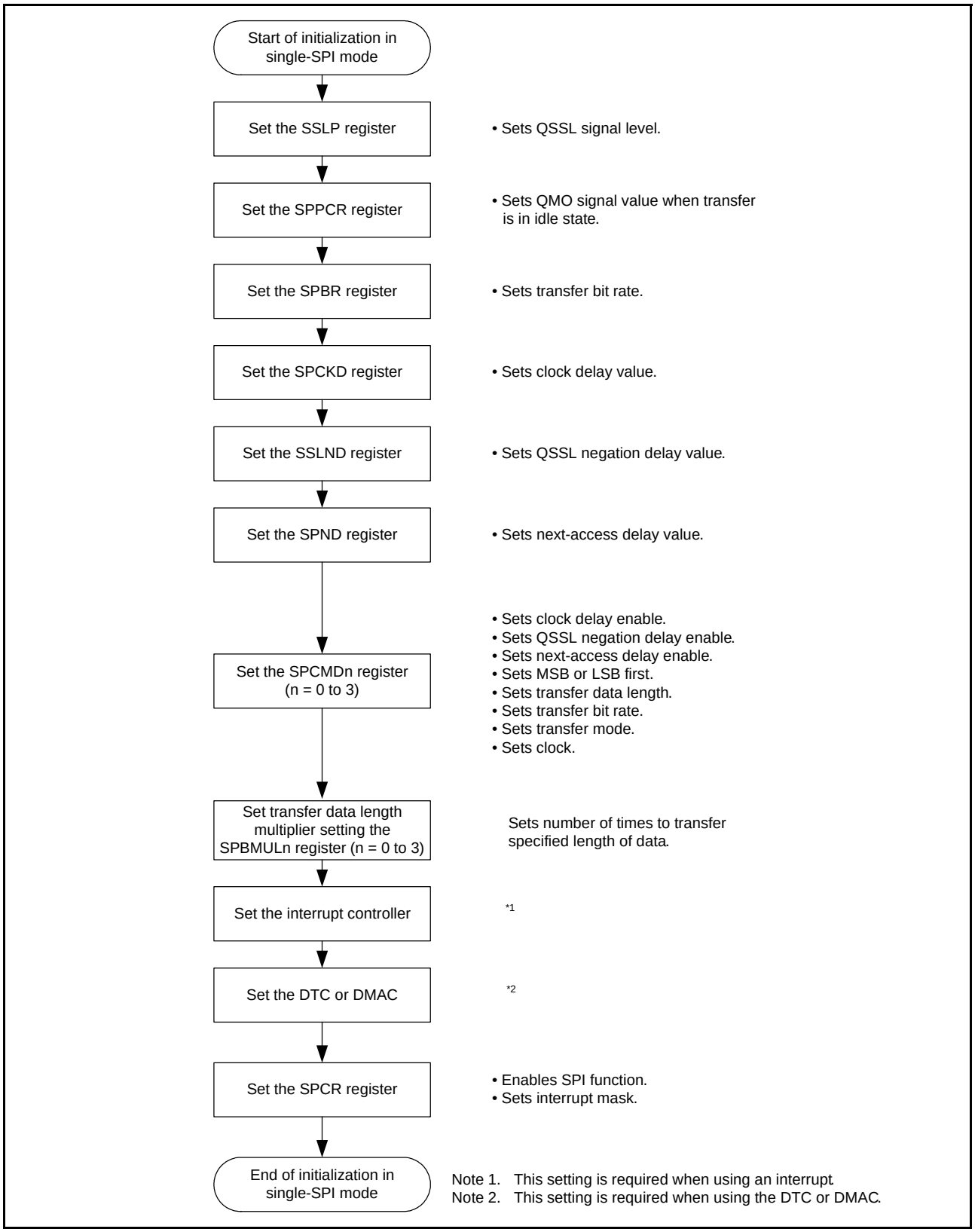


Figure 45.15 Example of Initialization Flowchart in Single-SPI Mode

(f) Transfer Operation Flowchart

Figure 45.16 is a flowchart illustrating a transfer in SPI operation when this module is used in single-SPI mode. Burst transfer by setting the transfer data length is also executed based on this flowchart.

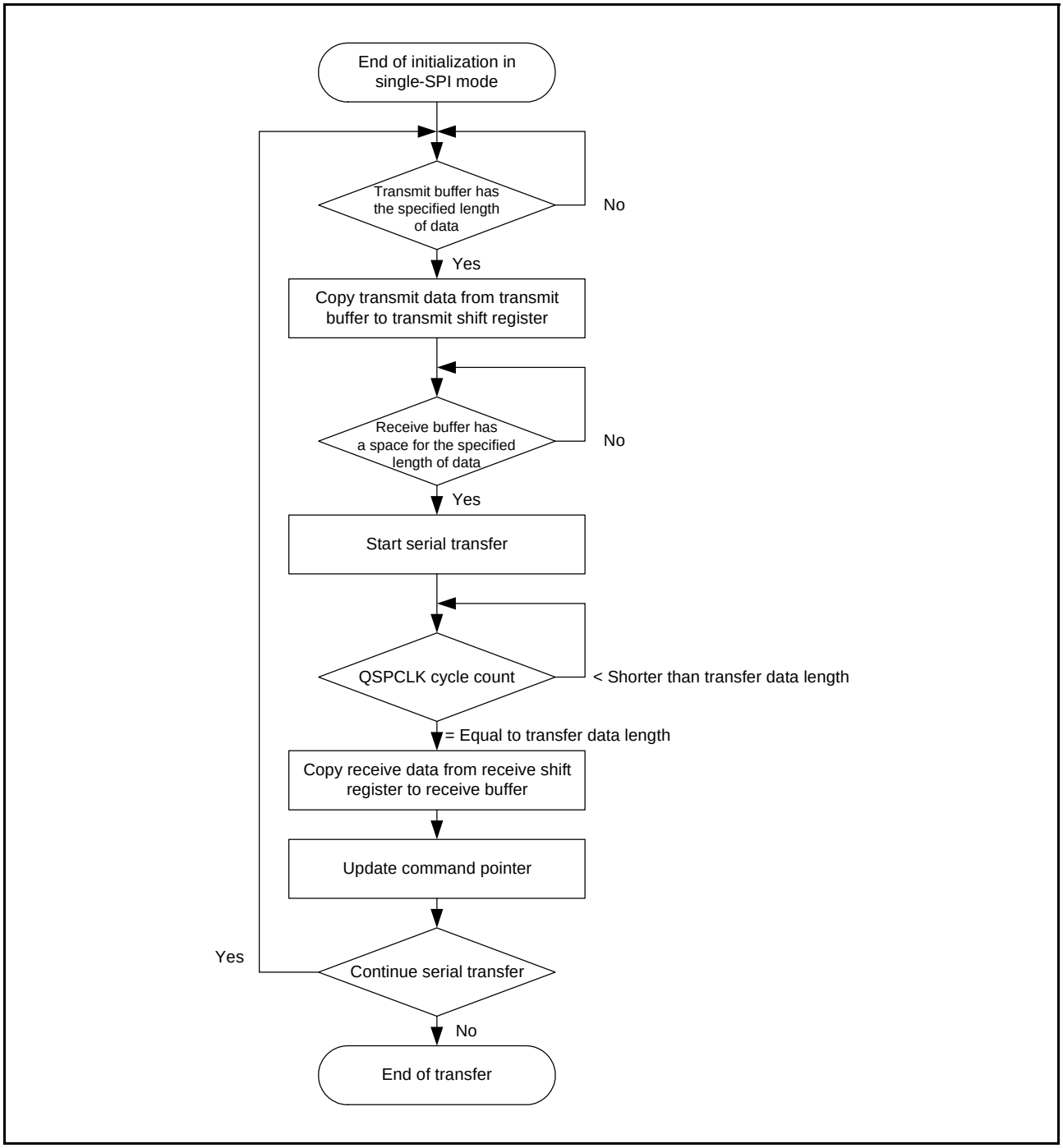


Figure 45.16 Transfer Operation Flowchart in Single-SPI Mode

(2) Dual-SPI/Quad-SPI Mode

(a) Starting Serial Transfer

In dual-/quad-SPI modes, the serial transfer start condition is different depending on the data transfer direction (transmission or reception).
In data transmission, the serial transfer start condition is that there is the specified length of data in the transmit buffer.
In data reception, the serial transfer start condition is that there is a space for the specified length of data in the receive buffer.

(b) Terminating Serial Transfer

Irrespective of data transmission or reception, this module terminates the serial transfer after transmitting an QSPCLK edge corresponding to the final sampling timing.
During idle cycles in dual-/quad-SPI modes, the I/O pins are controlled differently depending on whether it is after write or read operation. The I/O pins output either the last output data or the fixed level depending on the register setting after write operation, whereas the I/O pins are driven to the Hi-Z state after read operation. Figure 45.17 shows an example of the pin states after quad-SPI mode access is completed. The following describes operations (1) and (2) in the figure.

(1) During write operation, QIO0 to QIO3 serve as output pins. Thus, when QSSL is negated upon completion of write operation, QIO0 to QIO3 output different values depending on the value of SPPCR.MOIFE bit. The I/O pins output the level specified by the SPPCR.MOIFV bit when the SPPCR.MOIFE bit is 1, whereas the I/O pins output the last output data when SPPCR.MOIFE bit is 0.
(2) During read operation, QIO0 to QIO3 serve as input pins. Thus, when QSSL is negated upon completion of read operation, QIO0 to QIO3 are driven to Hi-Z state irrespective of the values of SPPCR.MOIFE and SPPCR.MOIFV bits. In dual-/quad-SPI read operations, receive operation is continued as long as the receive buffer has an enough space for the receive data. To terminate read operation, clear the SPCR.SPE bit to 0 after receiving the required length of data, or execute write operation for the last sequence to empty the transmit buffer.

For details on the pin control in dual-/quad-SPI modes, see section 45.3.2, Pin Control.

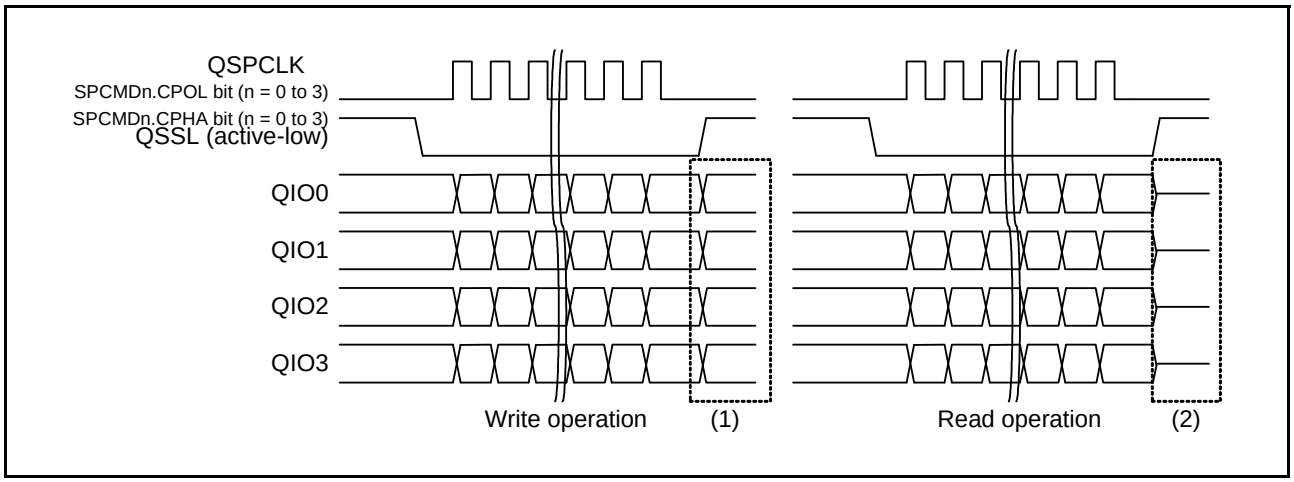


Figure 45.17 Example of Pin States after Quad-SPI Mode Transfer is Completed

(c) Sequence Control

As with the single-SPI mode, in dual-/quad-SPI modes, according to the sequence length that is assigned to the SPSCR register, this module makes up a sequence comprised of the SPCMDn register (n = 0 to 3) and the SPBMULn register (n = 0 to 3). For details on operation, refer to section 45.3.7 (1) (c), Sequence Control.
Dual-/quad-SPI modes only provide operation of a single direction, that is, either transmission or reception for serial transfer. Transmission or reception is set using the the SPCMDn.SPRW bit. One of the three operating modes including dual-SPI mode, quad-SPI mode, and single-SPI mode is set using the SPCMDn.SPIMOD[1:0] bits. Combining these bits allow switching single-SPI mode, dual-SPI mode transmission/reception, and quad-SPI mode transmission/reception to control sequence. Figure 45.18 shows an example of sequence configuration with transfer mode switching.
For sequence control, use either of the following methods.

(1) When a sequence is started by write operation and then read operation is performed, terminate the sequence upon completion of read operation. Do not perform write operation again after read operation.

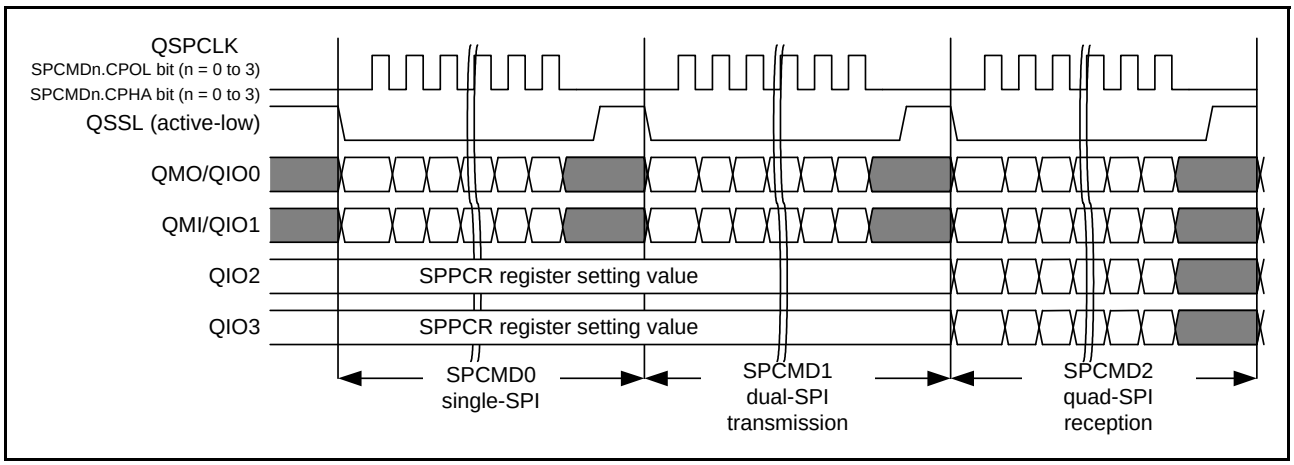


Figure 45.18 Example of Sequence Configuration with Transfer Mode Switching

(2) Note the following when configuring a sequence in dual-/quad-SPI modes. When all the commands configuring a sequence are dual-/quad-SPI read operations, the sequential operation is continued as long as the receive buffer has an enough space for the receive data. Figure 45.19 and Figure 45.20 show examples of sequence configuration with transfer mode switching.

To terminate read operation, there are the following two methods.

[Termination Method 1]

Clear the SPCR.SPE bit to 0 after receiving the required length of data. The receive command is continuously executed, allowing QSPCLK to be output and QSSL to be asserted. However, SPE is forcibly cleared to 0 when the receive buffer is read as the condition for terminating read operation. At that time, QSSL and QSPCLK are stopped asynchronously.

[Termination Method 2]

After receiving the required length of data, execute write operation in quad-SPI mode for the last sequence and ensure the transmit buffer is empty. Since the transmit command is not started unless transmit data is written, unnecessary QSPCLK is not output and QSSL is not asserted.

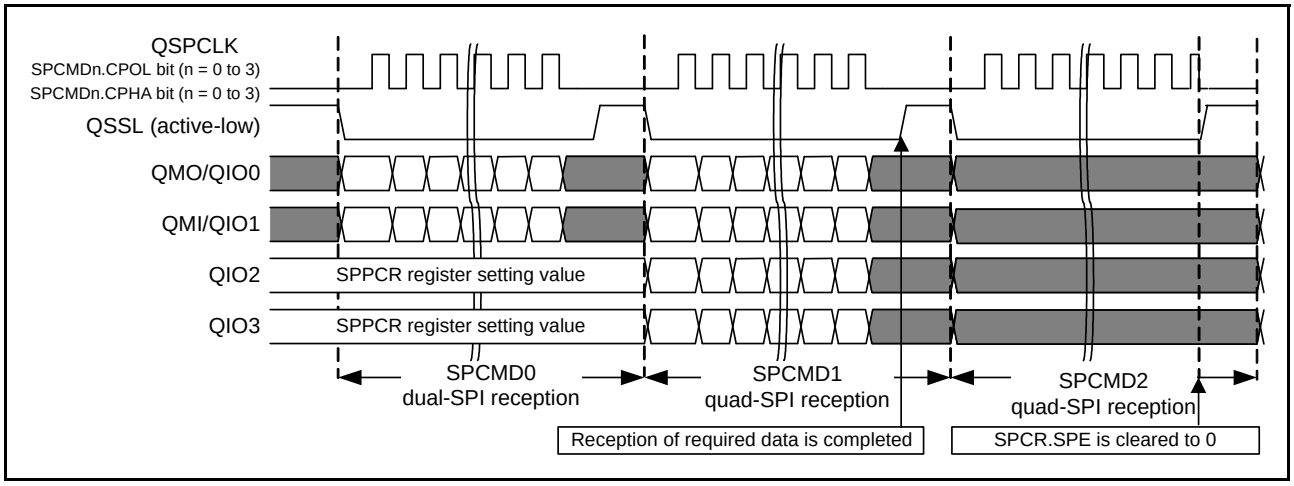


Figure 45.19 Example of Sequence Configuration with Transfer Mode Switching (Termination Method 1 When All the Commands Configuring a Sequence Are Dual-/Quad-SPI Read Operations)

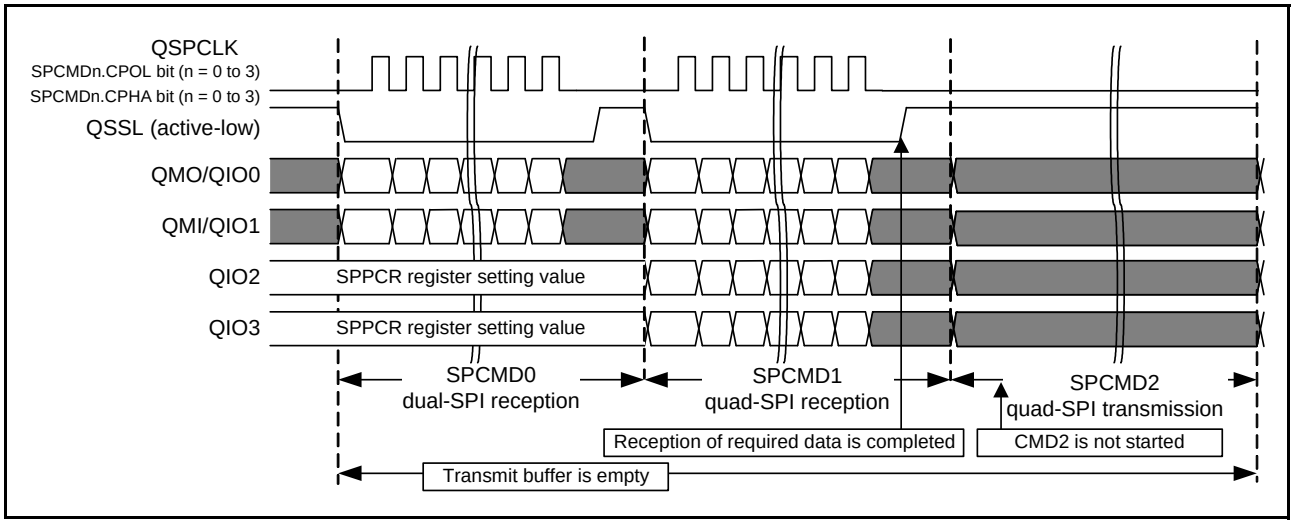


Figure 45.20 Example of Sequence Configuration with Transfer Mode Switching (Termination Method 2 When All the Commands Configuring a Sequence Are Dual-/Quad-SPI Read Operations)

(d) Burst Transfer

This module can execute burst transfer with the following two methods in dual/quad SPI modes.
One method uses the SPCMDn.SPB[3:0] bits (n = 0 to 3) and the SPBMULn register (n = 0 to 3). As with the single-SPI mode, setting the SPCMDn.SPB[3:0] bits to select 8, 16, or 32 bits and setting the SPBMULn register to select 1 through 4,294,967,296 allows the specified length of data to be continuously transferred for the specified times, where the length is specified by the SPCMDn.SPB[3:0] bits and the number of times is specified by the SPBMULn register. However, if the transmit buffer (SPTXB) becomes empty during transfer, or the receive buffer (SPRXB) has no longer a space enough to receive the specified length of data defined by the SPCMDn.SPB[3:0] bits, the clock is stopped until transfer is resumed. This method is effective to transfer a large amount of data in dual-/quad SPI modes. Figure 45.21 shows a burst transfer example in which the SPCMDn.SPB[3:0] bits are set to select 32 bits and the SPBMULn register to select four times thus specifying 128 bits as a total transfer data amount. The following describes operations (1) to (4) in the figure.

(1) First 32-bit data transfer
(2) Second 32-bit data transfer
(3) When the transmit buffer becomes empty or the receive buffer has no longer a space for 32 or more bits, the clock is stopped. Here, when QIO3 to QIO0 serve as output pins, QIO3 to QIO0 continue outputting the previous value. When QIO3 to QIO0 serve as input pins, the inputs to QIO3 to QIO0 depend on the output value of the device to communicate with. When data is written to the transmit buffer or an enough space is created in the receive buffer, the clock output is resumed to restart transfer.
(4) Third and fourth 32-bit data transfer

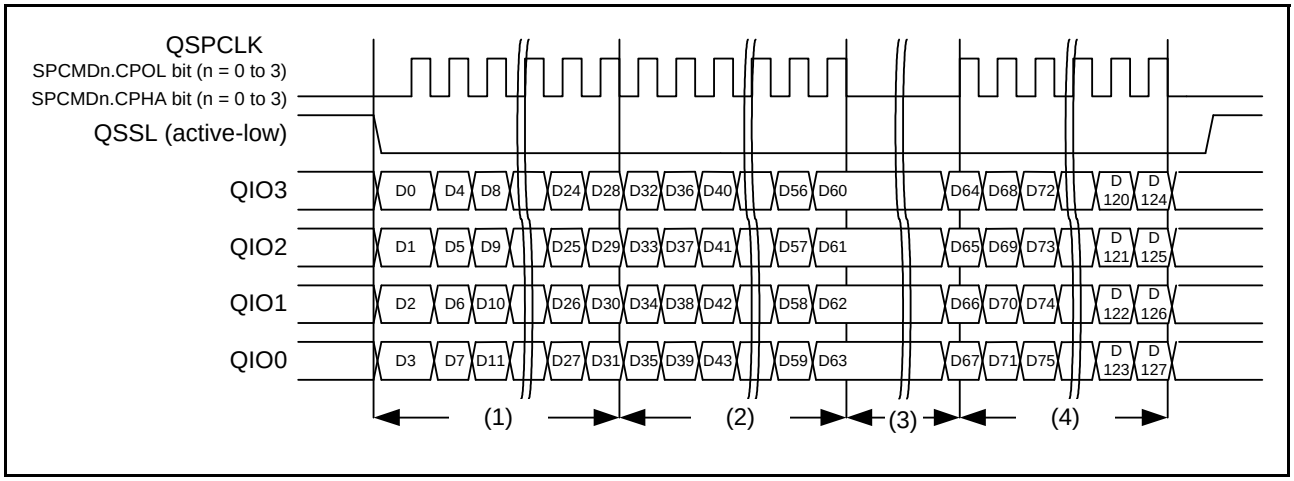


Figure 45.21 Burst Transfer Example in which Total Transfer Data Amount is 128 Bits (Quad-SPI Mode)

The other method uses the QSSL signal level keeping function as in single-SPI mode. Since this method allows switching the SPI transfer modes (single-/dual-/quad-SPI) during a transfer, it is particularly convenient when used with serial flash memory, where command data is written in single-SPI mode and data to be stored in memory is written in quad-SPI mode. Note, however, that at least one delay cycle should be inserted between transfers when switching the SPI transfer modes. Figure 45.22 shows a burst transfer example in which both single-SPI and quad-SPI modes are used. The following describes operations (1) to (6) in the figure.

(1) Clock delay period according to the SPCMD0 register setting. The setting should be made so that the delay period should be at least 1.5 QSPCLK cycles for the first transfer in the burst transfer.
(2) QSSL negation delay period according to the SPCMD0 register setting. Since the SPCMD0.SSLKP bit is set to 1, QSSL is not negated even after QSSL negation delay period is over. The QSSL negation delay period depends on the SPCMD0.SLNDEN bit setting. When the SPCMD0.SLNDEN bit is 1, the QSSL negation delay period is determined by the SSLND register setting, and the delay period is 0 QSPCLK cycle when the SPCMD0.SLNDEN bit is 0.
(3) Next-access delay period according to the SPCMD0 register setting. Since the SPCMD0.SSLKP bit is set to 1, QSSL is not negated even during next-access delay period. The next-access delay period depends on the SPCMD0.SPNDEN bit setting. When the SPCMD0.SPNDEN bit is 1, the next-access delay period is determined by the SPND register setting, and the delay period is 0 QSPCLK cycle when the SPCMD0.SPNDEN bit is 0. Up to this period, the data pin is driven according to the SPCMD0 register setting.
(4) Clock delay period according to the SPCMD1 register setting. The clock delay period depends on the the SPCMD1.SCKDEN bit setting. When the SPCMD1.SCKDEN bit is 1, the clock delay period is determined by the SPCKD register setting, and the delay period is 0 QSPCLK cycle when the SPCMD1.SCKDEN bit is 0.
(5) QSSL negation delay period according to the SPCMD1 register setting. The setting should be made so that the delay period should be at least one QSPCLK cycle for the last transfer in the burst transfer. Since the SPCMD1.SSLKP bit is set to 0, QSSL is negated after QSSL negation delay period is over.
(6) Next-access delay period according to the SPCMD1 register setting. The setting should be made so that the delay period should be at least one QSPCLK cycle for the last transfer in the burst transfer. Set the SPCMD1.SSLKP bit to 0 to negate QSSL.

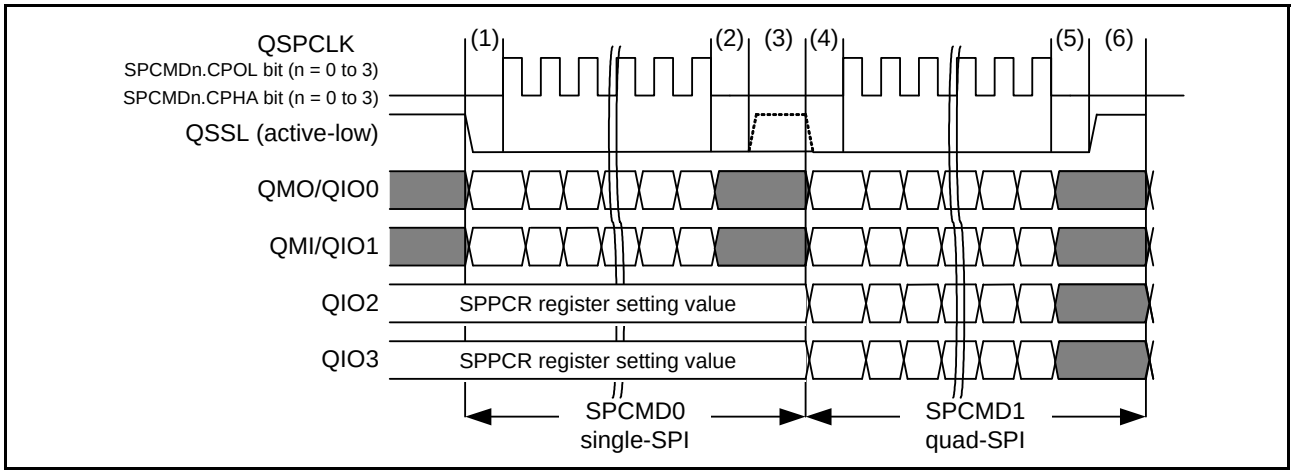


Figure 45.22 Burst Transfer Example in which QSSL Signal Level Keeping Function is Used (Single- and Quad-SPI Modes Used)

Note the following when specifying a burst transfer using this method.
Periods (2) to (4) should be inserted without fail when changing the clock polarity through command update.

(e) Initialization Flowchart

Figure 45.23 is a flowchart illustrating an example of initialization in SPI operation when this module is used in dual-/quad-SPI mode. For a description of how to set up the interrupt controller, DTC, or DMAC, see the descriptions given in the corresponding sections.

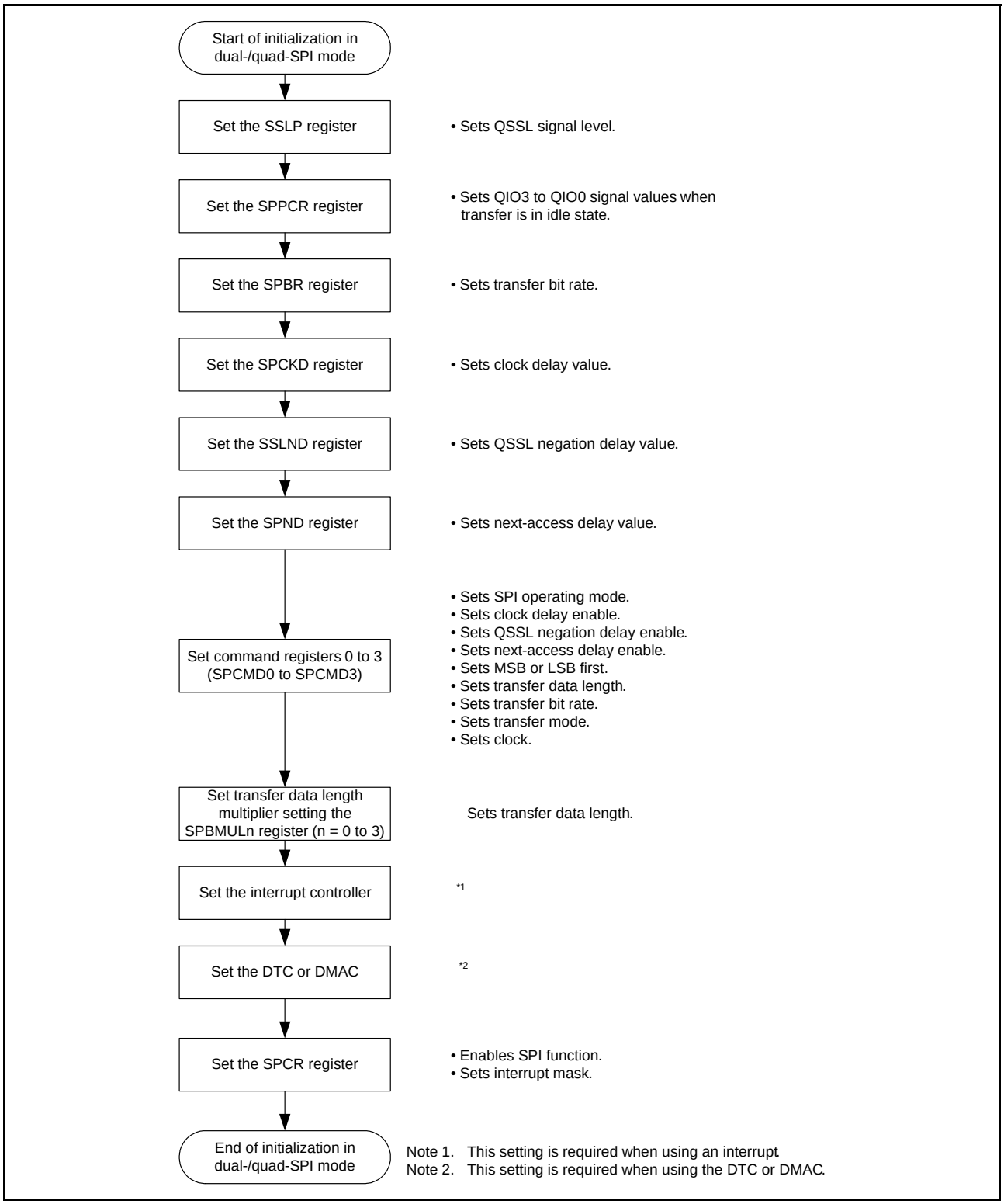


Figure 45.23 Example of Initialization Flowchart in Dual-/Quad-SPI Mode

(f) Transfer Operation Flowchart

Figure 45.24 shows an operation flowchart (the sequence is started by write operation) and Figure 45.25 shows an operation flowchart (the sequence is configured only with dual-/quad-read operation) in dual-/quad-SPI mode.

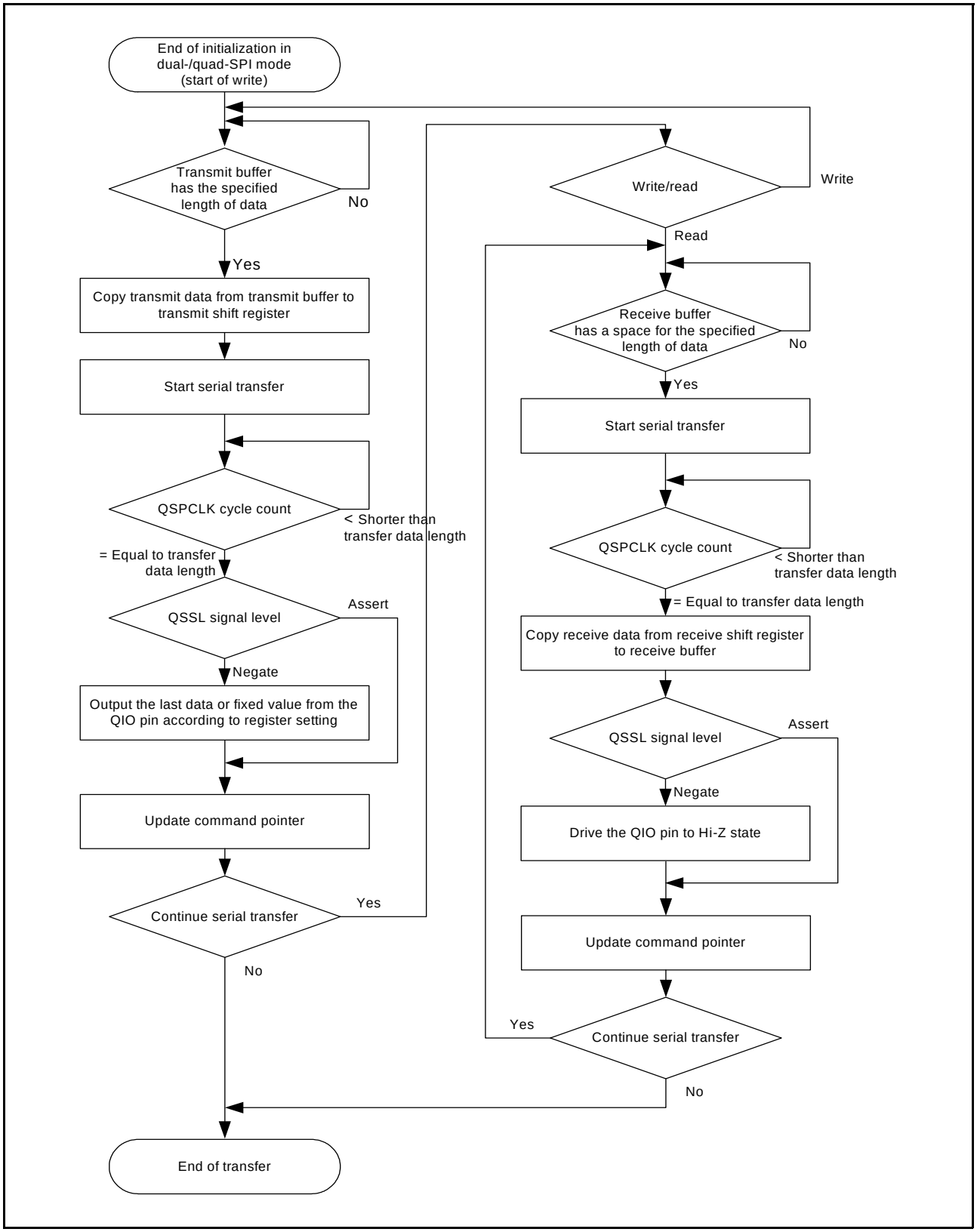


Figure 45.24 Transfer Operation Flowchart in Dual-/Quad-SPI Mode (Sequence is Started by Write Operation)

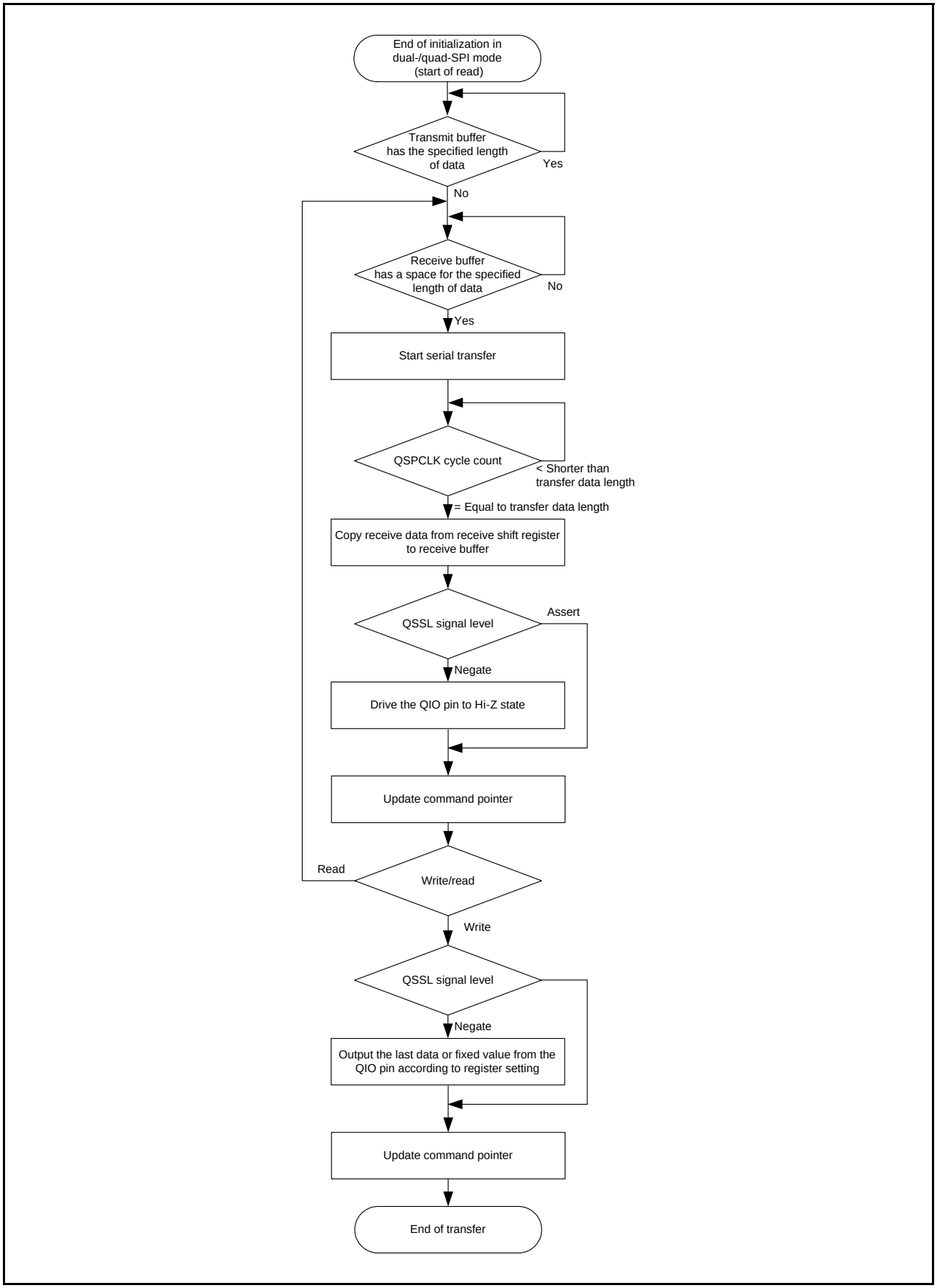


Figure 45.25 Transfer Operation Flowchart in Dual-/Quad-SPI Mode (Sequence is Configured only with Dual-/Quad Read Operations)

45.3.8 Interrupt Sources

This module has interrupt sources of receive buffer full, transmit buffer empty and QSSL negation. In addition, the DTC or DMAC can be activated by the receive buffer full or transmit buffer empty interrupt for data transfer.
Table 45.10 shows the interrupt sources. When any of the interrupt conditions in Table 45.10 is met, an interrupt is generated. The interrupt sources should be cleared with data transfer by the CPU, DTC, or DMAC.

Table 45.10 Interrupt Sources

Name	Interrupt Source	Interrupt Condition
SPRI	Receive buffer full	SPSR.SPRFF flag =1 and SPCR.SPRIE bit = 1
SPTI	Transmit buffer empty	SPSR.SPTEF flag = 1 and SPCR.SPTIE bit = 1
QSPSSLI	QSSL negation	SPSR.SPSSLF flag = 1 and SPCR.SPSSLIE bit = 1

If the conditions for a transmit buffer empty interrupt (SPTI) and receive buffer full interrupt (SPRI) are satisfied while the ICU.IRn.IR flag is 1, the interrupt request is not output to the ICU, but is retained internally. When the ICU.IRn.IR flag is cleared to 0, the retained interrupt request is output to the ICU.
For the ICU.IRn.IR flag, refer to section 15, Interrupt Controller (ICUA).
If the interrupt source (transmit buffer empty or receive buffer full) for the transfer buffer empty interrupt or receive buffer full interrupt is not cleared upon completion of transfer, the interrupt request generation conditions are as follows.

- DTC or DMAC transfer: When the interrupt condition is satisfied after the completion of access with the last DTC or DMAC transfer.
- CPU interrupt: When the interrupt condition is satisfied after the status flag has been cleared by register access.

Note that while the DTC or DMAC is in use, access to the flags by register reading or writing is prohibited.

45.3.9 Loopback Mode

This module provides loopback mode for testing. Writing 1 to the SPPCR.SPLP bit enables loopback mode. In loopback mode, this module connects the outputs from the transmit shift register to the inputs to the receive shift register instead.
Figure 45.26 shows a schematic internal connection in loopback mode.

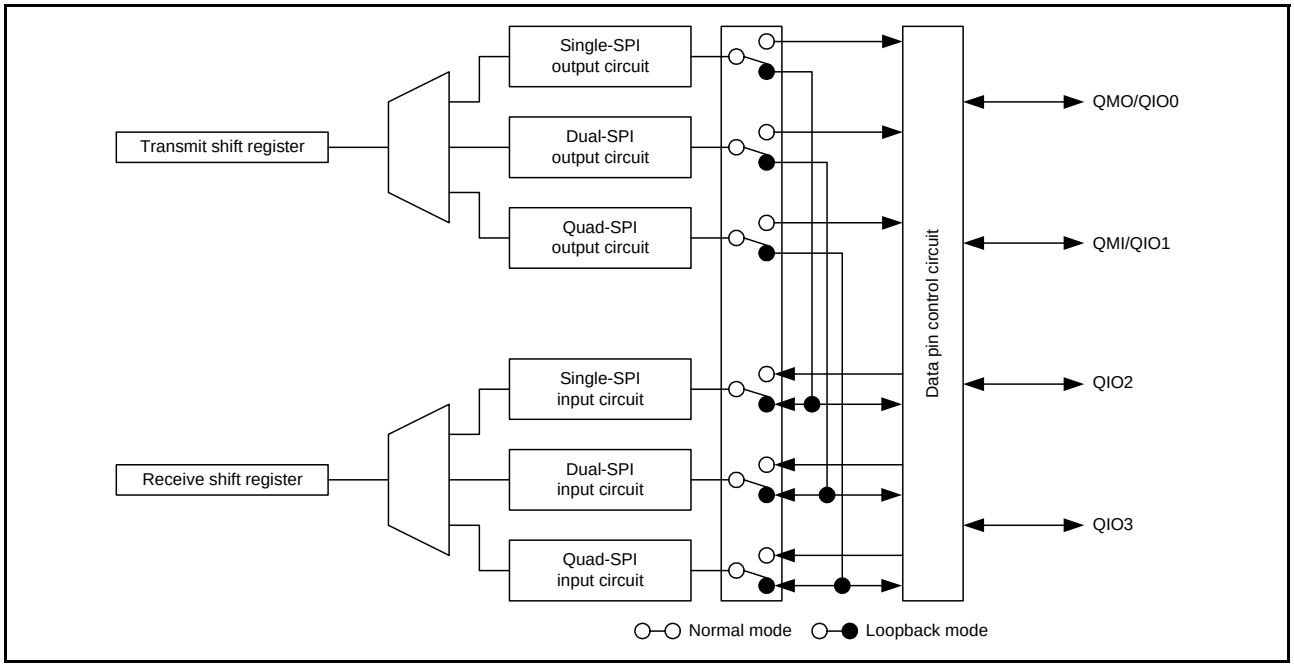


Figure 45.26 Schematic Internal Connection in Loopback Mode

45.4 Usage Notes

45.4.1 Notes on Starting Transfer

When the ICU.IRn.IR flag for the DTC or DMAC is 1 at the time transfer is to start, follow the procedure below to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1).

(1) Confirm that transfer has stopped (i.e. that the SPCR.SPE bit is 0).
(2) Clear the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) to 0.
(3) Read the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) and confirm that its value is 0.
(4) Clear the ICU.IRn.IR flag to 0.

45.4.2 Module Stop Function Setting

Operation of the QSPI can be disabled or enabled by module stop control register C (MSTPCRC). The initial setting is for operation of the QSPI to be stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

45.4.3 Notes on Using the Serial Flash Memory

When using the flash memory in dual- or quad-SPI operating mode, set the SPCMDn.CPOL and CPHA bits (n = 0 to 3) to 1 and select SPI mode 3. SPI mode 0 to 2 cannot be used. In addition, set the SPCMDn.SPNDEN, SLNDEN, and SCKDEN bits to 1 to secure delay period.

46. CRC Calculator (CRC)

The CRC (Cyclic Redundancy Check) calculator generates CRC codes.

46.1 Overview

Table 46.1 lists the specifications of the CRC calculator, and Figure 46.1 shows a block diagram of the CRC calculator.

Table 46.1 CRC Specifications

Item	Description
Data for CRC calculation*1	CRC codes are generated for any desired data in 8n-bit units (where n is a whole number)
CRC processor unit	8-bit parallel processing
CRC generating polynomial	One of three generating polynomials is selectable • 8-bit CRC $X^8 + X^2 + X + 1$ • 16-bit CRC $X^{16} + X^{15} + X^2 + 1$ $X^{16} + X^{12} + X^5 + 1$
CRC calculation switching	The bit order of CRC calculation results can be switched for LSB first or MSB first communication
Low power consumption function	Module stop state can be set.

Note 1. The circuit does not have a function to divide data for calculation into CRC calculation units. Write data in 8-bit units.

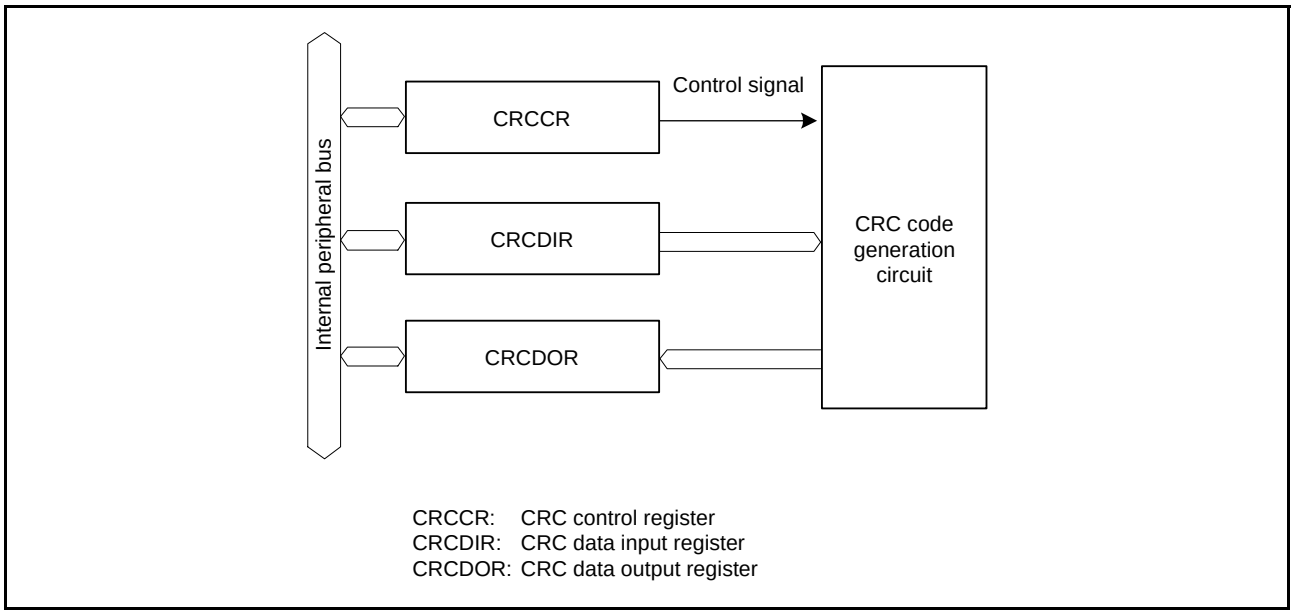


Figure 46.1 CRC Block Diagram

46.2 Register Descriptions

46.2.1 CRC Control Register (CRCCR)

Address(es): 0008 8280h

	b7	b6	b5	b4	b3	b2	b1	b0
	DORCLR	—	—	—	—	LMS	GPS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	GPS[1:0]	CRC Generating Polynomial Switching	b1 b0 0 0: No calculation is executed. 0 1: 8-bit CRC ($X^8 + X^2 + X + 1$) 1 0: 16-bit CRC ($X^{16} + X^{15} + X^2 + 1$) 1 1: 16-bit CRC ($X^{16} + X^{12} + X^5 + 1$)	R/W
b2	LMS	CRC Calculation Switching	0: Generates CRC for LSB first communication. 1: Generates CRC for MSB first communication.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DORCLR	CRCDOR Register Clear	1: Clears the CRCDOR register. This bit is read as 0.	R/W*1

Note 1. Only 1 can be written.

LMS Bit (CRC Calculation Switching)

This bit selects the bit order of generated 16-bit CRC code. Transmit the lower-order byte (b7 to b0) of the CRC code first for LSB first communication and the higher-order byte (b15 to b8) first for MSB first communication. For details on the transmission and reception of CRC codes, refer to section 46.3, Operation.

DORCLR Bit (CRCDOR Register Clear)

Write 1 to this bit so that the CRCDOR register is set to 0000h.
This bit is read as 0. Only 1 can be written.

46.2.2 CRC Data Input Register (CRCDIR)

Address(es): 0008 8281h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

CRCDIR is a readable and writable register. Write data for CRC calculation to this register.

46.2.3 CRC Data Output Register (CRCDOR)

Address(es): 0008 8282h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CRCDOR is a readable and writable register.
Since its initial value is 0000h, rewrite the CRCDOR register to perform calculation using a value other than the initial value.
Data written to the CRCDIR register is CRC calculated and the result is stored in the CRCDOR register. If the CRC code is calculated following the transferred data and the result is 0000h, there is no CRC error.
When an 8-bit CRC ($X^8 + X^2 + X + 1$ polynomial) is in use, the valid CRC code is obtained in the low-order byte (b7 to b0). The high-order byte (b15 to b8) is not updated.

46.3 Operation

The CRC calculator generates CRC codes for use in LSB first or MSB first transfer.
The following shows examples of generating the CRC code for input data (F0h) using the 16-bit CRC generating polynomial ($X^{16} + X^{12} + X^5 + 1$). In these examples, the value of the CRC data output register (CRCDOR) is cleared before CRC calculation.
When an 8-bit CRC (with the polynomial $X^8 + X^2 + X + 1$) is in use, the valid bits of the CRC code are obtained in the lower-order byte of the CRCDOR register.

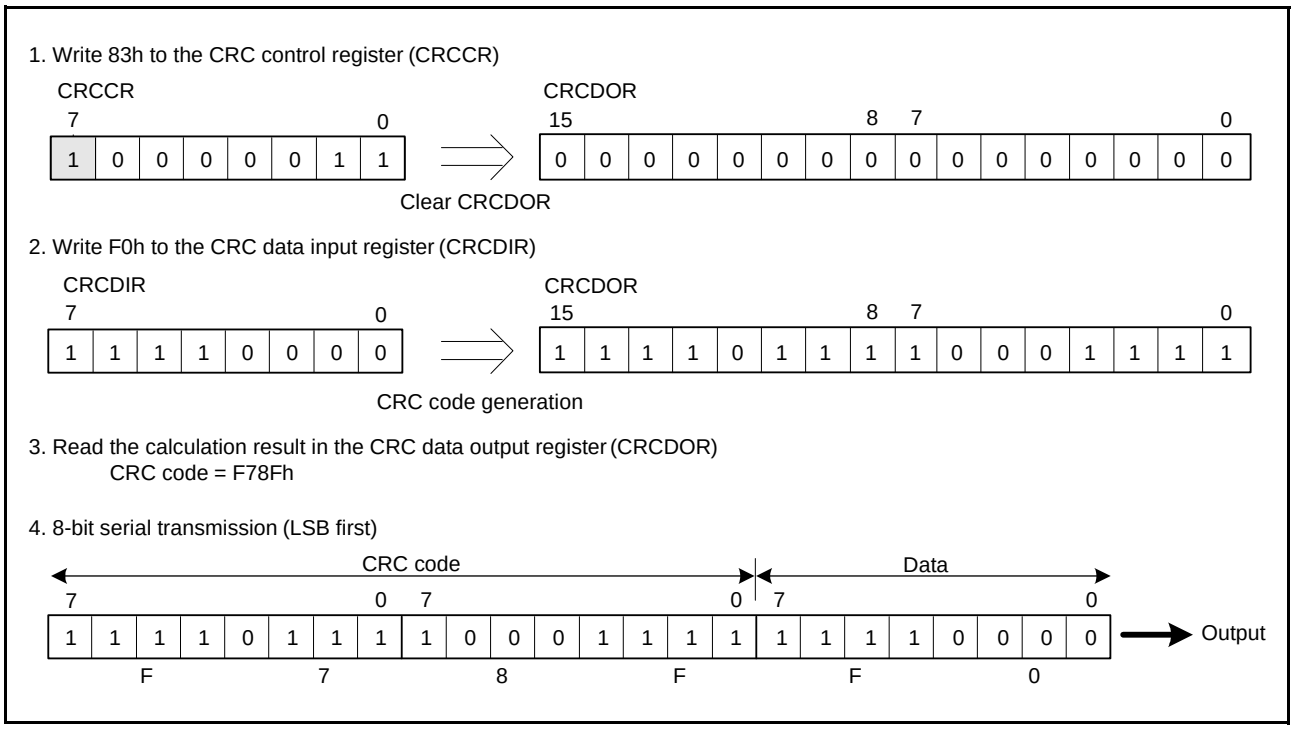


Figure 46.2 LSB First Data Transmission

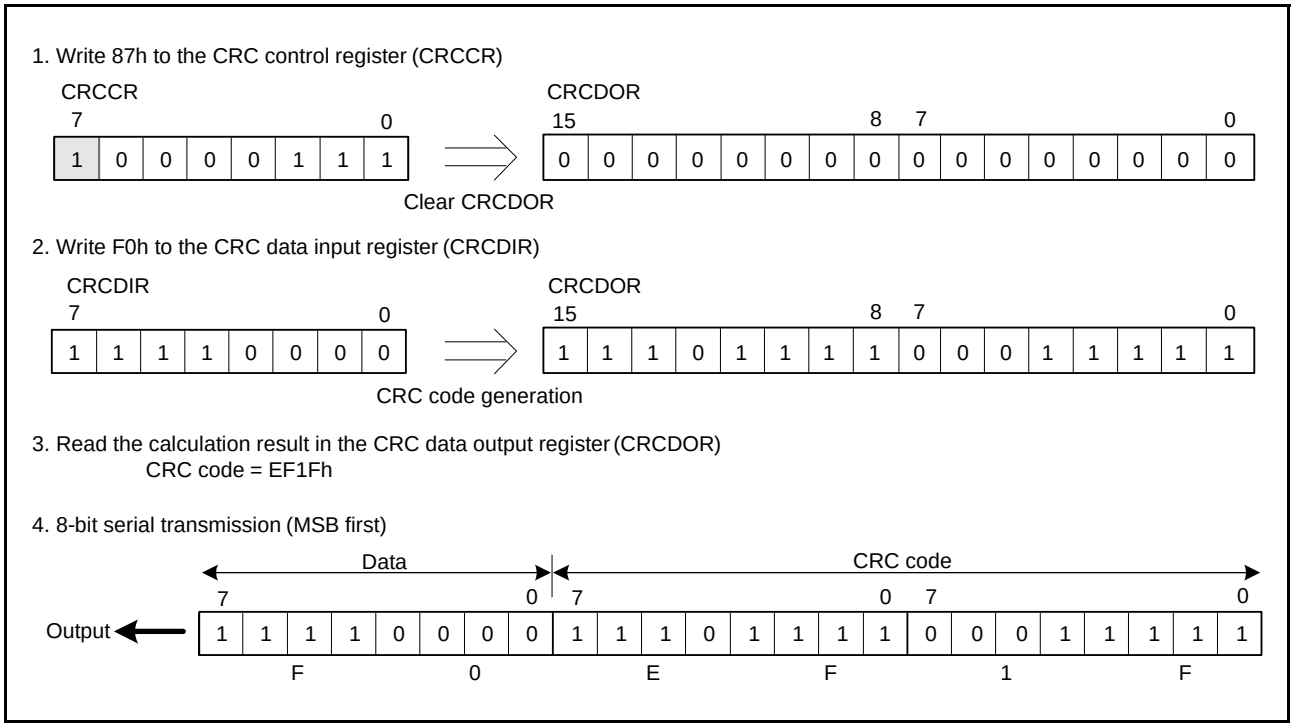


Figure 46.3 MSB First Data Transmission

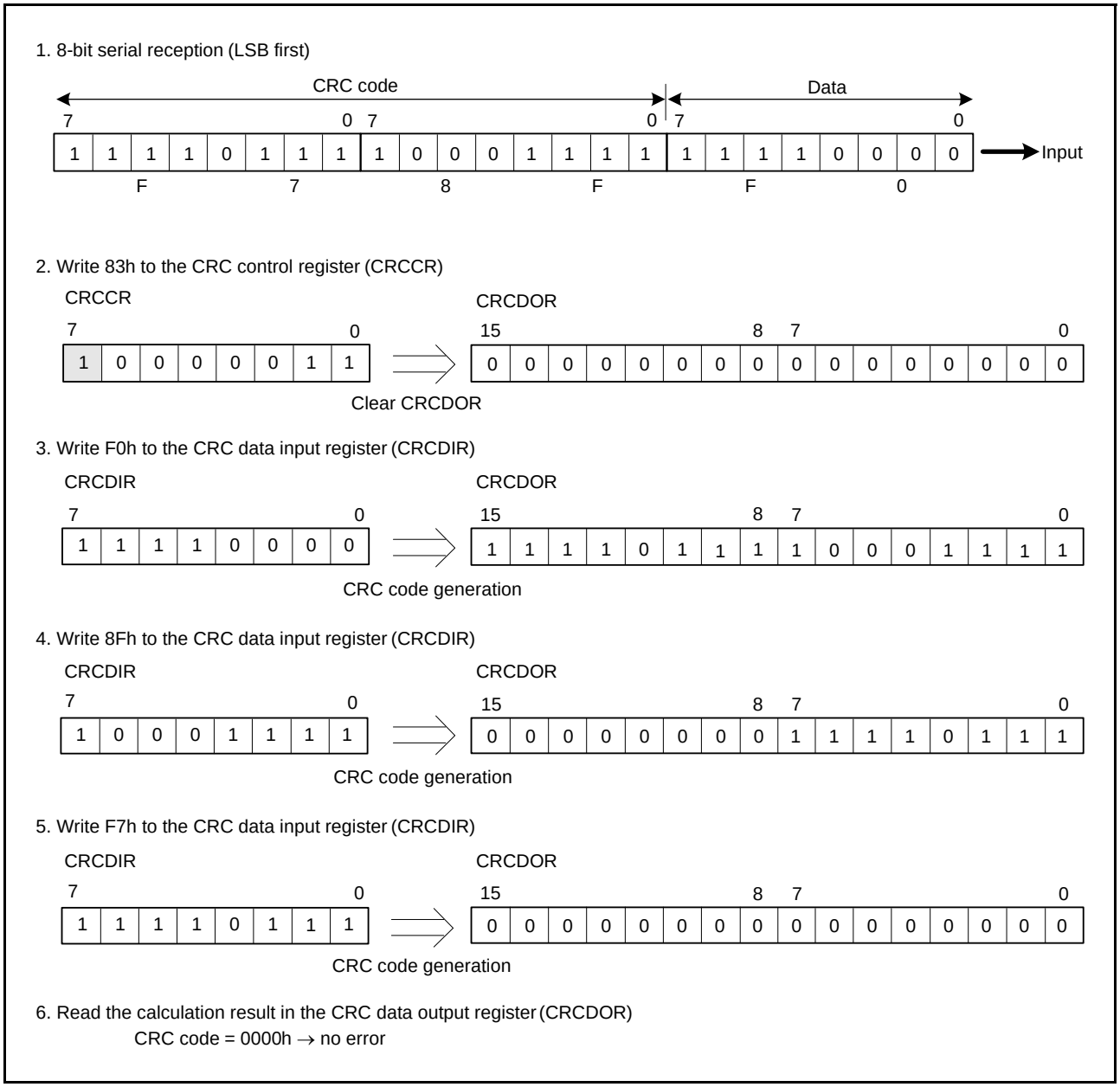
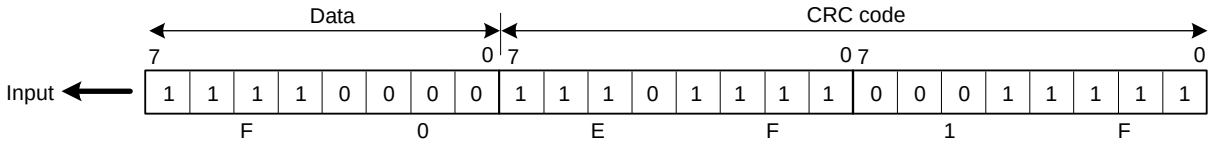
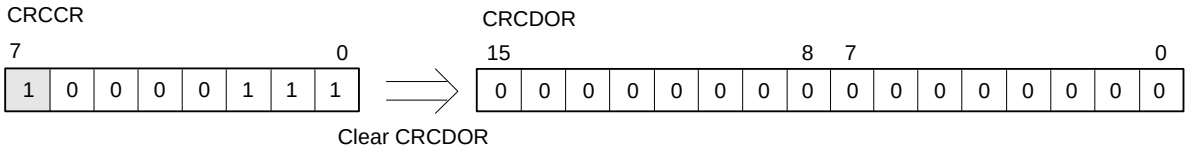


Figure 46.4 LSB First Data Reception

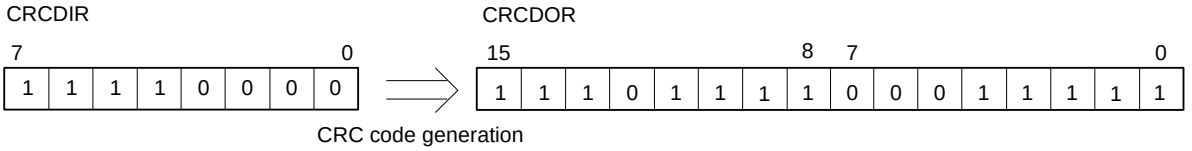
1. 8-bit serial reception (MSB first)



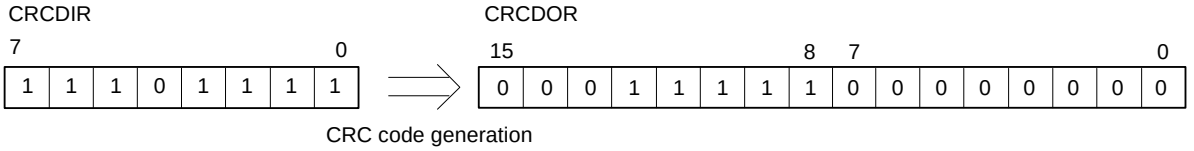
2. Write 87h to the CRC control register (CRCCR)



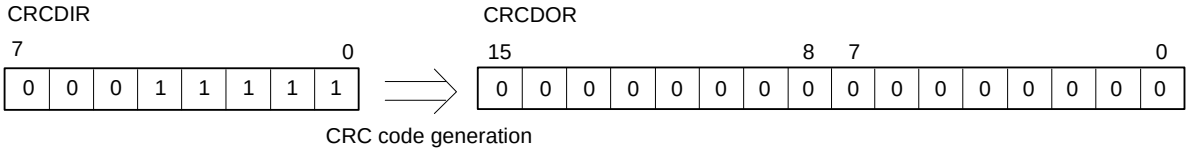
3. Write F0h to the CRC data input register (CRCDIR)



4. Write EFh to the CRC data input register (CRCDIR)



5. Write 1Fh to the CRC data input register (CRCDIR)



6. Read the calculation result in the CRC data output register (CRCDOR)

CRC code = 0000h → no error

Figure 46.5 MSB First Data Reception

46.4 Usage Notes

46.4.1 Module Stop Function Setting

Operation of the CRC calculator can be disabled or enabled using the module stop control register B (MSTPCRB). After a reset, the CRC is in the module stop state. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

46.4.2 Note on Transmission

Note that the sequence of transmission for the CRC code differs according to whether transmission is LSB first or MSB first.

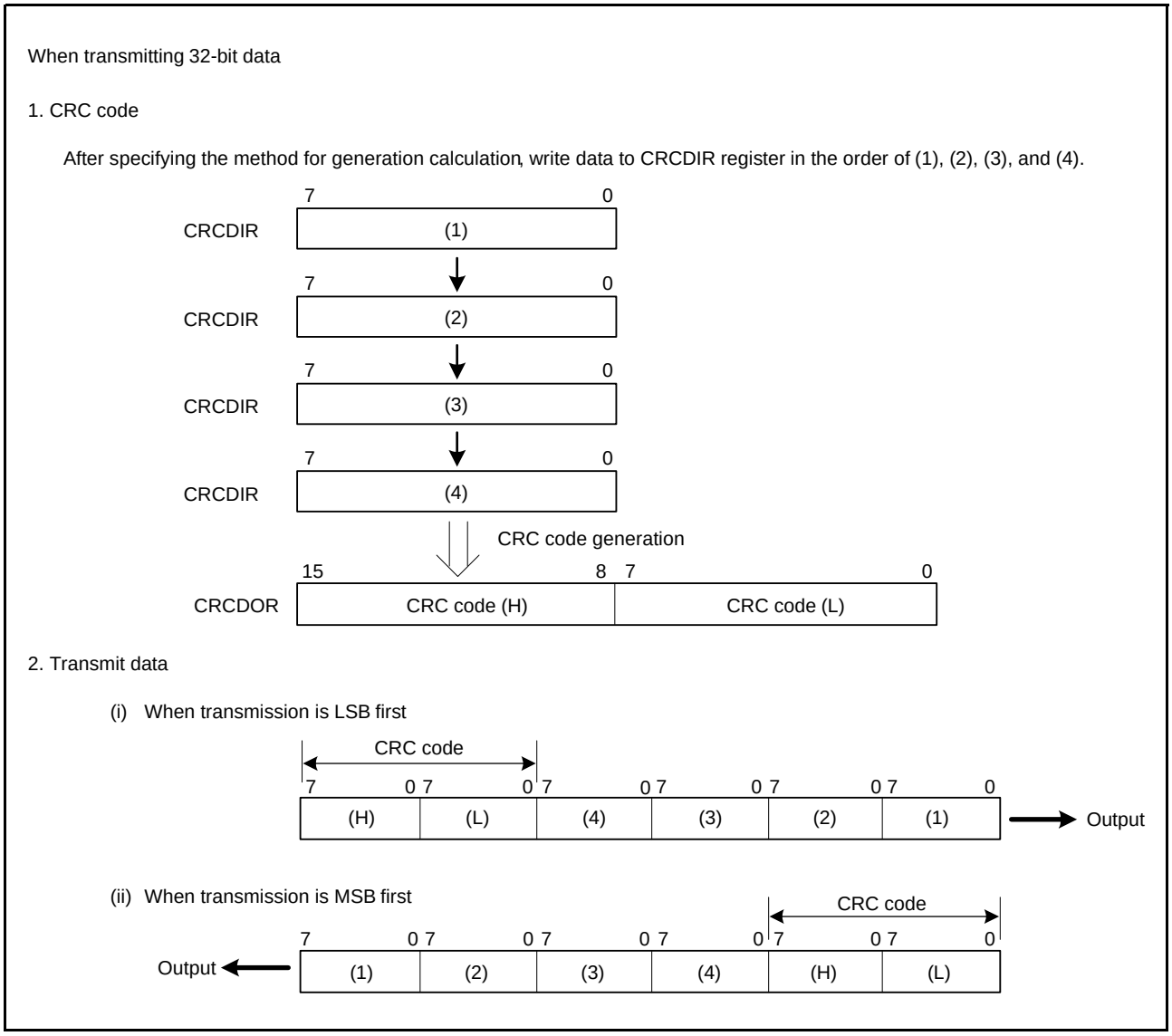


Figure 46.6 LSB First and MSB First Data Transmission

47. Serial Sound Interface (SSI)

This MCU integrates two channels of the serial sound interface (SSI) compliant with the I^{2}S bus specification. The SSI supports I^{2}S data format and MSB-first and left-justified/right-justified formats, so it can be used to send or receive audio data with various devices.

47.1 Overview

Table 47.1 SSI Specifications

Item	Specifications
Number of channels	Two channels (SSI0, SSI1)
Operating mode	Non-compressed mode
Transmission formats	• I^{2}S format supported • MSB-first supported Left-justified/right-justified formats selectable
Function	• Serves as both a transmitter and a receiver Channel 0 supports full-duplex communications. • Capable of various audio formats • SSISCKn (serial bit clock) can be selected from among 16, 32, 48, and 64 fs (fs: Sampling rate) (n = 0, 1) • The master clock (MCLK) is input from the master clock pin for audio (AUDIO_MCLK) (1 to 50 MHz) • Includes 8-stage FIFO buffers in transmitter and receiver • Capable of selecting whether to stop word select (SSIWSn) or not when data transmission is stopped
Interrupt sources	Three sources • Communication error Transmit underflow, transmit overflow, receive underflow, receive overflow, and idle • Receive data full • Transmit data empty
Low power consumption function	Module stop state can be set.

Figure 47.1 shows a block diagram of SSI (SSI0) and Figure 47.2 shows a block diagram of SSI (SSI1).

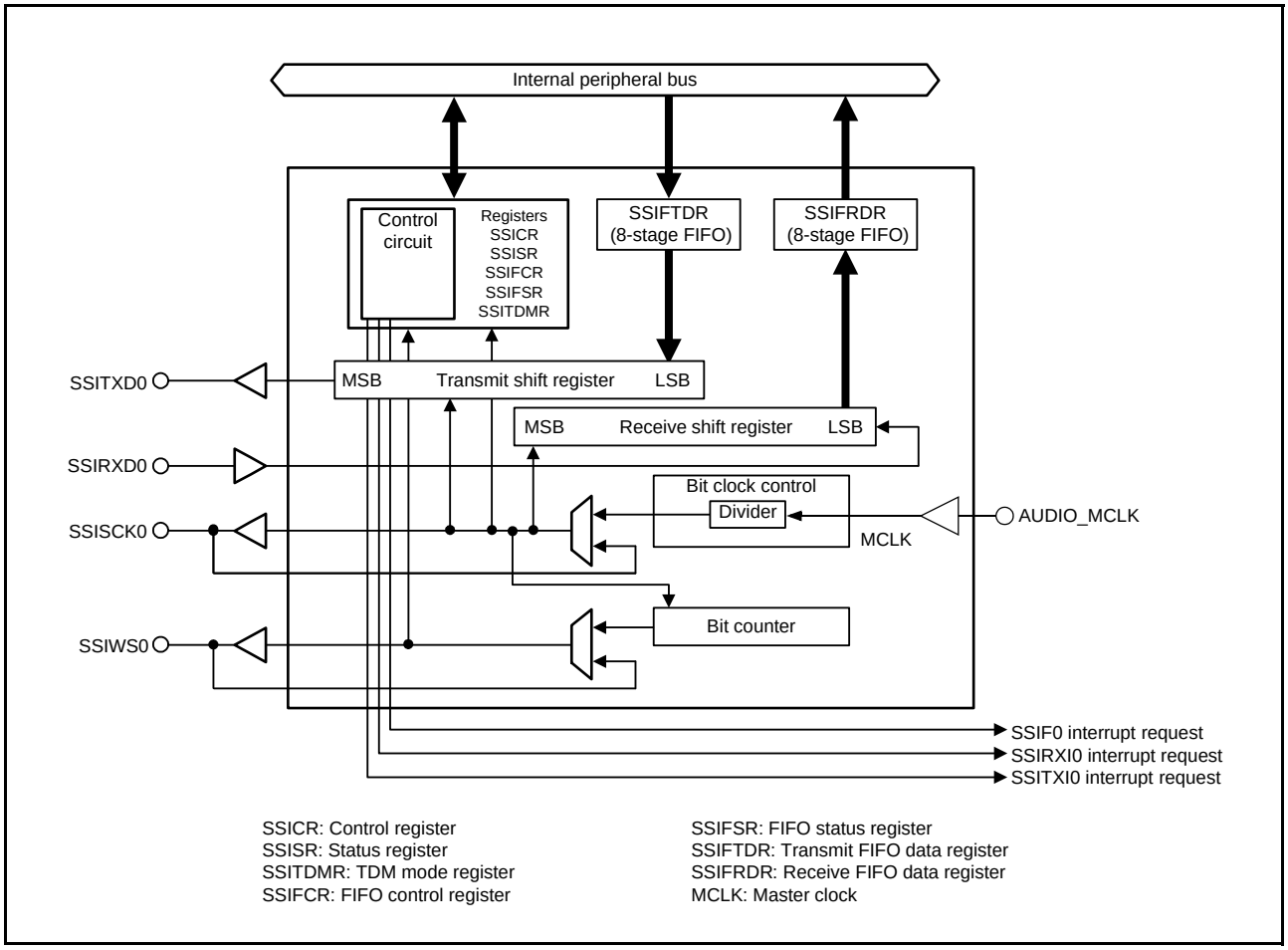


Figure 47.1 Block Diagram of SSI (SSI0)

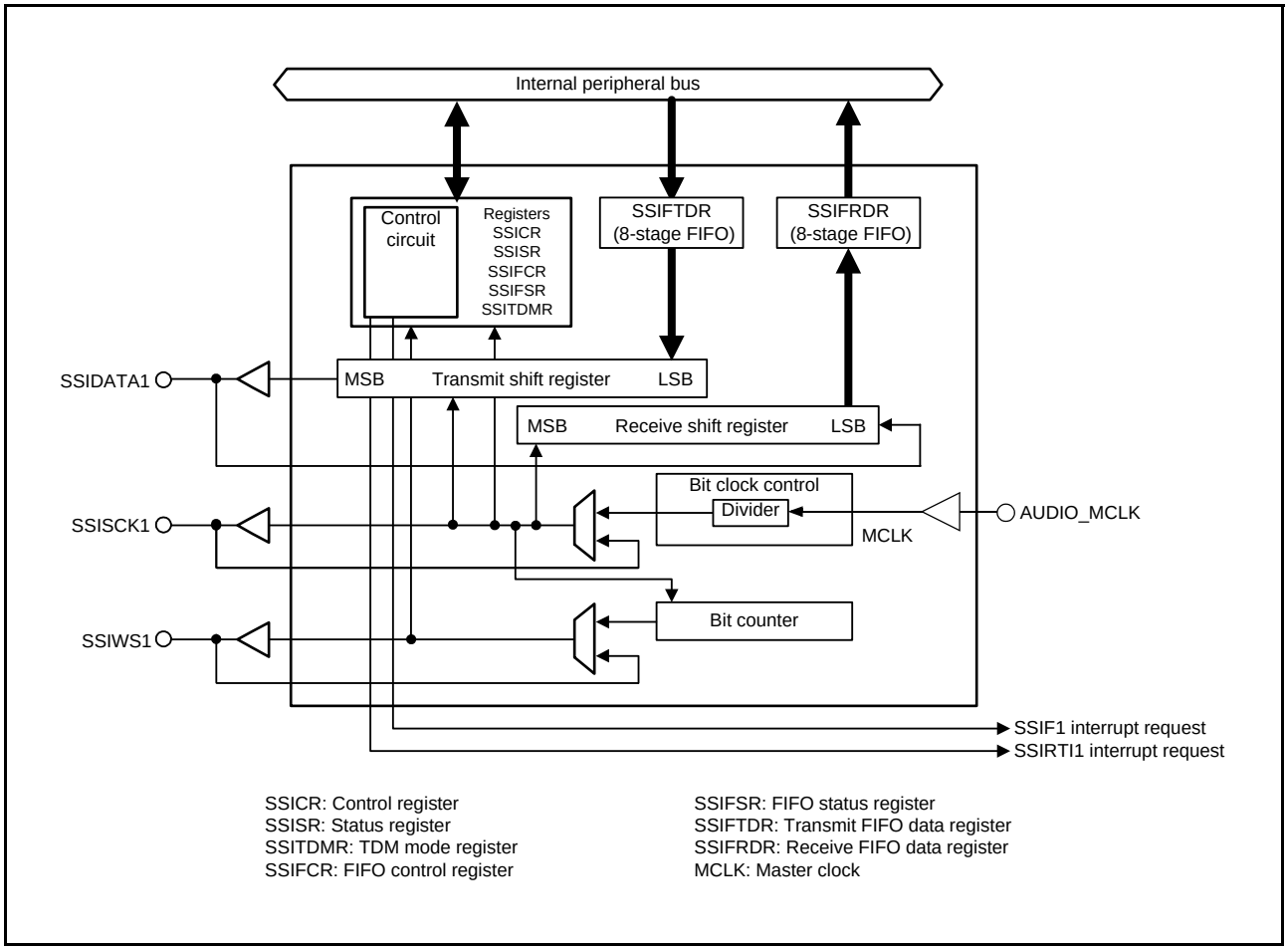


Figure 47.2 Block Diagram of SSI (SSI1)

Table 47.2 lists the I/O pins of the SSI.

Table 47.2 SSI I/O Pins

Channel	Pin Name	I/O	Description
SSI0	SSISCK0	I/O	Serial bit clock pin
	SSIWS0	I/O	Word selection pin
	SSITXD0	Output	Serial data output pin
	SSIRXD0	Input	Serial data input pin
SSI1	SSISCK1	I/O	Serial bit clock pin
	SSIWS1	I/O	Word selection pin
	SSIDATA1	I/O	Serial data input/output pin
SSI0, SSI1	AUDIO_MCLK	Input	Master clock for audio pin (input master clock)

47.2 Register Descriptions

47.2.1 Control Register (SSICR)

Address(es): SSI0.SSICR 0008 A500h, SSI1.SSICR 0008 A540h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	CHNL[1:0]		DWL[2:0]			SWL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	REN	Receive Enable	0: Disables receive operation. 1: Enables receive operation.	R/W
b1	TEN	Transmit Enable	0: Disables transmit operation. 1: Enables transmit operation.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	MUEN	Mute Enable*1	0: Not muted. 1: Muted.	R/W
b7 to b4	CKDV[3:0]	Serial Bit Clock Frequency Setting*3	b7 b6 b5 b4 0 0 0 0: MCLK 0 0 0 1: MCLK/2 0 0 1 0: MCLK/4 0 0 1 1: MCLK/8 0 1 0 0: MCLK/16 0 1 0 1: MCLK/32 0 1 1 0: MCLK/64 0 1 1 1: MCLK/128 1 0 0 0: MCLK/6 1 0 0 1: MCLK/12 1 0 1 0: MCLK/24 1 0 1 1: MCLK/48 1 1 0 0: MCLK/96 Settings other than above are prohibited.	R/W
b8	DEL	Serial Data Delay*3	0: I^2S format compatibility One clock cycle delay between SSIWSn and SSITXD0/ SSIRXD0/SSIDATA1 (n = 0, 1) 1: MSB-first left-justified/right-justified format compatibility No delay between SSIWSn and SSITXD0/SSIRXD0/ SSIDATA1	R/W
b9	PDTA	Parallel Data Allocation*3	When data word length is 8 or 16 bits: 0: The lower bits of parallel data (SSIFTDR, SSIFRDR) are transferred prior to the upper bits. 1: The upper bits of parallel data (SSIFTDR, SSIFRDR) are transferred prior to the lower bits. When data word length is 18, 20, 22, or 24 bits: 0: Parallel data (SSIFTDR, SSIFRDR) is left-justified. 1: Parallel data (SSIFTDR, SSIFRDR) is right-justified.	R/W
b10	SDTA	Serial Data Alignment*3	0: Transmitting and receiving in the order of serial data and padding bits 1: Transmitting and receiving in the order of padding bits and serial data	R/W
b11	SPDP	Serial Padding Polarity*3	0: Padding data is 0. 1: Padding data is 1.	R/W

Bit	Symbol	Bit Name	Description	R/W
b12	SWSP	Word Select Polarity	0: SSIWSn is low for the 1st system word, high for the 2nd system word. 1: SSIWSn is high for the 1st system word, low for the 2nd system word.	R/W
b13	SCKP	Serial Bit Clock Polarity*3	0: SSIWSn, SSITXD0, and SSIDATA1 change at the SSISCKn falling edge (SSIWSn, SSIRXD0, and SSIDATA1 are sampled at the SSISCKn rising edge). 1: SSIWSn, SSITXD0, and SSIDATA1 change at the SSISCKn rising edge (SSIWSn, SSITXD0, and SSIDATA1 are sampled at the SSISCKn falling edge).	R/W
b14	SWSD	Word Select Direction*2, *3	0: SSIWSn pin is input (slave mode). 1: SSIWSn pin is output (master mode).	R/W
b15	SCKD	Serial Bit Clock Direction*2, *3	0: SSISCKn pin is input (slave mode). 1: SSISCKn pin is output (master mode).	R/W
b18 to b16	SWL[2:0]	System Word Length*3	Set the system word length to the bit clock frequency/2 fs. b18 b16 0 0 0: 8 bits (serial bit clock frequency = 16 fs) 0 0 1: 6 bits (serial bit clock frequency = 32 fs) 0 1 0: 24 bits (serial bit clock frequency = 48 fs) 0 1 1: 32 bits (serial bit clock frequency = 64 fs) Settings other than above are prohibited.	R/W
b21 to b19	DWL[2:0]	Data Word Length*3	b21 b19 0 0 0: 8 bits 0 0 1: 16 bits 0 1 0: 18 bits 0 1 1: 20 bits 1 0 0: 22 bits 1 0 1: 24 bits Settings other than above are prohibited.	R/W
b23, b22	CHNL[1:0]	Channels*3	b23 b22 0 0: One channel Settings other than above are prohibited.	R/W
b24	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b25	IIEN	Idle Interrupt Enable	0: Disables an idle interrupt. 1: Enables an idle interrupt.	R/W
b26	ROIEN	Receive Overflow Interrupt Enable	0: Disables a receive overflow interrupt. 1: Enables a receive overflow interrupt.	R/W
b27	RUIEN	Receive Underflow Interrupt Enable	0: Disables a receive underflow interrupt. 1: Enables a receive underflow interrupt.	R/W
b28	TOIEN	Transmit Overflow Interrupt Enable	0: Disables a transmit overflow interrupt. 1: Enables a transmit overflow interrupt.	R/W
b29	TUIEN	Transmit Underflow Interrupt Enable	0: Disables a transmit underflow interrupt. 1: Enables a transmit underflow interrupt.	R/W
b30	CKS	Audio Clock Select*3	0: AUDIO_MCLK input 1: Setting prohibited	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. While this module is muted, low is transmitted regardless of the value of serial data, but data transmission is not stopped. Since the number of data in the transmit FIFO decreases, write dummy data to the SSIFTDR register to prevent the generation of a transmit underflow. When the MUEN bit is set to 1, the SSITXD0 and SSIDATA1 pins immediately become low without synchronizing SSIWSn pin.
Note 2. Set the SCKD and SWSD bits to the same value. Other settings are prohibited.
Note 3. Rewriting is allowed only in the idle state.

REN Bit (Receive Enable)

This bit enables or disables receive operation. Setting this bit to 1 starts receive operation.

TEN Bit (Transmit Enable)

This bit enables or disables transmit operation. Setting this bit to 1 starts transmit operation.
SSITXD0 pin of SSI0 is set as output while SSITXD0 is selected by the multi-function pin controller (MPC), regardless of the TEN bit setting. SSIDATA1 pin of SSI1 is set as output when the TEN bit is 1 and input when the TEN bit is 0 while SSIDATA1 is selected by MPC.

Table 47.3 SSITXD0, SSIRXD0, and SSIDATA1 Pin States

Register Settings			**SSI0**		**SSI1**
MPC setting	TEN	REN	SSITXD0	SSIRXD0	SSIDATA1
SSI function	0	0	Output	Input	Input
	0	1	Output	Input	Input
	1	0	Output	Input	Output
	1	1	Output	Input	—
Other than SSI function	x	x	Depends on the selected function	Depends on the selected function	Depends on the selected function

x: Don't care
—: Settings prohibited.

CKDV[3:0] Bits (Serial Bit Clock Frequency Setting)

These bits select the frequency of the serial bit clock in master mode. Since the input clock from the SSISCKn pin is used in slave mode, the setting of these bits is ignored (n = 0, 1). The serial bit clock is used as the operating clock of the shift register.

Calculation Example:
When fs (sampling rate) = the SSIWSn frequency = 96 kHz and the system word length = 32 bits
The bit clock frequency = 96 kHz × 32 bits × 2 = 6.144 MHz is necessary, so set CKDV[3:0] = 0001b (MCLK/2) when MCLK = 12.288 MHz.

PDTA Bit (Parallel Data Allocation)

The setting of this bit specifies the allocation of data to be stored in the SSIFRDR register in receive mode and the SSIFTDR register in transmit mode.
During receive operation, the SSI stores the data received from the serial bus in the SSIFRDR register according to the PDTA bit setting.
During transmit operation, the SSI stores the data stored in the SSIFTDR register in the transmit shift register, and transmits the data to the serial bus according to the PDTA bit setting.

(1) When PDTA bit is 0

DWL[2:0] Bits	SSIFTDR and SSIFRDR Registers
000b	31 24 23 16 15 8 7 0 4th word | 3rd word | 2nd word | 1st word
001b	31 16 15 0 2nd word | 1st word
010b	31 14 13 0 Valid | Invalid
011b	31 12 11 0 Valid | Invalid
100b	31 10 9 0 Valid | Invalid
101b	31 8 7 0 Valid | Invalid

(2) When PDTA bit is 1

DWL[2:0] Bits	SSIFTDR and SSIFRDR Registers
000b	31 24 23 16 15 8 7 0 1st word | 2nd word | 3rd word | 4th word
001b	31 16 15 0 1st word | 2nd word
010b	31 18 17 0 Invalid | Valid
011b	31 20 19 0 Invalid | Valid
100b	31 22 21 0 Invalid | Valid
101b	31 24 23 0 Invalid | Valid

SCKP Bit (Serial Bit Clock Polarity)

This bit is used to select the polarity of SSISCKn signal (n = 0, 1).
Table 47.4 lists the setting of SCKP bit and signal I/O timing.

Table 47.4 Setting of SCKP Bit and Signal Timing

	SCKP Bit = 0	SCKP Bit = 1
SSIRXD0 and SSIDATA1 input sampling timing for reception	SSISCKn rising edge	SSISCKn falling edge
SSITXD0 and SSIDATA1 output changing timing for transmission	SSISCKn falling edge	SSISCKn rising edge
SSIWSn input sampling timing in slave mode (SWSD bit = 0)	SSISCKn rising edge	SSISCKn falling edge
SSIWSn output changing timing in master mode (SWSD bit = 1)	SSISCKn falling edge	SSISCKn rising edge

CHNL[1:0] Bits (Channels)

These bits select the number of channels to be decoded in each system word. Set these bits to 00b in this module.

47.2.2 Status Register (SSISR)

Address(es): SSI0.SSISR 0008 A504h, SSI1.SSISR 0008 A544h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	TCHNO[1:0]		TSWNO	RCHNO[1:0]		RSWNO	IDST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	IDST	Idle Status Flag	0: SSI communication is in progress. 1: SSI communication is idle.	R
b1	RSWNO	Receive System Word Number Flag	Receive word number	R
b3, b2	RCHNO[1:0]	Receive Channel Number Flag	These bits are read as 00b.	R
b4	TSWNO	Transmit System Word Number Flag	Transmit word number	R
b6, b5	TCHNO[1:0]	Transmit Channel Number Flag	These bits are read as 00b.	R
b24 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25	IIRQ	Idle Interrupt Status Flag	0: Not in idle state 1: In idle state	R
b26	ROIRQ	Receive Overflow Interrupt Status Flag	0: No receive overflow has occurred. 1: A receive overflow has occurred.	R/(W) *1
b27	RUIRQ	Receive Underflow Interrupt Status Flag	0: No receive underflow has occurred. 1: A receive underflow has occurred.	R/(W) *1
b28	TOIRQ	Transmit Overflow Interrupt Status Flag	0: No transmit overflow has occurred. 1: A transmit overflow has occurred.	R/(W) *1
b29	TUIRQ	Transmit Underflow Interrupt Status Flag	0: No transmit underflow has occurred. 1: A transmit underflow has occurred.	R/(W) *1
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 0 after confirming the flag to be 1 clears the flag. To clear flags, write 0 only to the flags to be cleared; write 1 to the other flags. Do not write 0 to a status flag indicating 0.

IDST Flag (Idle Status Flag)

This status flag indicates that the SSI is in idle state where communication is stopped.

This flag is set to 0 when communication starts after the SSICR.TEN bit or SSICR.REN bit is set to 1. Also, this flag is set to 1 if both the TEN and REN bits are set to 0 and system word communication is completed.

If the external device stops inputting the serial bit clock before communication is completed, this flag is not set to 1.

RSWNO Flag (Receive System Word Number Flag)

The initial value of this flag is 1, and its value is inverted when the data is transferred from the receive shift register to the SSIFRDR register.

This flag is initialized to 1 when the SSICR.REN bit value is changed from 0 to 1.

When the data word length specified by the SSICR.DWL[2:0] bits is 18 bits or more, this flag indicates which system word the data in the SSIFRDR register represents.

TSWNO Flag (Transmit System Word Number Flag)

This status flag indicates the current word number.
The initial value of this is 1, and its value is inverted when the data is transferred from the SSIFTDR register to the transmit shift register.
This flag is initialized to 1 when the SSICR.TEN bit value is changed from 0 to 1.
When the data word length specified by the SSICR.DWL[2:0] bits is 18 bits or more, this flag indicates the system word that is in the data transferred from the SSIFTDR register to the transmit shift register.

IIRQ Flag (Idle Interrupt Status Flag)

This status flag indicates whether this module is in idle state.
This flag is set regardless of the value of the SSICR.IIEN bit to allow polling.
The interrupt can be masked by setting the SSICR.IIEN bit to 0, but the interrupt cannot be cleared by writing 0 to this flag.
If IIRQ flag = 1 and SSICR.IIEN bit = 1, an interrupt occurs.

ROIRQ Flag (Receive Overflow Interrupt Status Flag)

This status flag indicates that receive data was supplied at a higher rate than was required. If a receive overflow occurs, stop reception and start from the beginning of the flowchart again.
This flag is set to 1 regardless of the setting of the SSICR.ROIEN bit. This flag can be set to 0 by writing 0 after confirming it to be 1.
If ROIRQ flag = 1 and SSICR.ROIEN bit = 1, an interrupt occurs.
If ROIRQ flag = 1, the data was transferred from the transmit shift register to the SSIFRDR register while the receive FIFO is full (SSIFSR.RDC[3:0] flags = 8h). This may lead to the loss of data.

Note: When an overflow occurs, the current data in the data buffer of this module is overwritten by the next incoming data from the SSI interface.

RUIRQ Flag (Receive Underflow Interrupt Status Flag)

This status flag indicates that receive data was supplied at a lower rate than was required. If a receive underflow occurs, stop reception and start the flowchart again from the beginning.
This flag is set to 1 regardless of the setting of the SSICR.RUIEN bit. This flag can be set to 0 by writing 0 after confirming it to be 1.
If RUIRQ flag = 1 and SSICR.RUIEN bit = 1, an interrupt occurs.
If RUIRQ flag = 1, the SSIFRDR register was read while the receive FIFO is empty (SSIFSR.RDC[3:0] flags = 0h). This may cause invalid receive data to be stored.

TOIRQ Flag (Transmit Overflow Interrupt Status Flag)

This status flag indicates that transmit data was supplied at a higher rate than was required. If a transmit overflow occurs, stop transmission and start from the beginning of the flowchart again.
This flag is set to 1 regardless of the setting of the SSICR.TOIEN bit. This flag can be set to 0 by writing 0 after confirming it to be 1.
If TOIRQ flag = 1 and SSICR.TOIEN bit = 1, an interrupt occurs.
If TOIRQ flag = 1, the SSIFTDR register had data written to it while the transmit FIFO is full (SSIFSR.TDC[3:0] flags = 8h). This may lead to the loss of data.

TUIRQ Flag (Transmit Underflow Interrupt Status Flag)

This status flag indicates that transmit data was supplied at a lower rate than was required. If a transmit underflow occurs, stop transmission and start from the beginning of the flowchart again.
This flag is set to 1 regardless of the setting of the SSICR.TUIEN bit. This flag can be set to 0 by writing 0 after

confirming it to be 1.
If TUIRQ flag = 1 and SSICR.TUIEN bit = 1, an interrupt occurs.
If TUIRQ flag = 1, the SSIFTDR register did not have data written to it before it was required for transmission. This may lead to the same data being transmitted once more.

Note: When a transmit underflow occurs, the last data input to the SSIFTDR register is transmitted until this module is in the idle state after transmission is stopped.

47.2.3 FIFO Control Register (SSIFCR)

Address(es): SSI0.SSIFCR 0008 A510h, SSI1.SSIFCR 0008 A550h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TTRG[1:0]		RTRG[1:0]		TIE	RIE	TFRST	RFRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFRST	Receive FIFO Data Register Reset*4	0: Release the receive FIFO data reset. 1: Initiates the receive FIFO data reset.	R/W
b1	TFRST	Transmit FIFO Data Register Reset*4	0: Release the transmit FIFO data reset. 1: Initiates the transmit FIFO data reset.	R/W
b2	RIE	Receive Data Full Interrupt Enable	0: Receive data full interrupt (RXI) request is disabled. 1: Receive data full interrupt (RXI) request is enabled.*1	R/W
b3	TIE	Transmit Data Empty Interrupt Enable	0: Transmit data empty interrupt (TXI) request is disabled. 1: Transmit data empty interrupt (TXI) request is enabled.*2	R/W
b5, b4	RTRG[1:0]	Receive FIFO Threshold Setting *4	b5 b4 0 0: 1 0 1: 2 1 0: 4 1 1: 6	R/W
b7, b6	TTRG[1:0]	Transmit FIFO Threshold Setting *4	b7 b6 0 0: 7 (1)*3 0 1: 6 (2)*3 1 0: 4 (4)*3 1 1: 2 (6)*3	R/W
b15 to b8	—	Reserved	These bits are read as undefined. The write value should be 0.	R/W
b16	SSIRST	SSI Software Reset	0: Clears the SSI software reset. 1: Initiates the SSI software reset.	R/W
b30 to b17	—	Reserved	These bits are read as undefined. The write value should be 0.	R/W
b31	AUCKE	Master Clock Enable*4	0: The master clock is disabled. 1: The master clock is enabled.	R/W

Note 1. The RXI request can be cleared by setting the SSIFSR.RDF flag to 0 (see the description of the SSIFSR.RDF flag for details) or RIE bit to 0.
Note 2. The TXI request can be cleared by setting the SSIFSR.TDE flag to 0 (see the description of the SSIFSR.TDE flag for details) or TIE bit to 0.
Note 3. The values in parenthesis are the number of empty stages in SSIFTDR at which the SSIFSR.TDE flag is set.
Note 4. Rewriting is allowed only in the idle state.

The SSIFCR register resets the number of the data bytes stored in the SSIFTDR and SSIFRDR registers, and specifies transmit FIFO and receive FIFO threshold values.

RFRST Bit (Receive FIFO Data Register Reset)

This bit invalidates the data in the SSIFRDR register to reset the FIFO to an empty state.

TFRST Bit (Transmit FIFO Data Register Reset)

This bit invalidates the data in the SSIFTDR register to reset the FIFO to an empty state.

RIE Bit (Receive Data Full Interrupt Enable)

This bit enables or disables generation of receive data full interrupt (RXI) requests when the SSIFSR.RDF flag is set to 1 during reception.

TIE Bit (Transmit Data Empty Interrupt Enable)

This bit enables or disables generation of transmit data empty interrupt (TXI) requests when the SSIFSR.TDE flag is set to 1 during transmit operation.

RTRG[1:0] Bits (Receive FIFO Threshold Setting)

These bits specify the receive FIFO threshold value. When the number of received data bytes stored in the SSIFRDR register (receive FIFO) has become equal to or greater than the value specified by the RTRG[1:0] bits, the SSIFSR.RDF flag is set to 1 and reading the received data is requested. If the SSIFCR.RIE bit is 1 at this time, a receive data full interrupt (RXI) request is generated.

TTRG[1:0] Bits (Transmit FIFO Threshold Setting)

These bits specify the transmit FIFO threshold value. When the number of transmit data bytes stored in the SSIFTDR register (transmit FIFO) has become equal to or less than the value specified by the TTRG[1:0], the SSIFSR.TDE flag is set to 1 and writing the transmit data is requested. If the SSIFCR.TIE bit is 1 at this time, a transmit data empty interrupt (TXI) request is generated.

SSIRST Bit (SSI Software Reset)

Writing 1 to this bit initializes the SSI internal status, registers other than the SSIFCR register, and bits other than this bit in the SSIFCR register. Since this bit is not automatically cleared to 0, confirm that 1 is written to it before writing 0. Do not write 0 to this bit and 1 to other bits at the same time. After modifying this bit, confirm that its value is modified before proceeding to the next processing.

47.2.4 FIFO Status Register (SSIFSR)

Address(es): SSI0.SSIFSR 0008 A514h, SSI1.SSIFSR 0008 A554h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	TDC[3:0]				—	—	—	—	—	—	—	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	RDC[3:0]				—	—	—	—	—	—	—	RDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RDF	Receive Data Full Flag	0: Number of received data bytes in the SSIFRDR register is less than the specified receive FIFO threshold value. 1: Number of received data bytes in the SSIFRDR register is equal to or greater than the specified receive FIFO threshold value.	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	RDC[3:0]	Receive Data Indicate Flag	Indicate the number of data units stored in the SSIFRDR register.	R
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	TDE	Transmit Data Empty Flag	0: Number of data bytes for transmission in the SSIFTDR register is greater than the specified transmit FIFO threshold value. 1: Number of data bytes for transmission in the SSIFTDR register is equal to or less than the specified transmit FIFO threshold value.*2	R/(W) *1
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b24	TDC[3:0]	Transmit Data Indicate Flag	Indicate the number of data units stored in the SSIFTDR register.	R
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 0 after confirming the flag to be 1 clears the flag. To clear flags, write 0 only to the flags to be cleared; write 1 to the other flags. Do not write 0 to a status flag indicating 0.

Note 2. Since the SSIFTDR register is an 8-stage FIFO register, the amount of data that can be written to it while TDE flag = 1 is “8 - specified transmit FIFO threshold value” bytes at maximum. Writing more data will be ignored. The number of data bytes in the SSIFTDR register is indicated in the TDC[3:0] flags.

The SSIFSR register consists of status flags indicating the operating status of the SSIFTDR register and SSIFRDR register.

RDF Flag (Receive Data Full Flag)

This flag indicates that, when the received data is transferred to the SSIFRDR register, the number of data bytes in the SSIFRDR register has become equal to or greater than the receive FIFO threshold value, and thus reading the received data from the SSIFRDR register has been enabled.

[Setting condition]

- The number of receive data bytes that is equal to or greater than the value specified by the SSIFCR.RTRG[1:0] bits is stored in the SSIFRDR register.

[Clearing conditions]

- 0 is written to the RDF flag after the RDF flag is confirmed to be 1.
- Received data is read from the SSIFRDR register using DMA or DTC transfer (transfer of the last block in block transfer). Do not clear the RDF flag to 0 during DMA or DTC transfer.

Note: Since the SSIFRDR register is a 32-byte FIFO register, the maximum number of data bytes that can be read from it while the RDF flag is 1 is indicated in the RDC[3:0] flags. If reading data from the SSIFRDR register is continued after all the data is read, undefined values will be read.

RDC[3:0] Flags (Receive Data Indicate Flag)

These flags indicate the number of data bytes stored in the SSIFRDR register.
RDC[3:0] flags = 0h indicates no received data. RDC[3:0] flags = 8h indicates that 32 bytes of received data is stored in the SSIFRDR register.

TDE Flag (Transmit Data Empty Flag)

This flag indicates that, when data is transferred from the SSIFTDR register to the transmit shift register, the number of data bytes in the SSIFTDR register has become less than the transmit FIFO threshold value, and thus writing transmit data to the SSIFTDR register has been enabled.
[Setting condition]

- The number of the transmit data bytes written to the SSIFTDR register is equal to or less than the value specified by the SSIFCR.TTRG[1:0] bits.

[Clearing conditions]

- 0 is written to the TDE flag after the TDE flag is confirmed to be 1.
- Transmit data is written to the SSIFTDR register using DMA or DTC transfer (transfer of the last block in block transfer). Do not clear the TDE flag to 0 during DMA or DTC transfer.

Note: Since the SSIFTDR register is a 32-byte FIFO register, the maximum number of bytes that can be written to it while the TDE flag is 1 is 8 – TDC[3:0]. If writing data to the SSIFTDR register is continued after all the data is written, writing will be invalid and an overflow occurs.

TDC[3:0] Flags (Transmit Data Indicate Flag)

These flags indicate the number of data bytes stored in the SSIFTDR register.
TDC[3:0] flags = 0h indicates no data for transmission. TDC[3:0] flags = 8h indicates that 32 bytes of data for transmission is stored in the SSIFTDR register.

47.2.5 Transmit FIFO Data Register (SSIFTDR)

Address(es): SSI0.SSIFTDR 0008 A518h, SSI1.SSIFTDR 0008 A558h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

The SSIFTDR register is a write-only FIFO register consisting of eight stages of 32-bit registers for storing transmit data. Write transmit data to the SSIFTDR register in 64-bit (two stages of FIFO) units regardless of the data word length setting. If transmit data ends on a 32-bit boundary, write 00000000h for the other 32 bits, and stop transmission while 64-bit writing is completed. When the transmit shift register is empty, the SSI transfers the transmit data written to the SSIFTDR register to start serial transmission, which can be continued until the SSIFTDR register becomes empty. Note that when the SSIFTDR register is full of data (32 bytes), the next data cannot be written to it. If writing is attempted, it will be ignored and an overflow occurs.

47.2.6 Receive FIFO Data Register (SSIFRDR)

Address(es): SSI0.SSIFRDR 0008 A51Ch, SSI1.SSIFRDR 0008 A55Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

The SSIFRDR register is a read-only FIFO register consisting of eight stages of 32-bit registers for storing received data. Each time 4 bytes of serial data is received, the SSI stores the received serial data in the SSIFRDR register from the receive shift register according to the PDTA bit setting. Receive operation can be continued until a maximum 32 bytes of data have been stored to in the SSIFRDR register. The SSIFRDR register can be read but cannot be written to. Note that when the SSIFRDR register is read when it stores no received data, undefined values will be read and a receive underflow occurs.
After the SSIFRDR register becomes full of received data, the data received thereafter will be lost and a receive overflow occurs.

47.2.7 TDM Mode Register (SSITDMR)

Address(es): SSI0.SSITDMR 0008 A520h, SSI1.SSITDMR 0008 A560h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CONT	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	CONT	WS Continue Mode*1	0: Disables WS continue mode. 1: Enables WS continue mode.	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit can be set only in master mode (SSICR.SCKD bit = 1 and SSICR.SWSD bit = 1).

The SSITDMR register is a readable/writable 32-bit register that enables or disables WS continue mode.

47.3 Operation

47.3.1 Bus Format

This module can operate as a transmitter or a receiver and can be configured into many serial bus formats in either mode. The bus format can be selected from one of the six modes shown in Table 47.5.

Table 47.5 Bus Format

	TEN	REN	SCKD	SWSD	MUEN	IIEN	TOIEN	TUIEN	ROIEN	RUIEN	CONT	SWSP	DEL	PDTA	SDTA	SPDP	SCKP	SWL[2:0]	DWL[2:0]	CHNL[1:0]
Non-compression slave receiver	0	1	0	0	Control bits							Configuration bits								
Non-compression slave transmitter	1	0	0	0																
Non-compression slave transceiver	1	1	0	0																
Non-compression master receiver	0	1	1	1																
Non-compression master transmitter	1	0	1	1																
Non-compression master transceiver	1	1	1	1																

47.3.2 Non-Compressed Mode

This SSI supports non-compressed mode only. It supports the I^2S compatible format as well as MSB-first and left-justified/right-justified.

(1) Slave Receiver

This mode allows the module to receive serial data from another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of this module, operation is not guaranteed.

(2) Slave Transmitter

This mode allows the module to transmit serial data to another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of this module, operation is not guaranteed.

(3) Slave Transceiver

This mode allows serial data transmission and reception between this module and another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of this module, operation is not guaranteed.

(4) Master Receiver

This mode allows the module to receive serial data from another device. The clock and word select signals are internally derived from the master clock. The format of these signals is defined in the configuration fields of this module. If the incoming data does not follow the configured format, operation is not guaranteed.

(5) Master Transmitter

This mode allows the module to transmit serial data to another device. The clock and word select signals are internally derived from the master clock. The format of these signals is defined in the configuration fields of this module.

(6) Master Transceiver

This mode allows serial data transmission and reception between this module and another device. The clock and word select signals are internally derived from the master clock. The format of these signals is defined in the configuration fields of this module.

(7) Operating Settings Related to Word Length

All bits related to the SSICR register's word length are valid in non-compressed modes. There are many configurations this module supports, but some of the combinations are shown below for the I²S compatible format, MSB-first and left-justified format, and MSB-first and right-justified format.
In this section SSITXD0, SSIRXD0, and SSIDATA1 are referred to SSIDATA.

- I²S Compatible Format

Figure 47.3 and Figure 47.4 show the I²S compatible format both without and with padding.
Padding occurs when the data word length is smaller than the system word length.

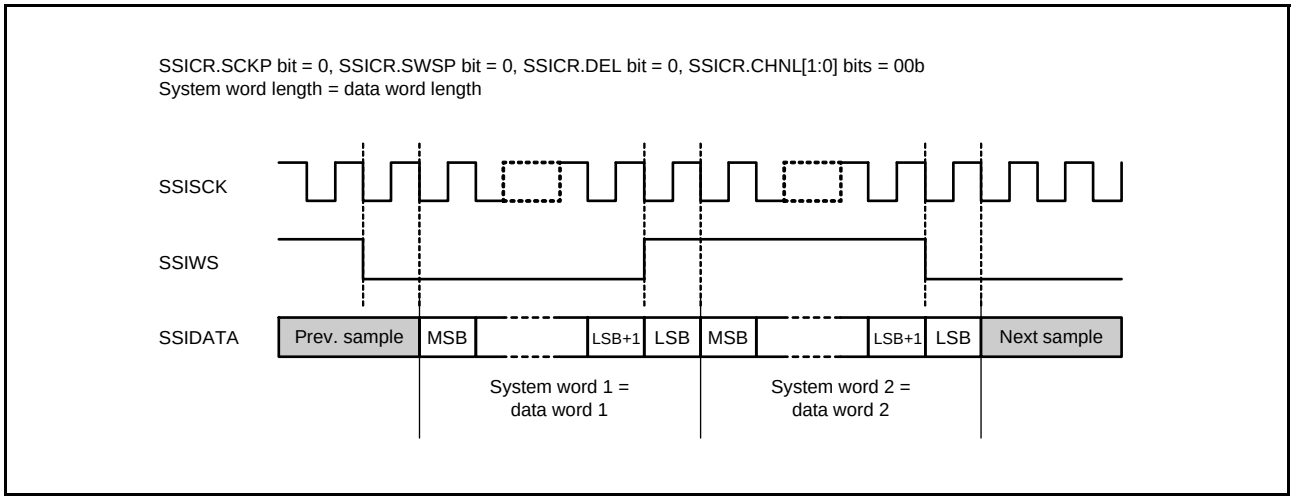


Figure 47.3 I²S Compatible Format (without Padding)

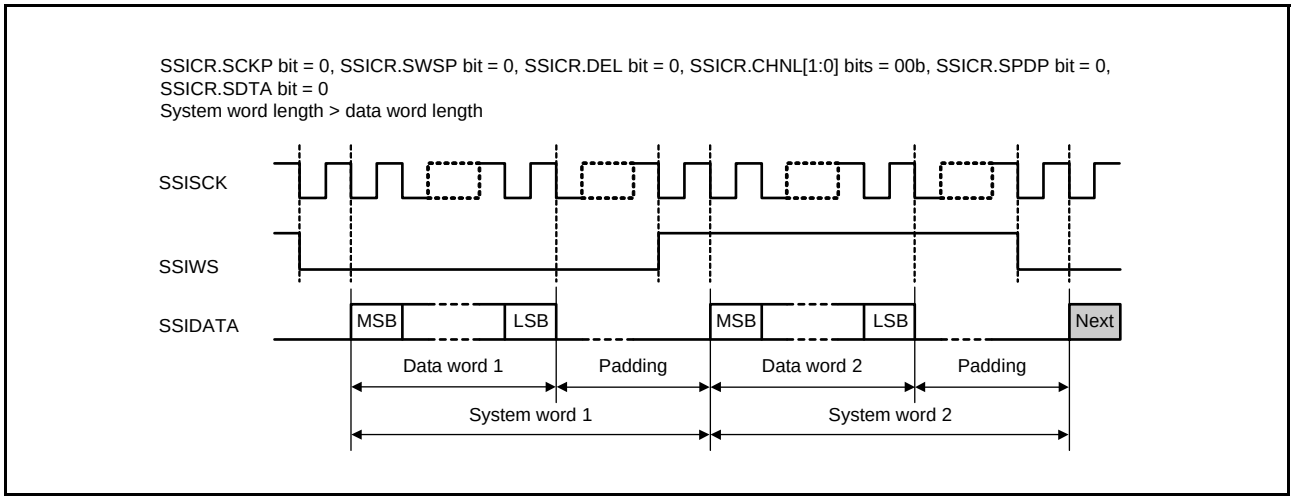


Figure 47.4 I²S Compatible Format (with Padding)

- MSB-First and Left-Justified Format

Figure 47.5 shows the MSB-first and left-justified format with padding.

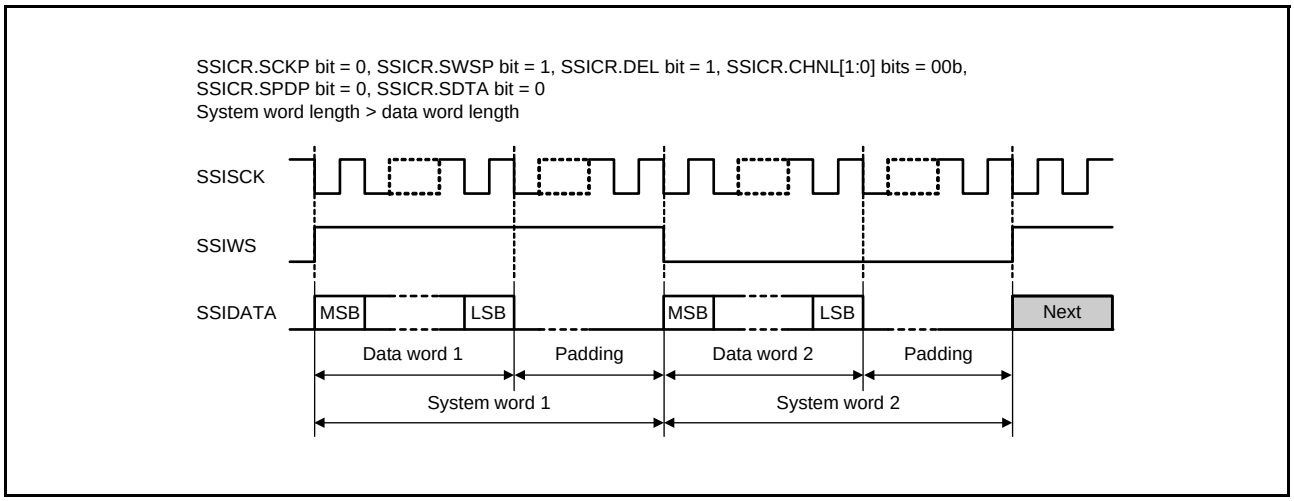


Figure 47.5 MSB-First and Left-Justified Format (Transmitted and Received in the Order of Serial Data and Padding Bits)

- MSB-First and Right-Justified Format

Figure 47.6 shows the MSB-first and right-justified format with padding.

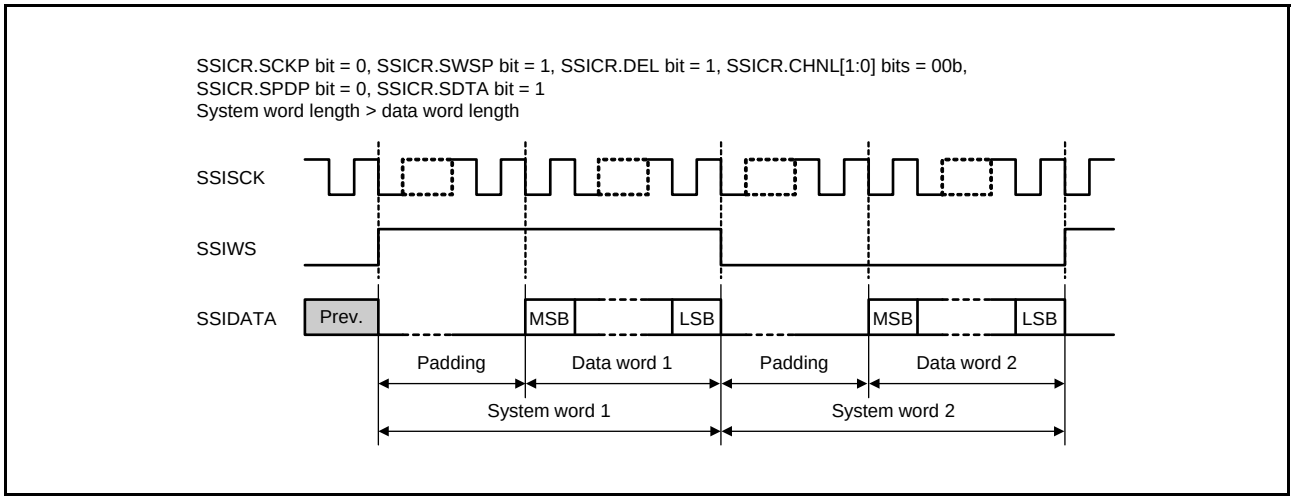


Figure 47.6 MSB-First and Right-Justified Format (Transmitted and Received in the Order of Padding Bits and Serial Data)

Table 47.6 shows the number of padding bits for each of the valid setting.

Table 47.6 Number of Padding Bits per System Word for Each Valid Setting

SSICR.CHNL[1:0] Bits (Decoded Channels per System Word)		SSICR.SWL[2:0] Bits (System Word Length)		SSICR.DWL[2:0] Bits (Data Word Length)					
				000b	001b	010b	011b	100b	101b
				8 bits	16 bits	18 bits	20 bits	22 bits	24 bits
00b	1 channel	000b	8 bits	0	—	—	—	—	—
		001b	16 bits	8	0	—	—	—	—
		010b	24 bits	16	8	6	4	2	0
		011b	32 bits	24	16	14	12	10	8

(8) Operating Settings Other than Word Length Related Settings

Several more configuration bits in non-compressed mode are shown below. These bits are not mutually exclusive, but some combinations may not be useful.

These configuration bits are described below with reference to the basic format sample in Figure 47.7.
In Figure 47.7 to Figure 47.15, a system word length of 6 bits and a data word length of 4 bits are used for simplification of these figures.

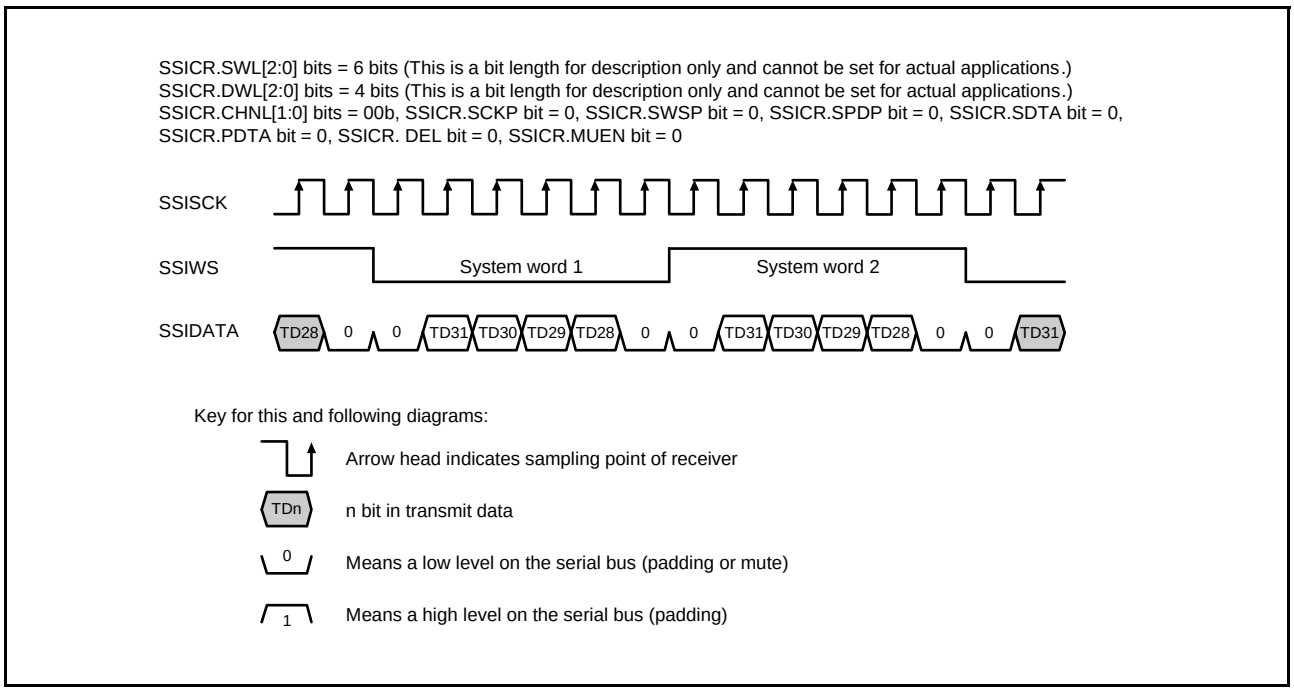


Figure 47.7 Basic Format Sample (Transmit Mode)

- Inverted Clock

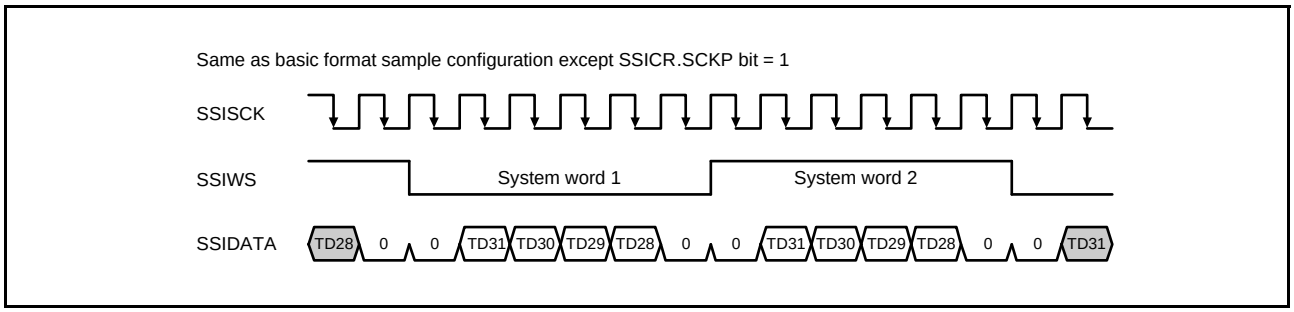


Figure 47.8 Inverted Clock

- Inverted Word Select

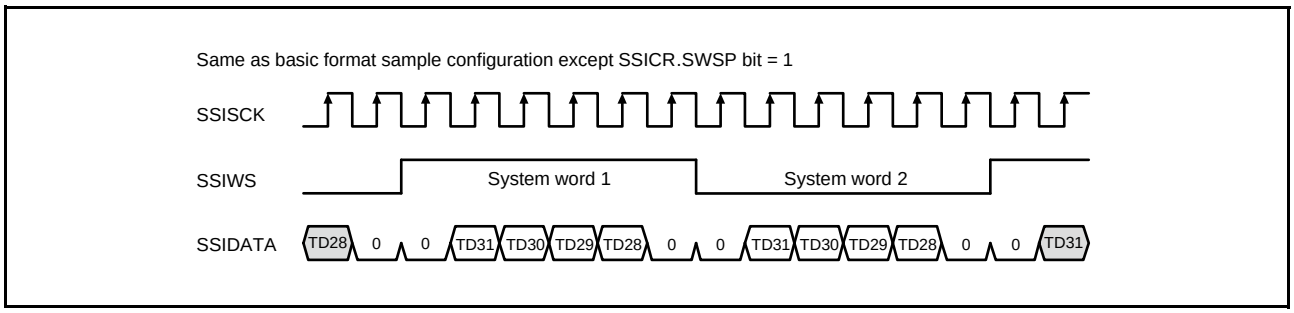


Figure 47.9 Inverted Word Select

- Inverted Padding Polarity

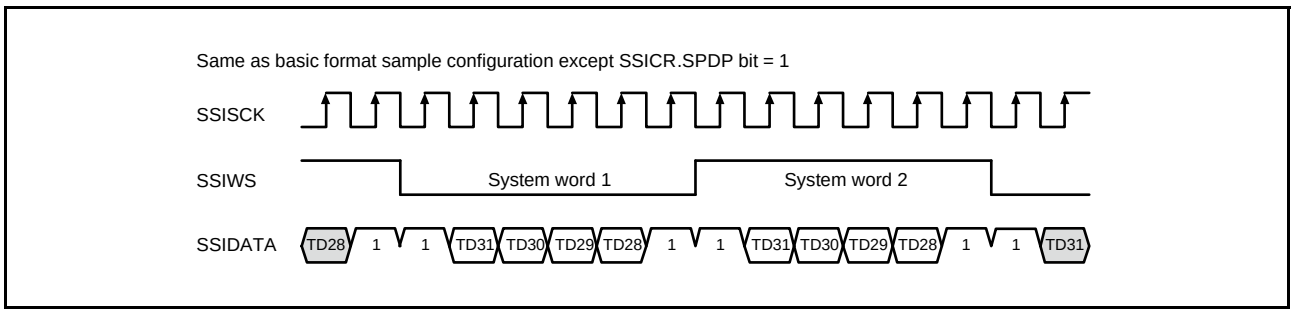


Figure 47.10 Inverted Padding Polarity

- Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

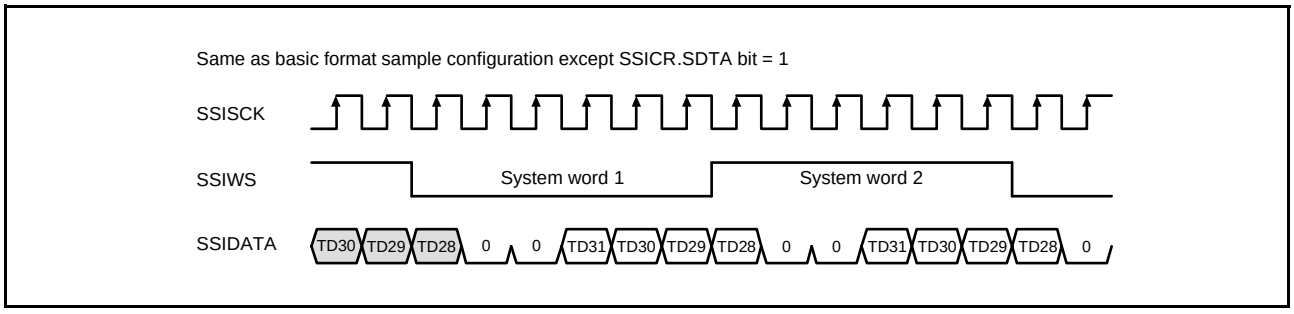


Figure 47.11 Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

- Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

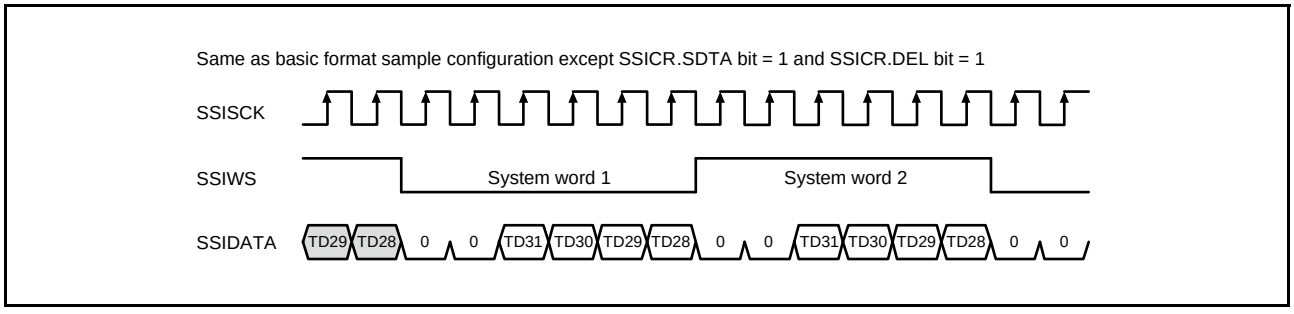


Figure 47.12 Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

- Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

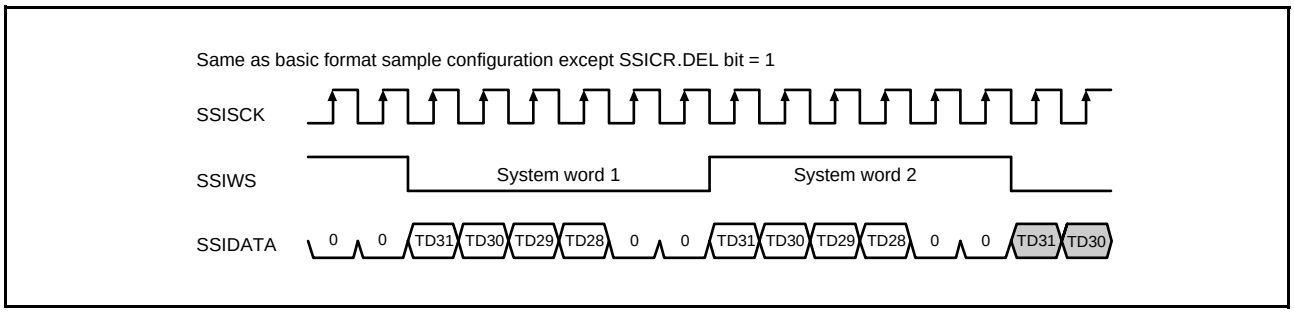


Figure 47.13 Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

- Parallel Right-Justified with Delay

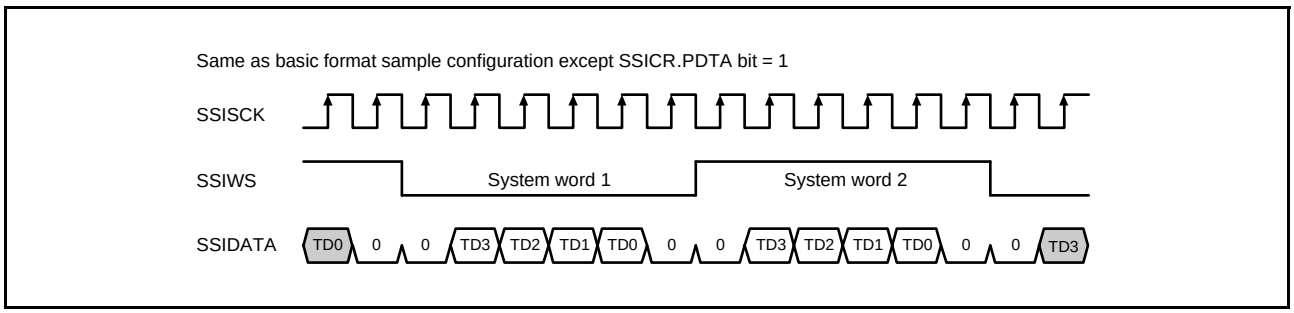


Figure 47.14 Parallel Right-Justified with Delay

- Mute Enabled

When the SSICR.MUEN bit is set to 1, the SSITXD0 and SSIDATA1 pins become low (0) without synchronizing SSIWSn (n = 0, 1).

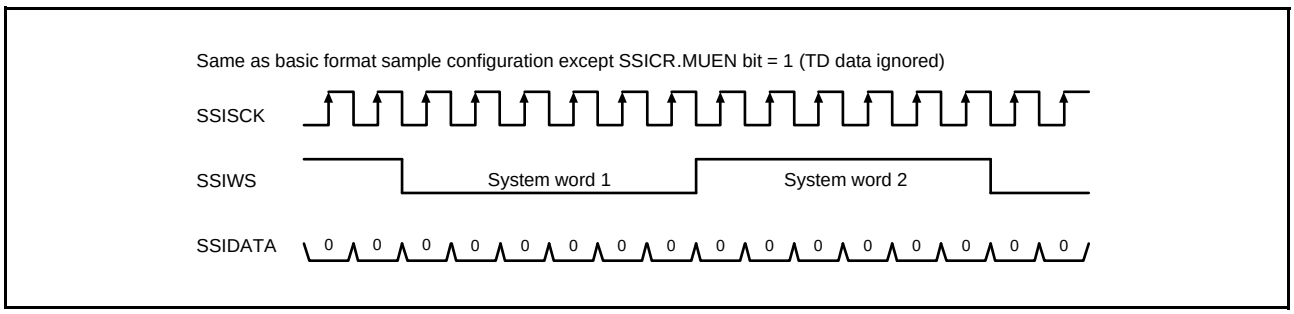


Figure 47.15 Mute Enabled

47.3.3 WS Continue Mode

In WS continue mode, the SSIWSn signal continues to be toggled irrespective whether data transmission is enabled or disabled (n = 0, 1). This mode can be set using the SSITDMR.CONT bit. With this mode enabled, the SSIWSn signal does not stop but continues toggling even if the SSICR.TEN and REN bits are both set to 0 (transmission disabled). With this mode disabled, the SSIWSn signal stops if the SSICR.TEN and REN bits are both set to 0.

Figure 47.16 and Figure 47.17 show the operations with WS continue mode enabled and disabled, respectively.

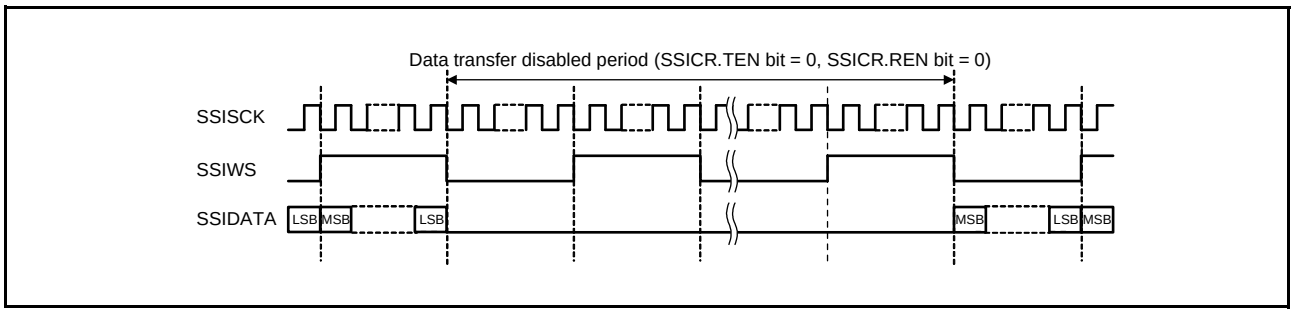


Figure 47.16 WS Continue Mode Enabled

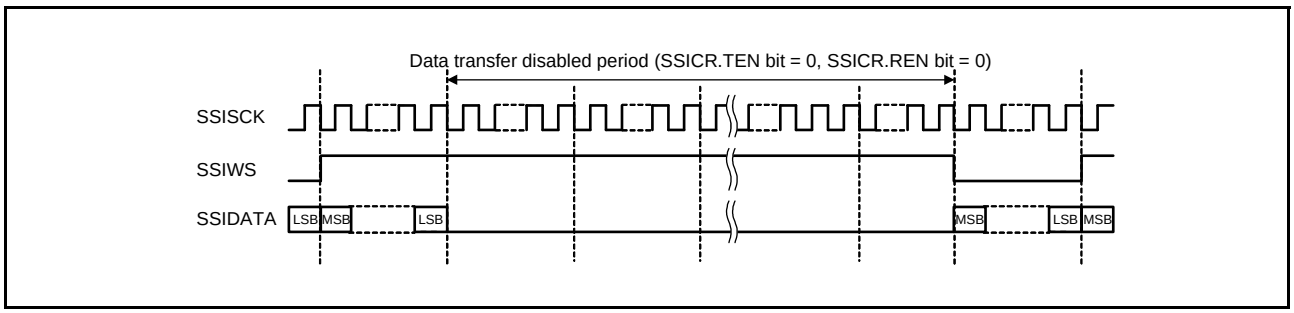


Figure 47.17 WS Continue Mode Disabled

47.3.4 Operating States

There are three states of operation: idle, communication, and waiting for idle. Figure 47.18 shows the operating state transitions.

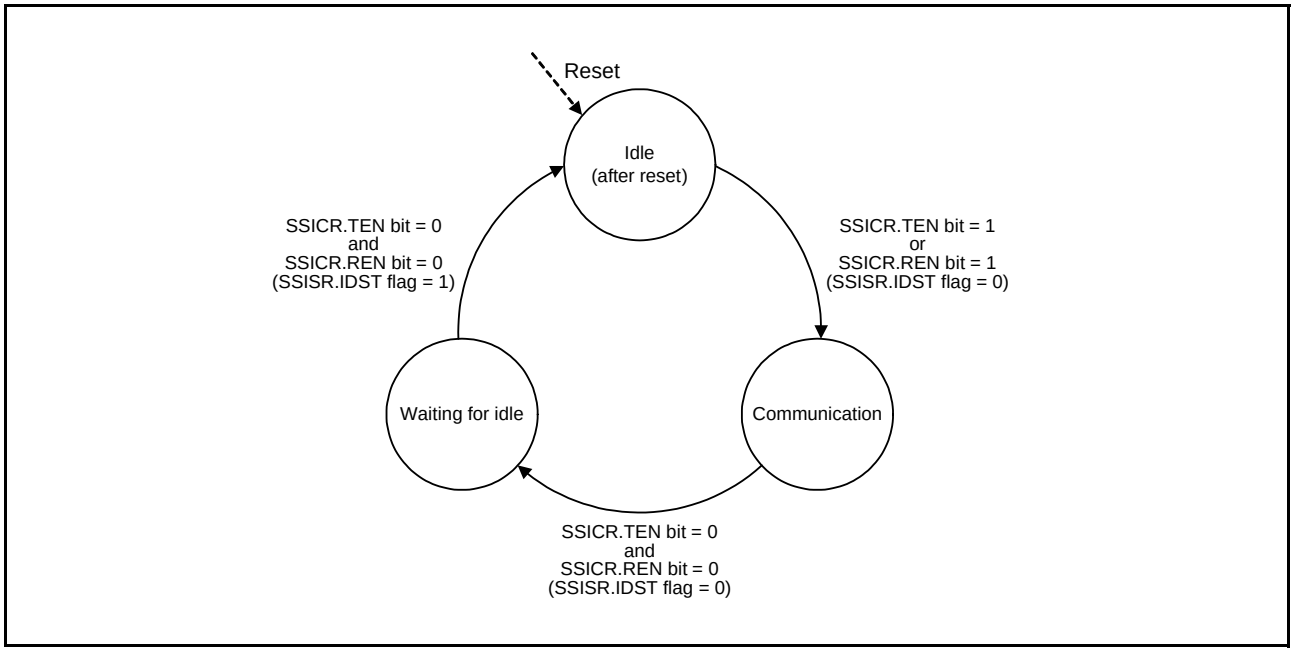


Figure 47.18 Operating State Transitions

(1) Idle State

This module enters this state when the MSTPCRD.MSTPD14 and MSTPD15 bits are set to 0 after a reset is released. All required configuration fields in the control register should be defined in this state. After the settings are made, the module enters communication state when the SSICR.TEN bit or SSICR.REN bit is set to 1.

(2) Communication State

Communication in this state depends on the selected operating state. For details, refer to section 47.3.5, Transmit Operation and section 47.3.6, Receive Operation.

(3) Waiting for Idle

This module enters this state when both the SSICR.TEN and SSICR.REN bits are set to 0 in communication state. If system word communication is completed in this state, the SSISR.IDST flag is set to 1 and this module enters the idle state.

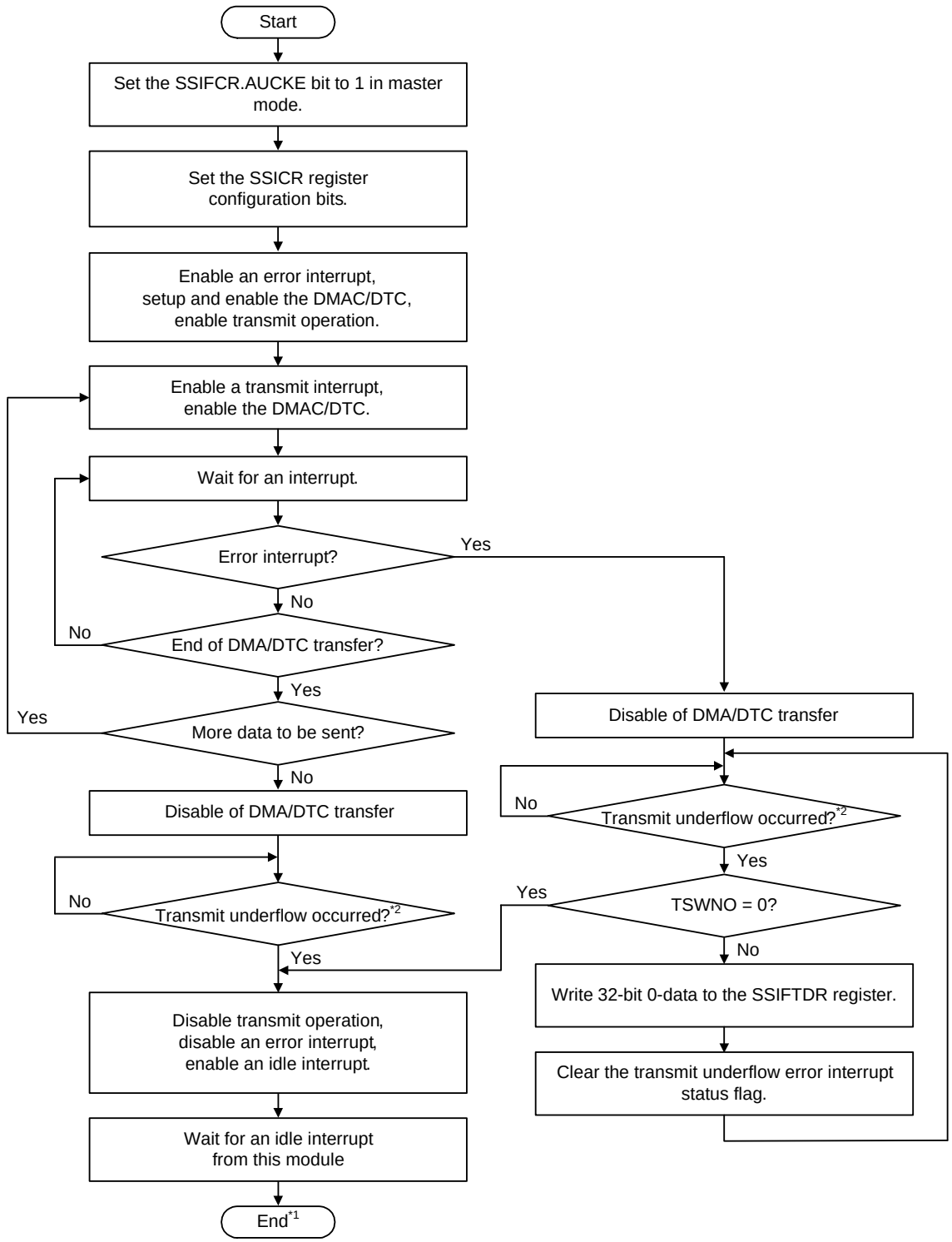
47.3.5 Transmit Operation

Transmission can be controlled either by DMA/DTC transfer or interrupt.
DMAC/DTC control is preferred to reduce the processor load. In transmission using the DMAC/DTC, the processor will only receive interrupts if there is an underflow or overflow of data or if DMA/DTC transfer has been completed. In transmission using DMA/DTC transfer, set the number of DMA/DTC transfers to multiples of 2 to write transmit data to the SSIFTDR register in 64-bit (two stages of FIFO) units.
The alternative method is using the interrupts that this module generates to supply data as required. In transmission using interrupts, write transmit data in 64-bit units regardless of the data format. If transmit data ends on a 32-bit boundary, write 00000000h after the last transmit data is written, and complete writing on a 64-bit boundary.
When stopping transmission, stop writing to the SSIFTDR register while 64-bit writing is completed. After writing is stopped, wait until a transmit underflow occurs before setting the SSICR.TEN bit to 0. During transmit underflow, the last data input to SSIFTDR register is continuously transmitted until this module enters the idle state. After setting the TEN bit to 0, continue to supply the clock*1 until the SSISR.IIRQ flag becomes 1 (in idle state). If a transmit underflow error or transmit overflow error occurs during data transmission, transmit data to SSIFTDR register may not be written in a 64-bit units. In that case, stop writing data, wait until a transmit underflow error occurs, and check the SSISR.TSWNO flag when the transmit underflow has occurred. When the TSWNO flag is 1, write 00000000h to SSIFTDR register and wait until an underflow occurs again. Once the TSWNO flag is confirmed to be 0, Set the TEN bit to 0 and continue to supply the clock*1 until the SSISR.IIRQ flag becomes 1 (in idle state).

Figure 47.19 shows transmission flow using the DMA/DTC, and Figure 47.20 shows transmission flow using interrupts.

Note 1. Input clock from the SSISCKn pin when SSICR.SCKD bit = 0 (n = 0, 1).
Master clock when SSICR.SCKD bit = 1.

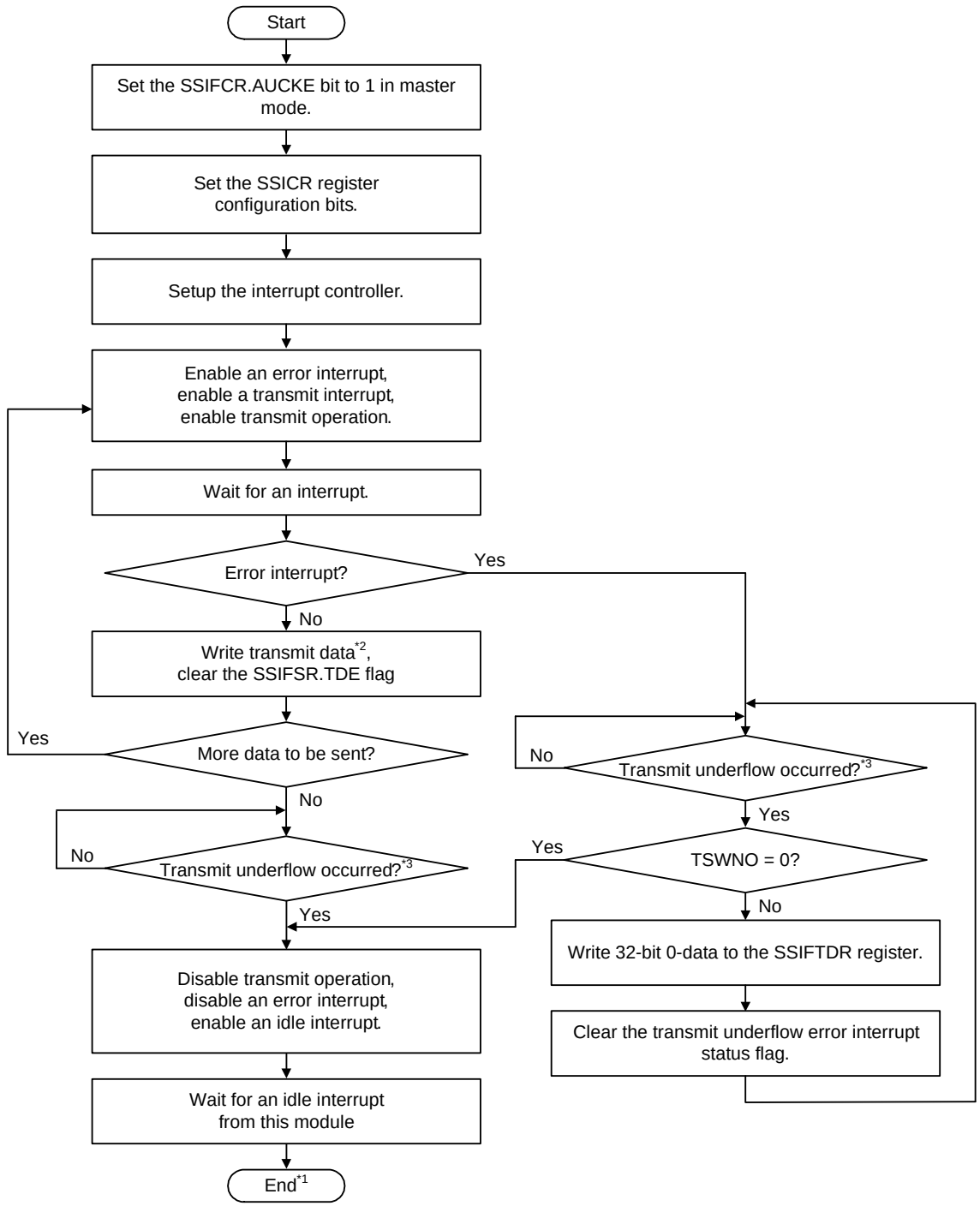
(1) Transmission Using the DMAC/DTC



Note 1. When restarting transmission after transmit operation is disabled (TEN = 0) while WS continue mode is disabled, execute a software reset and go back to the start of the flowchart again.
Note 2. Stop writing transmit data to the SSIFTDR register and wait until a transmit underflow occurs. During transmit underflow, the last data input to SSIFTDR register is continuously transmitted until the module is in the idle state.

Figure 47.19 Transmission Using the DMAC/DTC

(2) Transmission Using Interrupts



Note 1. When restarting transmission after transmit operation is disabled (TEN = 0) while WS continue mode is disabled, execute a software reset and go back to the start of the flowchart again.
Note 2. Write transmit data in 64-bit units regardless of the data word setting.
Note 3. Stop writing transmit data to the SSIFTDR register and wait until a transmit underflow occurs. During transmit underflow, the last data input to SSIFTDR register is continuously transmitted until the module is in the idle state.

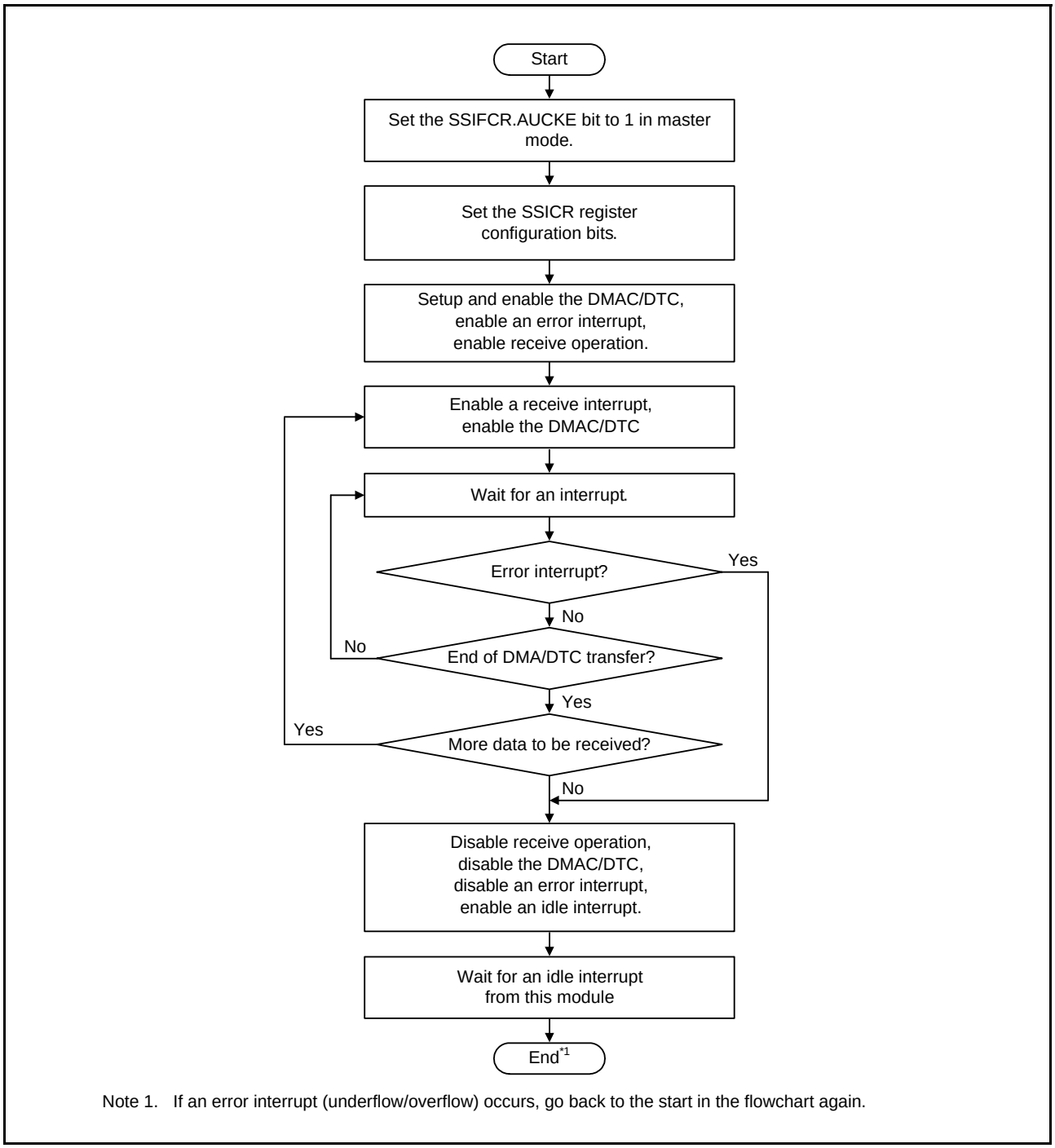
Figure 47.20 Transmission Using Interrupts

47.3.6 Receive Operation

Like transmission, reception can be controlled either by DMA/DTC transfer or interrupt.
Figure 47.21 and Figure 47.22 show the flow of operation.
When stopping reception, set the SSICR.REN bit to 0 and continue to supply the clock*1 until the SSISR.IIRQ flag becomes 1 (in idle state).

Note 1. Input clock from the SSISCKn pin when SSICR.SCKD bit = 0 (n = 0, 1).
Master clock when SSICR.SCKD bit = 1.

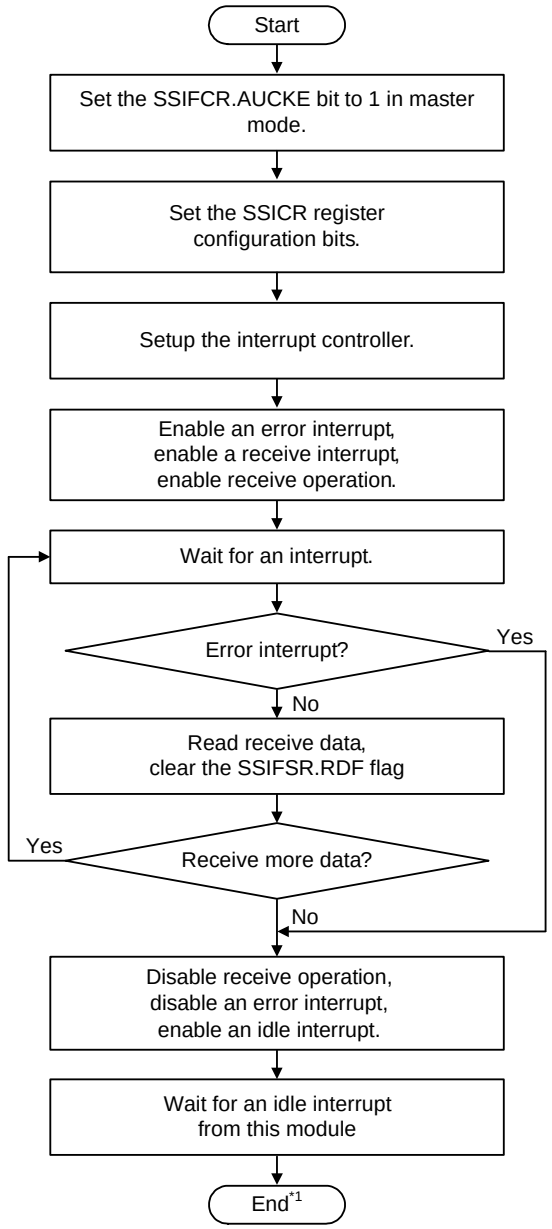
(1) Reception Using the DMAC/DTC



Note 1. If an error interrupt (underflow/overflow) occurs, go back to the start in the flowchart again.

Figure 47.21 Reception Using the DMAC/DTC

(2) Reception Using Interrupts



Note 1. If an error interrupt (underflow/overflow) occurs, go back to the start in the flowchart again.

Figure 47.22 Reception Using Interrupts

47.3.7 Serial Bit Clock Control

The SSI controls and selects the serial bit clock, according to the SSICR.SCKD and CKDV[3:0] bits setting.
If the serial bit clock direction is set to input (SCKD bit = 0), this module is in slave mode and the shift register uses the clock that was input to the SSISCKn pin as the bit clock (n = 0, 1).

If the serial bit clock direction is set to output (SCKD bit = 1), this module is in master mode, and the shift register uses the master clock (MCLK) or a divided master clock as the bit clock. The master clock is divided by the ratio specified by the SSICR.CKDV[3:0] bits for use as the bit clock by the shift register.
In either case the module pin, SSISCKn, is the same as the bit clock.

47.4 Interrupt Sources

Table 47.7 lists the interrupt sources of the SSI. Each interrupt source can be enabled or disabled by the SSICR.TUIEN, TOIEN, RUIEN, ROIEN and IIEN bits, and the SSIFCR.TIE and RIE bits.

Table 47.7 SSI Interrupt Sources

Channel	Interrupt Source	Description	Interrupt Status Flag	Interrupt Enable Bit	DMAC/DTC Start Trigger
SSI0	SSIF0	Transmit underflow interrupt Transmit overflow interrupt Receive underflow interrupt Receive overflow interrupt Idle interrupt	SSISR.TUIRQ SSISR.TOIRQ SSISR.RUIRQ SSISR.ROIRQ SSISR.IIRQ	SSICR.TUIEN SSICR.TOIEN SSICR.RUIEN SSICR.ROIEN SSICR.IIEN	Not available
	SSIRXI0	Receive data full interrupt (RXI)	SSIFSR.RDF	SSIFCR.RIE	Available
	SSITXI0	Transmit data empty interrupt (TXI)	SSIFSR.TDE	SSIFCR.TIE	Available
SSI1	SSIF1	Transmit underflow interrupt Transmit overflow interrupt Receive underflow interrupt Receive overflow interrupt Idle interrupt	SSISR.TUIRQ SSISR.TOIRQ SSISR.RUIRQ SSISR.ROIRQ SSISR.IIRQ	SSICR.TUIEN SSICR.TOIEN SSICR.RUIEN SSICR.ROIEN SSICR.IIEN	Not available
	SSIRTI1	Receive data full interrupt (RXI) Transmit data empty interrupt (TXI)	SSIFSR.RDF SSIFSR.TDE	SSIFCR.RIE SSIFCR.TIE	Available

47.5 Usage Notes

47.5.1 Setting the Module Stop Function

Module stop state can be entered or released using the MSTPCRD register. The initial setting of the SSI is in the module stop state. SSI register access is enabled by releasing the module stop state.
For details on the MSTPCRD register, refer to section 11, Low Power Consumption.

47.5.2 Notes on Changing Transmission Modes

For mode transitions between the transmitter, receiver, and transceiver while WS continue mode is disabled (SSITDMR.CONT = 0), set the SSICR.TEN and SSICR.REN bits to 0 and make a transition to the idle state once. Set the SSICR.TEN and SSICR.REN bits again while the module is in the idle state and restart transmission.

47.5.3 Limits on WS Continue Mode

If WS continue mode setting is changed, the operation of the SSISCKn and SSIWSn signals immediately after switching are not guaranteed (n = 0, 1). If it affects the device to be connected, do not change the setting dynamically.

47.5.4 Notes Regarding Clearing the Status Flag

The ROIRQ, RUIRQ, TOIRQ, and TUIRQ flags in the SSISR register can be set to 0 for only those that are read as 1. However, when reading any register in Table 47.8 while any of the flags is 1, the condition of reading the value 1 is satisfied even through the SSISR register is not read.
As a result, writing 0 to the applicable flag clears to the flag.
Similarly, regarding the RDF and TDE flags in the SSIFSR register, reading any register in Table 47.9 while either of the flag is 1, the condition of reading the value 1 is satisfied even though the SSIFSR register is not read. Just writing 0 to the applicable flag clears the flag.
Writing 0 only to the flag which is read as 1 when reading to clear the status flag.

Table 47.8 Registers that Satisfy the Condition of Reading the SSISR Register

Address	Module Symbol	Register Name	Register Symbol
0008 8004h	CMT0	Compare Match Timer Counter	CMCNT
0008 8044h	R12DA	D/A Control Register	DACR
0008 8144h	TPU3	Timer Interrupt Enable Register	TIER
0008 8204h	TMR0	Time Constant Register A	TCORA
0008 8204h	TMR01	Time Constant Register A	TCORA
0008 8304h	RIIC0	I^2C Bus Mode Register 3	ICMR3
0008 8344h	RIIC2	I^2C Bus Mode Register 3	ICMR3
0008 8544h	MMCIF	Interrupt Request Enable Register	CEINTEN
0008 9004h	S12AD	A/D Channel Select Register A0	ADANSA0
0008 9084h	S12AD	A/D Data Duplication Register A	ADDBLDRA
0008 9104h	S12AD1	A/D Channel Select Register A0	ADANSA0
0008 9144h	S12AD1	A/D Data Register 18	ADDR18
0008 9184h	S12AD1	A/D Data Duplication Register A	ADDBLDRA
0008 9E04h	QSPI	QSPI Data Register	SPDR
0008 A004h	SCI0	Serial Status Register	SSR
0008 A044h	SCI2	Serial Status Register	SSR
0008 A084h	SCI4	Serial Status Register	SSR
0008 A0C4h	SCI6	Serial Status Register	SSR
0008 AC44h	SDHI	SD Interrupt Mask Register 2	SDIMSK2
0008 ADC4h	SDHI	Version Register	SDVER
0008 B004h	CAC	CAC Status Register	CASTR
0008 B084h	DOC	DOC Data Setting Register	DODSR
0008 B104h	ELC	Event Link Setting Register 3	ELSR3
0008 B304h	SCI12	Serial Status Register	SSR
0008 C004h	PORT4	Port Direction Register	PDR
0008 C044h	PORT4	Port Input Register	PIDR
0008 C084h	PORT2	Open-Drain Control Register 0	ODR0
0008 C0C4h	PORT4	Pull-Up Resistor Control Register	PCR
0008 C104h	MPC	Address Output Enable Register 0	PFAOE0
0008 C1C4h	MPC	PG4 Pin Function Control Register	PG4PFS
0008 C404h	RTC	Minute Counter	RMINCNT
0008 C404h	RTC	Binary Counter 1	BCNT1
0008 C444h	RTC	Time Capture Control Register 2	RTCCR2
0008 C4C4h	POE3	Input Level Control/Status Register 2	ICSR2
0009 0844h	CAN0	Bit Configuration Register	BCR
0009 1844h	CAN1	Bit Configuration Register	BCR
0009 2844h	CAN2	Bit Configuration Register	BCR
0009 4204h	CMTW0	Timer Control Register	CMWCR
0009 4284h	CMTW1	Timer Control Register	CMWCR

Table 47.9 Registers that Satisfy the Condition of Reading the SSIFSR Register

Address	Module Symbol	Register Name	Register Symbol
0008 8014h	CMT2	Compare Match Timer Counter	CMCNT
0008 8114h	TPU0	Timer Interrupt Enable Register	TIER
0008 8154h	TPU4	Timer Interrupt Enable Register	TIER
0008 8214h	TMR2	Time Constant Register A	TCORA
0008 8214h	TMR23	Time Constant Register A	TCORA
0008 8514h	MMCIF	Transfer Block Setting Register	CEBLOCKSET
0008 9014h	S12AD	A/D Channel Select Register B0	ADANSB0
0008 9094h	S12AD	A/D Compare Channel Select Register 0	ADCMPANSR0
0008 9114h	S12AD1	A/D Channel Select Register B0	ADANSB0
0008 9194h	S12AD1	A/D Compare Channel Select Register 0	ADCMPANSR0
0008 9E14h	QSPI	QSPI Command Register 2	SPCMD2
0008 AC14h	SDHI	Block Count Register	SDBLKCNT
0008 B114h	ELC	Event Link Setting Register 19	ELSR19
0008 C094h	PORTA	Open-Drain Control Register 0	ODR0
0008 C154h	MPC	P24 Pin Function Control Register	P24PFS
0008 C194h	MPC	PA4 Pin Function Control Register	PA4PFS
0008 C414h	RTC	Hour Alarm Register	RHRAR
0008 C414h	RTC	Binary Counter 2 Alarm Register	BCNT2AR
0008 C454h	RTC	Minute Capture Register 0	RMINCP0
0008 C454h	RTC	BCNT1 Capture Register 0	BCNT1CP0
0008 C4D4h	POE3	Port Output Enable Control Register 6	POECR6
0009 0854h	CAN0	Time Stamp Register	TSR
0009 1854h	CAN1	Time Stamp Register	TSR
0009 2854h	CAN2	Time Stamp Register	TSR
0009 4214h	CMTW0	Compare Match Constant Register	CMWCOR
0009 4294h	CMTW1	Compare Match Constant Register	CMWCOR

48. Sample Rate Converter (SRC)

The sample rate converter is a module to convert the sample rate for data produced by various decoders such as WMA, MP3, or AAC.

48.1 Overview

Table 48.1 shows the specifications of the sample rate converter.

Table 48.1 Specifications of Sample Rate Converter

Item		Function
Data size		16 bits (stereo/monaural)
Sample rates	Input	8, 11.025, 12, 16, 22.05, 24, 32, 44.1, or 48 kHz is selectable.
	Output	8*1, 16*1, 32, 44.1, or 48 kHz is selectable.
Processing capacity		A sample output interval is a maximum of 7.7 µs (PCLKB = 60 MHz, 462 clocks).
SNR		80 dB or higher
Interrupt sources		Five Input FIFO empty, output FIFO full, output FIFO overflow, output FIFO underflow, and conversion end
DMA/DTC transfer sources		Two Input FIFO empty and output FIFO full
Module stop function		Power consumption can be reduced by stopping clock supply to this module when not used.

Note 1. Only when input of 44.1 kHz is selected.

Figure 48.1 shows a block diagram of the sample rate converter.

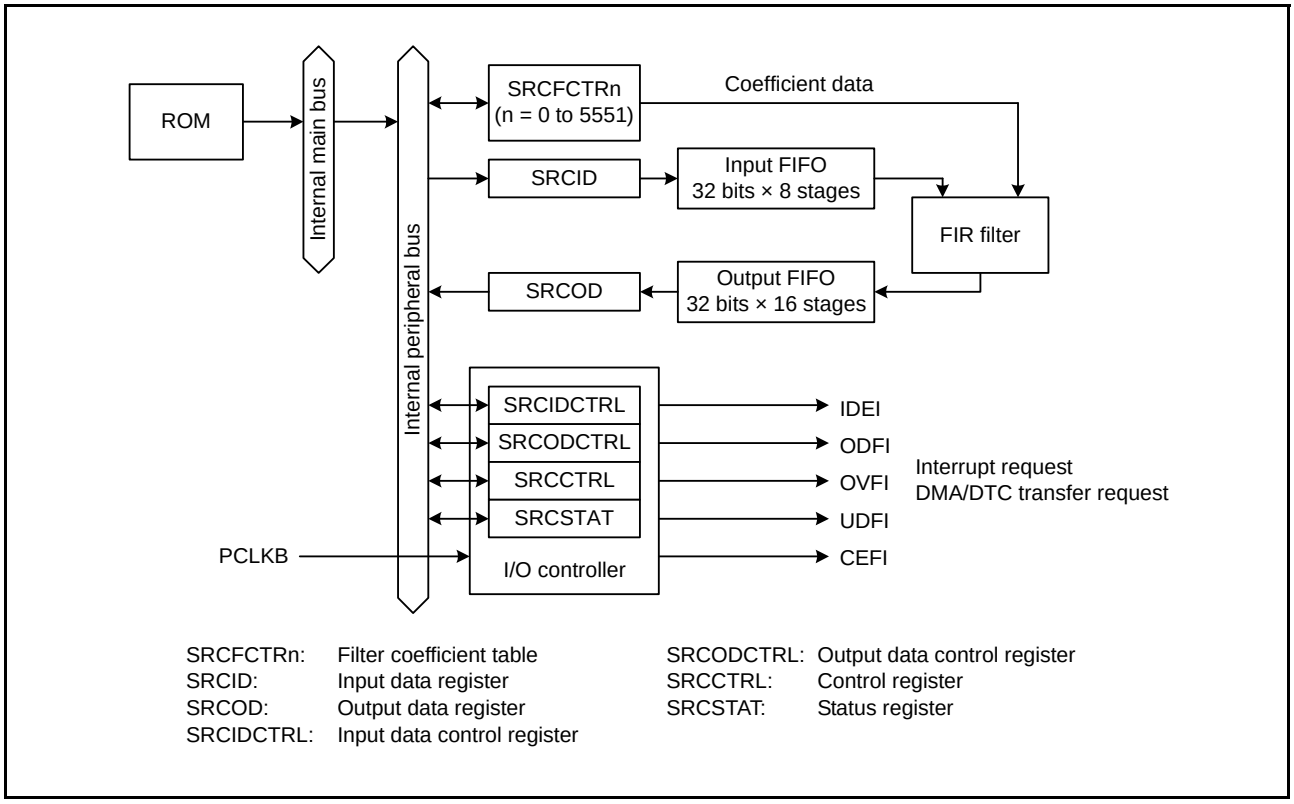


Figure 48.1 Block Diagram of Sample Rate Converter

48.2 Register Descriptions

48.2.1 Input Data Register (SRCID)

Address(es): SRC.SRCID 0009 DFF0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The SRCID register is a 32-bit readable/writable register that is used to input the data before sample rate conversion. This register is read as 0000 0000h. The data input to the SRCID register is stored in the 8-stage input FIFO. When the number of data in the input FIFO is 8, writing to the SRCID register has no effect.

For stereo data, bits 31 to 16 are for left channel data, and bits 15 to 0 are for right channel data. For monaural data, prepare valid data in bits 31 to 16 and dummy data in bits 15 to 0 before writing to the SRCID register in 32 bit units. The data arrangement must be changed by the setting of the SRCIDCTRL.IED bit when data to be converted are set to the SRCID register. Set the IED bit to 0 if the input data arrangement and the endian setting of the CPU are the same; set to 1 if these are different. For details on the endian setting of CPU, refer to section 7.2.5, Endian Select Register (MDE).

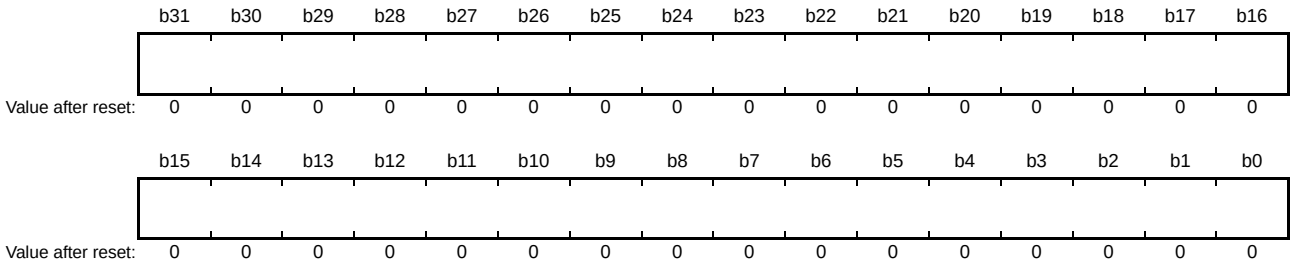
Table 48.2 shows the correspondence between the IED bit setting and data arrangement.

Table 48.2 Data Arrangement in the SRCID Register

SRCIDCTRL register	SRCID register			
IED bit	**b31 to b24**	**b23 to b16**	**b15 to b8**	**b7 to b0**
0	Lch[15:8]	Lch[7:0]	Rch[15:8]	Rch[7:0]
1	Lch[7:0]	Lch[15:8]	Rch[7:0]	Rch[15:8]

48.2.2 Output Data Register (SRCOD)

Address(es): 0009 DFF4h



The SRCOD register is a 32-bit read-only register used to output the data after sample rate conversion. The data in the 16-stage output FIFO is read through the SRCOD register. When the number of data in the output FIFO is zero after the start of conversion, the value previously read is read again.

The data arrangement in the SRCOD register depends on the settings of the OCH and OED bits in the SRCODCTRL register. Set the OED bit to 0 if the output data arrangement and the endian setting of the CPU are the same; set to 1 if these are different. For details on the endian setting of the CPU, refer to section 7.2.5, Endian Select Register (MDE).

Table 48.3 shows the correspondence between the OCH and OED bit setting and data arrangement in the SRCOD register.

Table 48.3 Data Arrangement in the SRCOD Register

SRCODCTRL register		SRCOD register			
OCH bit	**OED bit**	**b31 to b24**	**b23 to b16**	**b15 to b8**	**b7 to b0**
0	0	Lch[15:8]	Lch[7:0]	Rch[15:8]*1	Rch[7:0]*1
	1	Lch[7:0]	Lch[15:8]	Rch[7:0]*1	Rch[15:8]*1
1*2	0	Rch[15:8]	Rch[7:0]	Lch[15:8]	Lch[7:0]
	1	Rch[7:0]	Rch[15:8]	Lch[7:0]	Lch[15:8]

Note 1. When processing monaural data, the data in these bits is invalid. Discard the invalid data after reading from the SRCOD register in 32-bit units.

Note 2. When processing monaural data, do not set this bit to 1.

48.2.3 Input Data Control Register (SRCIDCTRL)

Address(es): SRC.SRCIDCTRL 0009 DFF8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IED	IEN	—	—	—	—	—	—	IFTRG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	IFTRG[1:0]	Input FIFO Data Triggering Number	b1 b0 0 0: 0 0 1: 2 1 0: 4 1 1: 6	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	IEN	Input FIFO Empty Interrupt Enable	0: Disables input FIFO empty interrupt requests. 1: Enables input FIFO empty interrupt requests.	R/W
b9	IED	Input Data Endian*2	0: Does not switch the byte endian (when the endian setting of the CPU (MDE*1) is the same as the endian of the input data). 1: Switch the byte endian (when the endian setting of the CPU (MDE*1) is different from the endian of the input data).	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. For MDE settings, refer to section 7.2.5, Endian Select Register (MDE).
Note 2. This bit should be rewritten while the SRCCTRL.SRCEN bit is 0.

The SRCIDCTRL register is a 16-bit readable/writable register that specifies the byte endian of input data, enables/disables the interrupt requests, and specifies the triggering number of data.

IFTRG[1:0] Bits (Input FIFO Data Triggering Number)

These bits specify the condition in terms of the number on which the SRCSTAT.IINT flag is set to 1. When the number of data stored in the input FIFO becomes equal to or less than the specified triggering number, the IINT flag is set to 1.

IEN Bit (Input FIFO Empty Interrupt Enable)

Enables/disables the input FIFO empty interrupt request to be issued when the number of data in the input FIFO becomes equal to or less than the triggering number specified by the IFTRG[1:0] bits, thus resulting in the SRCSTAT.IINT flag being set to 1.

IED Bit (Input Data Endian)

Specifies the byte endian of the input data. Set this bit to 0 if the endian of the input data is the same as the endian setting of the CPU; set to 1 if these are different. For details on the endian setting of the CPU, refer to section 7.2.5, Endian Select Register (MDE).

48.2.4 Output Data Control Register (SRCODCTRL)

Address(es): SRC.SRCODCTRL 0009 DFFAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	OCH	OED	OEN	—	—	—	—	—	—	OFTRG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OFTRG[1:0]	Output FIFO Data Trigger Number	b1 b0 0 0: 1 0 1: 4 1 0: 8 1 1: 12	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	OEN	Output FIFO Full Interrupt Enable	0: Disables output FIFO full interrupt requests. 1: Enables output FIFO full interrupt requests.	R/W
b9	OED	Output Data Endian	0: Does not switch the byte endian (when the endian setting of the CPU (MDE*1) is the same as the endian of the output data). 1: Switch the byte endian (when the endian setting of the CPU (MDE*1) is different from the endian of the output data).	R/W
b10	OCH	Output Data Channel Switching*2	0: Does not switch the data arrangement of the left and right channels (the same order of the input data) 1: Switch the data arrangement of the right and left channels (the reversed order of the input data)	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. For MDE settings, refer to section 7.2.5, Endian Select Register (MDE).
Note 2. This bit should be rewritten while the SRCCTRL.SRCEN bit is 0.

The SRCODCTRL register is a 16-bit readable/writable register that specifies whether to switch the channels for the output data, specifies the byte endian of output data, enables/disables the interrupt requests, and specifies the triggering number of data.

OFTRG[1:0] Bits (Output FIFO Data Trigger Number)

These bits specify the condition in terms of the number on which the SRCSTAT.OINT flag is set to 1. When the number of data in the output FIFO becomes equal to or greater than the specified triggering number, the OINT flag is set to 1.

OEN Bit (Output FIFO Full Interrupt Enable)

Enables/disables the output FIFO full interrupt request to be issued when the number of data in the output FIFO becomes equal to or greater than the number specified by the OFTRG[1:0] bits, thus resulting in the SRCSTAT.OINT flag being set to 1.

OED Bit (Output Data Endian)

Specifies the byte endian of the output data. Set this bit to 0 if the endian of the output data is the same as the endian setting of the CPU; set to 1 if these are different. For details on the endian setting of the CPU, refer to section 7.2.5, Endian Select Register (MDE).

OCH Bit (Output Data Channel Switching)

Specifies whether to switch the data arrangement of the left and right channels for the output data register (SRCOD). When processing monaural data, set this bit to 0.

48.2.5 Control Register (SRCCTRL)

Address(es): SRC.SRCCTRL 0009 DFFCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FICRAE	—	CEEN	SRCEN	UDEN	OVEN	FL	CL	IFS[3:0]				—	OFS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	OFS[2:0]	Output Sample Rate	b2 b1 b0 0 0 0: 44.1 kHz 0 0 1: 48.0 kHz 0 1 0: 32.0 kHz 1 0 0: 8.0 kHz*1 1 0 1: 16.0 kHz*1 Settings other than above are prohibited.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7 to b4	IFS[3:0]	Input Sample Rate	b7 b6 b5 b4 0 0 0 0 : 8.0 kHz 0 0 0 1 : 11.025 kHz 0 0 1 0 : 12.0 kHz 0 1 0 0 : 16.0 kHz 0 1 0 1 : 22.05 kHz 0 1 1 0 : 24.0 kHz 1 0 0 0 : 32.0 kHz 1 0 0 1 : 44.1 kHz 1 0 1 0 : 48.0 kHz Settings other than above are prohibited.	R/W
b8	CL	Internal Work Memory Clear	Writing 1 to this bit clears the input FIFO, output FIFO, input buffer memory, intermediate memory, and accumulator.	R/W
b9	FL	Internal Work Memory Flush	Writing 1 to this bit starts converting the sample rate of all data in the input FIFO, input buffer memory, and intermediate memory (i.e., flush processing).	R/W
b10	OVEN	Output FIFO Overflow Interrupt Enable	0: Disables output FIFO overflow interrupt requests. 1: Enables output FIFO overflow interrupt requests.	R/W
b11	UDEN	Output FIFO Underflow Interrupt Enable	0: Disables output FIFO underflow interrupt requests. 1: Enables output FIFO underflow interrupt requests.	R/W
b12	SRCEN	Module Enable	0: Disables this module operation. 1: Enables this module operation.*2	R/W
b13	CEEN	Conversion End Interrupt Enable	0: Disables conversion end interrupt requests. 1: Enables conversion end interrupt requests.	R/W
b14	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15	FICRAE	Filter Coefficient Table Access Enable	0: Reading/writing to filter coefficient table RAM is disabled. 1: Reading/writing to filter coefficient table RAM is enabled.	R/W

Note 1. Valid only when the IFS[3:0] bits are 1001b.
Note 2. When the SRCEN bit is 1, do not change the settings of the following bits:
SRCIDCTRL.IED bit, SRCODCTRL.OED and OCH bits, SRCCTRL.OFS[2:0], IFS[3:0], and FICRAE bits.

The SRCCTRL register is a 16-bit readable/writable register that enables/disables the access to the filter coefficient table, module operations, and interrupt requests, and specifies flush processing, clear processing of the internal work memory, and the input and output sample rates.

OFS[2:0] Bits (Output Sample Rate)

These bits specify the output sample rate.

IFS[3:0] Bits (Input Sample Rate)

These bits specify the input sample rate.

CL Bit (Internal Work Memory Clear)

Writing 1 to this bit clears the input FIFO, output FIFO, input buffer memory, intermediate buffer memory, and accumulator, then the CL bit is set to 0. This bit is read as 0. Even if the SRCEN bit is 0, writing 1 to this bit clears the processing.

FL Bit (Internal Work Memory Flush)

Writing 1 to this bit starts converting the sample rate of all data in the input FIFO, input buffer memory, and intermediate memory (i.e., flush processing). This bit is read as 0. When the SRCEN bit is 0, writing 1 to this bit does not trigger flush processing.
In addition, when 1 is written to the FL bit if the number of the input data is less than the values shown in Table 48.6, the effective output data cannot be obtained. Thus the internal work memory is cleared without triggering the flush processing.

OVEN Bit (Output FIFO Overflow Interrupt Enable)

Enables/disables the output FIFO overflow interrupt request to be issued when the SRCSTAT.OVF flag is set to 1.
When the OVEN bit is 1: Conversion processing is stopped until the OVF flag is cleared by the CPU accessing to SRCSTAT when the output FIFO overflow interrupt is generated. At this time, conversion result writing to the output FIFO is also stopped.
When the OVEN bit is 0: The OVF flag is automatically cleared when the output FIFO has space, and conversion processing can be continued.

UDEN Bit (Output FIFO Underflow Interrupt Enable)

Enables/disables the output FIFO underflow interrupt to be generated when the SRCSTAT.UDF flag is set to 1.

SRCEN Bit (Module Enable)

Enables/disables this module operation. Rewriting the SRCEN bit from 0 to 1 clears the internal work memory.

CEEN Bit (Conversion End Interrupt Enable)

Enables/disables a conversion end interrupt request to be issued when the SRCSTAT.CEF flag is set to 1 after flush processing is completed and all output data is read.

FICRAE Bit (Filter Coefficient Table Access Enable)

Enables/disables the access to the filter coefficient table RAM.

After flush processing has been completed, the number of output data obtained as a result of conversion can be calculated by using the following formula.
The value of n can be obtained from Table 48.4. The number of input data should be equal to or greater than the values in Table 48.5.

$$\frac{\text{Number of output data} - 1}{\text{Output sample rate}} = \frac{\text{Number of input data} \times n - 1}{\text{Input sample rate} \times n}$$

$$\text{Number of output data} = \left[(\text{Number of input data} \times n - 1) \times \frac{\text{Output sample rate}}{\text{Input sample rate} \times n} \right] + 1$$

Table 48.4 Sample Rate Settings and the Value of n

OFS[2:0] Bit Setting (Output Sample Rate [kHz])	IFS[3:0] Bit Setting (Input Sample Rate [kHz])								
	0000b (8.0)	0001b (11.025)	0010b (12.0)	0100b (16.0)	0101b (22.05)	0110b (24.0)	1000b (32.0)	1001b (44.1)	1010b (48.0)
000b (44.1)	6	4	4	3	2	2	3	—	1
001b (48.0)	6	4	4	3	2	2	3	1	—
010b (32.0)	4	8	4	2	4	2	—	2	1
100b (8.0)	—	—	—	—	—	—	—	1	—
101b (16.0)	—	—	—	—	—	—	—	1	—

Conversion processing is not started and thus output data is not obtained until the specified number of data are input. The minimum number of input data necessary for obtaining the first output data depends on the IFS[3:0] and OFS[2:0] bit settings. Table 48.5 shows the relation between the settings of the IFS[3:0] and OFS[2:0] bits and the number of initial input data required. Table 48.6 shows the relation between the settings of the IFS[3:0] and OFS[2:0] bits and the number of initial input data required for flush processing.

Table 48.5 Relation between Sample Rate Settings and Number of Initial Input Data Required

OFS[2:0] Bit Setting (Output Sample Rate [kHz])	IFS[3:0] BIt Setting (Input Sample Rate [kHz])								
	0000b (8.0)	0001b (11.025)	0010b (12.0)	0100b (16.0)	0101b (22.05)	0110b (24.0)	1000b (32.0)	1001b (44.1)	1010b (48.0)
000b (44.1)	38	40	40	43	48	48	43	—	63
001b (48.0)	38	40	40	43	48	48	43	32	—
010b (32.0)	40	37	40	48	40	48	—	48	63
100b (8.0)	—	—	—	—	—	—	—	63	—
101b (16.0)	—	—	—	—	—	—	—	63	—

Table 48.6 Relation between Sample Rate Settings and Number of Input Data Required for Flush Processing

OFS[2:0] Bit Setting (Output Sample Rate [kHz])	IFS[3:0] Bit Setting (Input Sample Rate [kHz])								
	0000b (8.0)	0001b (11.025)	0010b (12.0)	0100b (16.0)	0101b (22.05)	0110b (24.0)	1000b (32.0)	1001b (44.1)	1010b (48.0)
000b (44.1)	27	24	24	22	16	16	22	—	1
001b (48.0)	27	24	24	22	16	16	22	32	—
010b (32.0)	24	29	24	16	24	16	—	16	1
100b (8.0)	—	—	—	—	—	—	—	1	—
101b (16.0)	—	—	—	—	—	—	—	1	—

48.2.6 Status Register (SRCSTAT)

Address(es): SRC.SRCSTAT 0009 DFFEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OFDN[4:0]					IFDN[3:0]				—	CEF	FLF	UDF	OVF	IINT	OINT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	OINT	Output FIFO Full Interrupt Request Flag	0: Number of data in the output FIFO has not become equal to or greater than the specified triggering number. 1: Number of data in the output FIFO has become equal to or greater than the specified triggering number.	R/(W) *1
b1	IINT	Input FIFO Empty Interrupt Request Flag	0: Number of data in the input FIFO has not become equal to or smaller than the specified triggering number. 1: Number of data in the input FIFO has become equal to or smaller than the specified triggering number.	R/(W) *1
b2	OVF	Output FIFO Overflow Interrupt Request Flag	0: No output FIFO overflow 1: An output FIFO overflow has occurred.	R/(W) *1
b3	UDF	Output FIFO Underflow Interrupt Request Flag	0: No output FIFO underflow 1: An output FIFO underflow has occurred.	R/(W) *1
b4	FLF	Flush Processing Status Flag	0: Flush processing is completed. 1: Flush processing is in progress.	R
b5	CEF	Conversion End Flag	0: All of the output data has not been read out. 1: All of the output data has been read out.	R/(W) *1
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b7	IFDN[3:0]	Input FIFO Data Count	Indicate the number of data in the input FIFO.	R
b15 to b11	OFDN[4:0]	Output FIFO Data Count	Indicate the number of data in the output FIFO.	R

Note 1. Only 0 can be written after having read as 1.

The SRCSTAT register is a 16-bit readable/writable register that indicates the number of data in the input and output FIFOs, whether the various interrupt sources have been generated or not, and the flush processing status.

OINT Flag (Output FIFO Full Interrupt Request Flag)

Indicates that the number of data in the output FIFO has become equal to or greater than the triggering number specified by the SRCODCTRL.OFTRG[1:0] bits.

[Setting condition]

- When the number of data in the output FIFO has become equal to or greater than the specified triggering number.

[Clearing conditions]

- When 0 has been written to the OINT flag after confirming the OINT flag is 1.
- When the last DMA/DTC transfer is executed.
- When 1 has been written to the SRCCTRL.CL bit.
- When the SRCCTRL.SRCEN bit has been rewritten from 0 to 1.

IINT Flag (Input FIFO Empty Interrupt Request Flag)

Indicates that the number of data in the input FIFO has become equal to or smaller than the triggering number specified by the SRCIDCTRL.IFTRG[1:0] bits.

[Setting conditions]

- When the number of data in the input FIFO has become equal to or smaller than the specified triggering number.
- When 1 has been written to the SRCCTRL.CL bit.

- When the SRCCTRL.SRCEN bit has been rewritten from 0 to 1.

[Clearing conditions]

- When 0 has been written to the IINT flag after confirming the IINT flag is 1.
- When the last DMA/DTC transfer is executed.

OVF Flag (Output FIFO Overflow Interrupt Request Flag)

Indicates that data has been written when the output FIFO is full. The conversion is stopped until the OVF flag is cleared.

[Setting condition]

- When data has been written when the output FIFO is full.

[Clearing conditions]

- If the SRCCTRL.OVEN bit is 1, when 0 has been written to the OVF flag after confirming the OVF flag is 1.
- If the SRCCTRL.OVEN bit is 0, when the number of data in the output FIFO decreases after reading the SRCOD register.
- When 1 has been written to the SRCCTRL.CL bit.
- When the SRCCTRL.SRCEN bit has been rewritten from 0 to 1.

UDF Flag (Output FIFO Underflow Interrupt Request Flag)

Indicates that data has been read when the output FIFO is empty.

[Setting condition]

- When data has been read when the output FIFO is empty.

[Clearing conditions]

- When 0 has been written to the UDF flag after confirming the UDF flag is 1.
- When 1 has been written to the SRCCTRL.CL bit.
- When the SRCCTRL.SRCEN bit has been rewritten from 0 to 1.

FLF Flag (Flush Processing Status Flag)

Indicates whether flush processing is in progress or not.

[Setting condition]

- When 1 has been written to the SRCCTRL.FL bit.
  (When flush processing is not in progress, however, the FLF flag is not set to 1).

[Clearing conditions]

- When flush processing has been completed.
- When 1 has been written to the SRCCTRL.CL bit.
- When the SRCCTRL.SRCEN bit has been rewritten from 0 to 1.

CEF Flag (Conversion End Flag)

Indicates that all output data is read after flush processing is completed.

[Setting condition]

- When the number of data in the output FIFO is zero on completion of flush processing.

[Clearing conditions]

- When 0 has been written to the CEF flag after confirming the CEF flag is 1.
- When 1 has been written to the SRCCTRL.CL bit.
- When the SRCCTRL.SRCEN bit has been rewritten from 0 to 1.

IFDN[3:0] Bits (Input FIFO Data Count)

These bits indicate the number of data in the input FIFO.

OFDN[4:0] Bits (Output FIFO Data Count)

These bits indicate the number of data in the output FIFO.

48.2.7 Filter Coefficient Table n (SRCFCTRn) (n = 0 to 5551)

Address(es): SRC.SRCFCTR0 to SRC.SRCFCTR5551 0009 8000h to 0009 D6BFh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	SRCFCOE[21:16]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SRCFCOE[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

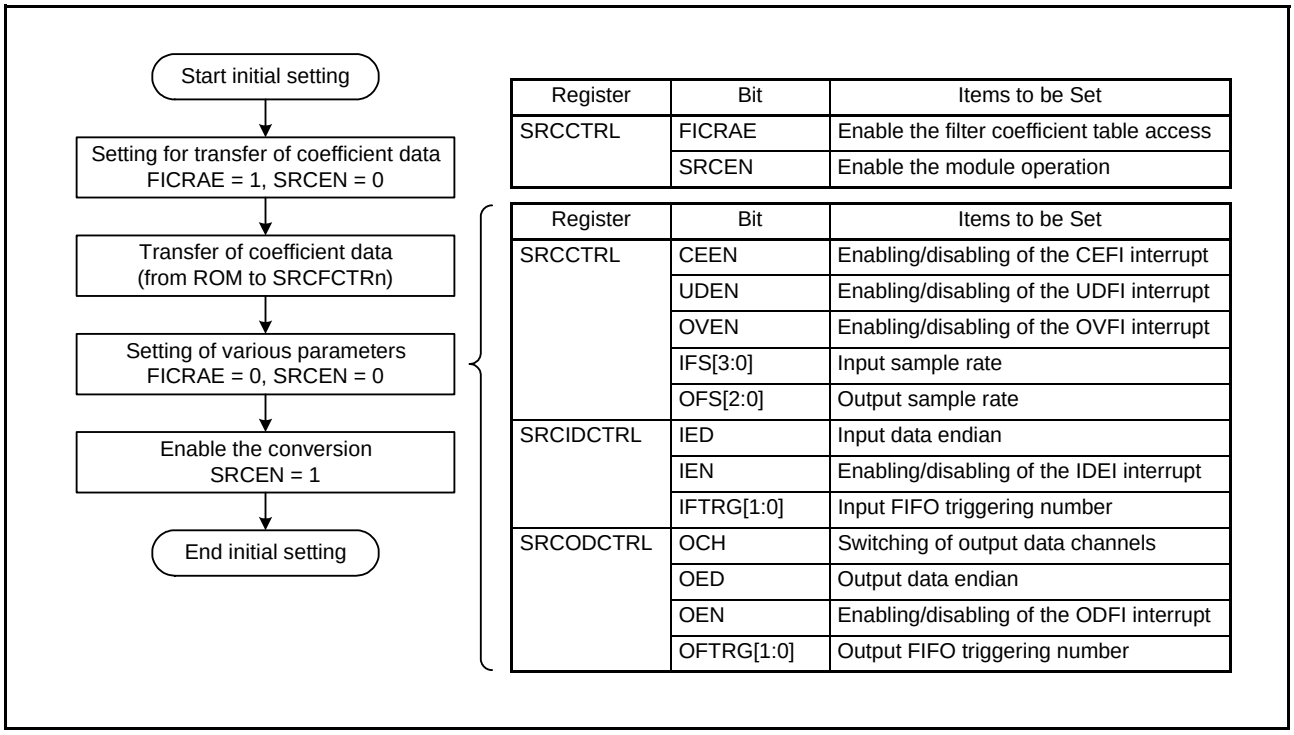
Bit	Symbol	Bit Name	Description	R/W
b21 to b0	SRCFCOE[21:0]	Filter Coefficient Table	Stores a filter coefficient value.	R/W
b31 to b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SRCFCTR0 to SRCFCTR5551 are 32-bit readable/writable RAM that store the filter coefficient to be used for sample rate conversion. This RAM can be read from and written to through the peripheral bus only when the SRCCTRL.FICRAE bit is 1 and the SRCCTRL.SRCEN bit is 0.

48.3 Operation

48.3.1 Initial Setting

Figure 48.2 shows a sample flowchart for initial setting. After the module stop state is released, the filter coefficient data stored in ROM and other areas needs to be transferred to the filter coefficient table (SRCFCTRn) before SRC conversion starts. When the filter coefficient data has been already stored in the filter coefficient table, skip the transfer mentioned above and set the various necessary parameters to start the conversion.



Register	Bit	Items to be Set
SRCCTRL	FICRAE	Enable the filter coefficient table access
	SRCEN	Enable the module operation

Register	Bit	Items to be Set
SRCCTRL	CEEN	Enabling/disabling of the CEFI interrupt
	UDEN	Enabling/disabling of the UDFI interrupt
	OVEN	Enabling/disabling of the OVFI interrupt
	IFS[3:0]	Input sample rate
	OFS[2:0]	Output sample rate
SRCIDCTRL	IED	Input data endian
	IEN	Enabling/disabling of the IDEI interrupt
	IFTRG[1:0]	Input FIFO triggering number
SRCODCTRL	OCH	Switching of output data channels
	OED	Output data endian
	OEN	Enabling/disabling of the ODFI interrupt
	OFTRG[1:0]	Output FIFO triggering number

Figure 48.2 Sample Flowchart for Initial Setting

48.3.2 Data Input

Figure 48.3 is a sample flowchart for data input.

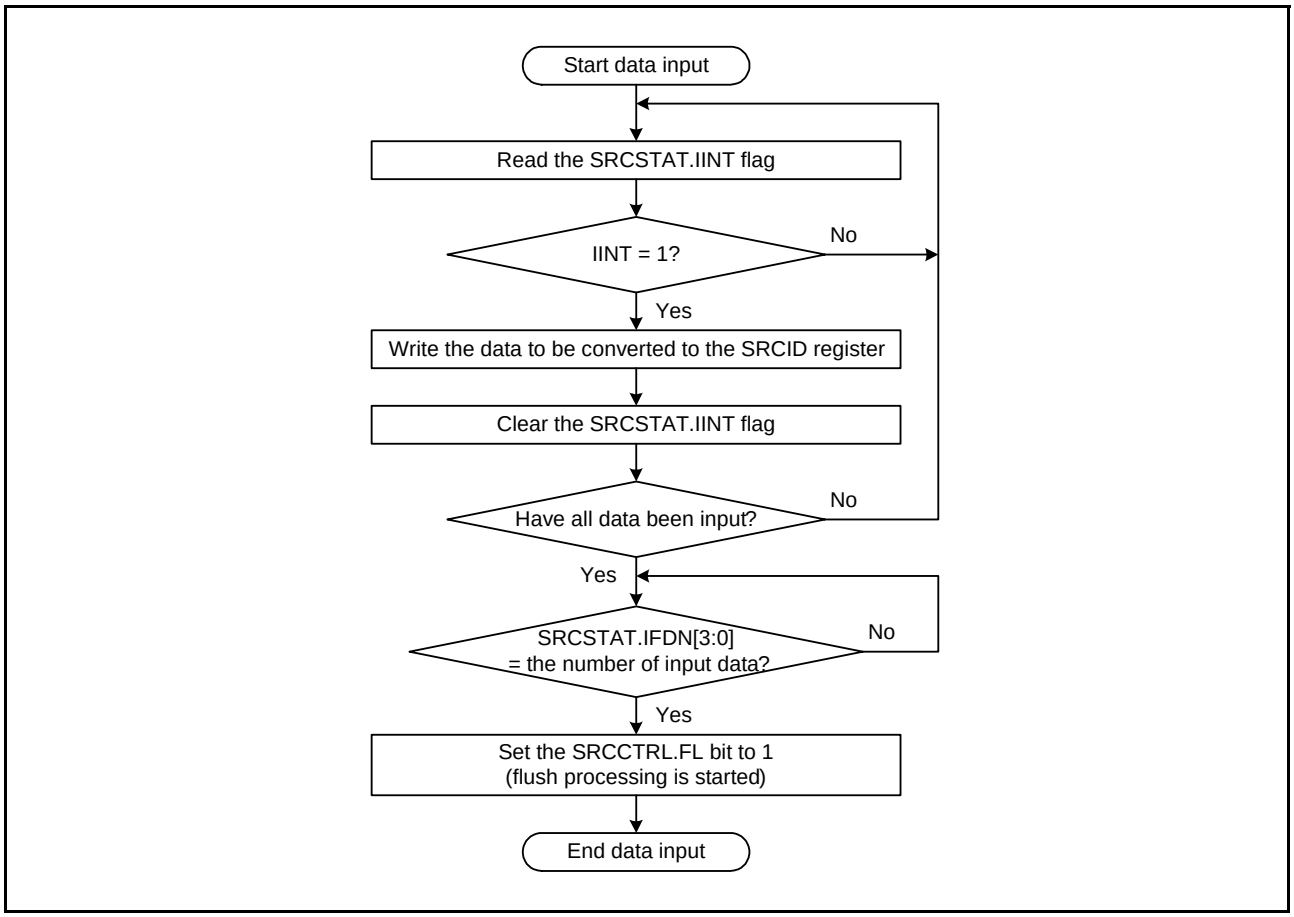


Figure 48.3 Flowchart for Data Input

(1) When Interrupts are Issued to CPU

1. Set the SRCIDCTRL.IEN bit to 1.
2. When the SRCSTAT.IINT flag is set to 1, the IDEI interrupt request is generated. In the interrupt processing routine, confirm that the IINT flag is 1, write data to the SRCID register, and write 0 to the IINT flag. Then return from the interrupt processing routine.
3. Repeat step 2 to complete input of all data.
4. Confirm that the value of the SRCSTAT.IFDN[3:0] bits matches with the number of data written to the SRCID register.
5. Write 1 to the SRCCTRL.FL bit.

(2) When Interrupts are Used to Activate DMAC

1. Set the request source for one of the channels of the DMAC as an IDEI interrupt request of this module.
2. Set the SRCIDCTRL.IEN bit to 1.
3. When the SRCSTAT.IINT flag is set to 1, an IDEI interrupt request is generated and the DMAC starts. When data is written to the SRCID register using DMA transfer and the number of data in the input FIFO exceeds the triggering number specified by the SRCIDCTRL.IFTRG[1:0] bits, the SRCSTAT.IINT flag becomes 0.
4. Repeat step 3 to complete input of all data, and write 1 to the SRCCTRL.FL bit.

(3) When Serial Sound Interface Interrupts are Used for Starting DMAC to Transfer Input Data from Serial Sound Interface

1. Set the request source for one of the channels of the DMAC as the serial sound interface. Set the SSIFRDR register of the serial sound interface as a transfer source and the SRCID register of the sample rate converter as a transfer destination, and set the serial sound interface to enable receive operation.
2. When the SSIFSR.RDF flag is set to 1, the serial sound interface interrupt request is generated and the DMAC starts. The DMAC then reads data from the SSIFRDR register and writes the data to the SRCID register.
3. Repeat step 2 to complete input of all data, and write 1 to the SRCCTRL.FL bit.

Note: The input FIFO has eight stages. The number of the data that can be transferred (i.e., the empty space in the FIFO) when an IDEI interrupt request is generated depends on the settings of the SRCIDCTRL.IFTRG[1:0] bits. Since the input FIFO is not equipped with a function to prevent or detect an overflow, the transferred data will be destroyed when an overflow occurs. When setting the number of consecutive data transfers by the DMAC, consider the settings of the SRCCTRL.IFTRG[1:0] bits.

48.3.3 Data Output

Figure 48.3 is a sample flowchart for data output.

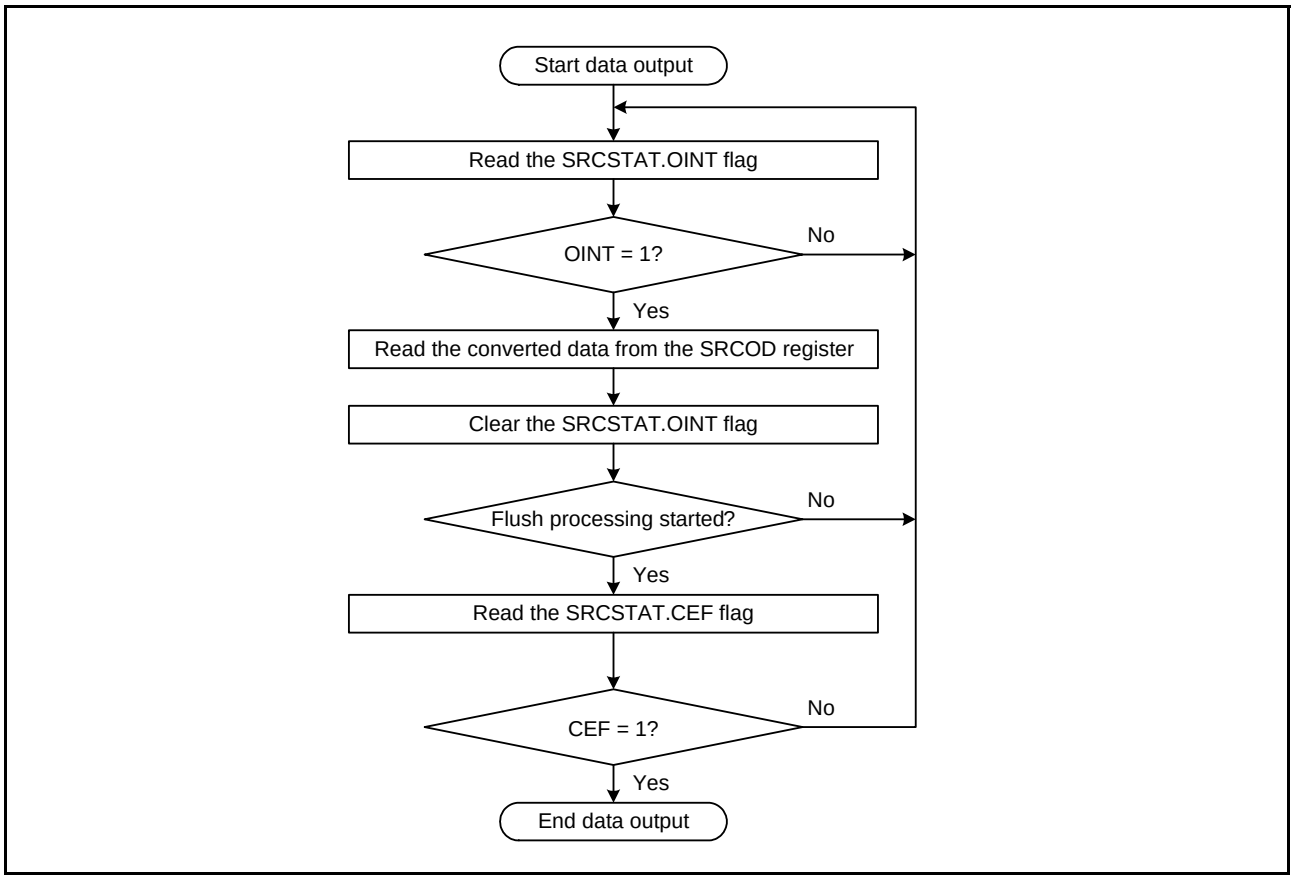


Figure 48.4 Flowchart for Data Output

(1) When Interrupts are Issued to CPU

1. Set the SRCODCTRL.OEN bit to 1.
2. When the SRCSTAT.OINT flag is set to 1, the ODFI interrupt request is generated. In the interrupt processing routine, confirm that the OINT flag is 1, read data from the SRCOD register, and write 0 to the OINT flag. Then return from the interrupt processing routine.
3. After flush processing starts, repeat step 2 until the SRCSTAT.CEF flag is read as 1.

(2) When Interrupts are Used to Activate DMAC

1. Assign the ODFI interrupt request of this module to one channel of the DMAC as a DMA transfer request source.
2. Set the SRCODCTRL.OEN bit to 1.
3. When the SRCSTAT.OINT flag is set to 1, the ODFI interrupt request is generated and the DMAC starts. When data is read from the SRCOD register using DMA transfer and the number of data in the output data FIFO becomes equal to or less than the triggering number specified by the SRCODCTRL.OFTRG[1:0] bits, the SRCSTAT.OINT flag becomes 0.
4. After flush processing starts, repeat step 3 until the SRCSTAT.FLF flag is read as 0.

(3) When Serial Sound Interface Interrupts are Used for Starting DMAC to Transfer Output Data to Serial Sound Interface

1. Set the SRCCTRL.OVEN bit to 0 to disable the OVFI interrupt request generation.
2. Assign the serial sound interface to one channel of the DMAC as a DMA transfer request source. Set the SRCOD register of the sample rate converter as a transfer source and the SSIFTDR register of the serial sound interface as a transfer destination, and set the serial sound interface to enable transmit operation.
3. When the SSIFSR.TDE flag is set to 1, the serial sound interface interrupt request is generated and the DMAC starts. The DMAC then reads data from the SRCOD register and writes the data to the SSIFTDR register.
4. After flush processing starts, repeat step 3 until the SRCSTAT.CEF flag is read as 1.

Note 1. The output FIFO has 16 stages. The conversion will be stopped when no data is read and overflow occurs in the output FIFO. Even in the state with overflow, data can be read from the output FIFO, but the procedure to restart conversion may be required depending on the settings. (For details, see the SRCCTRL.OVEN bit).

Note 2. When the number of data in the output FIFO is zero, incorrect data will be read. Thus take the settings of the SRCODCTRL.OFTRG[1:0] bits into consideration when setting the number of data to be continuously transferred by the DMAC.

48.4 Interrupts

This module has five interrupt sources: input FIFO empty (IDEI), output FIFO full (ODFI), output FIFO overflow (OVFI), output FIFO underflow (UDFI), and conversion end (CEFI). Table 48.7 lists interrupt request types and generation conditions.

Table 48.7 Interrupt Requests and Generation Conditions

Interrupt Request	Abbreviation	Interrupt Condition	DMA/DTC Request
Input FIFO empty	IDEI	(IINT = 1), (IEN = 1), and (SRCEN = 1)	Possible
Output FIFO full	ODFI	(OINT = 1), (OEN = 1), and (SRCEN = 1)	Possible
Output FIFO overflow	OVFI	(OVF = 1), (OVEN = 1), and (SRCEN = 1)	Not possible
Output FIFO underflow	UDFI	(UDF = 1), (UDEN = 1), and (SRCEN = 1)	Not possible
Conversion end	CEFI	(CEF = 1), (CEEN = 1), and (SRCEN = 1)	Not possible

When the condition for interrupt generation is satisfied, an interrupt request is output to the ICU.
The IDEI and ODFI interrupts can start the DMA/DTC transfer. If the DMAC or DTC is selected as an interrupt request destination, the interrupts from this module are not sent to the CPU.
Do not clear the IINT and OINT flags through writing by the CPU (i.e., writing 0 after reading 1) during the DMA/DTC transfer.

48.5 Usage Notes

48.5.1 Notes on Accessing Registers (1)

The specifications of access cycles depends on the register or filter coefficient table (RAM). For the number of access cycles, refer to section 5.1, I/O Register Addresses (Address Order).

48.5.2 Notes on Accessing Registers (2)

It takes three cycles of the peripheral module clock (PCLKB) from writing to the SRCCTRL register to updating the SRCSTAT register in the following cases:

- Until the SRCSTAT.FLF flag becomes 1 after 1 is written to the SRCCTRL.FL bit
- Until each bit in the SRCSTAT register is initialized after 1 is written to the SRCCTRL.CL bit
- Until each bit in the SRCSTAT register is initialized after the SRCCTRL.SRCEN bit has been rewritten from 0 to 1

On the other hand, since the CPU executes any subsequent instruction without waiting for the completion of writing to a register, the updated state of the SRCSTAT register cannot be correctly read out by an instruction immediately after the writing instruction to the SRCCTRL register. To check the updated state of the SRCSTAT register, perform a dummy read of the SRCCTRL or SRCSTAT register after the writing instruction to the SRCCTRL register.

48.5.3 Notes on Flush Processing

When 1 is written to the SRCCTRL.FL bit, this module adds dummy data of 0000 0000h after the data that have been input and continues flush processing. Perform flush processing when the last audio data has been input and there is no subsequent data.
To perform conversion again after flush processing, clear the internal work memory in either of the following ways.

- Write 1 to the SRCCTRL.CL bit.
- Write 0 and then 1 to the SRCCTRL.SRCEN bit.

48.5.4 Notes on DMA/DTC Transfer

When the DMAC or DTC is used for data transfer to the SRCID register or from the SRCOD register, do not clear the SRCSTAT.IINT or OINT flag by the CPU (writing 0 after reading 1) during the DMA/DTC transfer.

48.5.5 Notes on SRC Operation

Do not access the filter coefficient table when the SRC is operating (SRCCTRL.SRCEN = 1).

48.5.6 Conversion immediately after the Data Set in the SRCID Register

Confirm that the value of the SRCSTAT.IFDN[3:0] bits matches with the number of data written to the SRCID register when performing conversion immediately after setting the data in the SRCID register.

48.5.7 Module Stop Function Setting

Operation of the SRC can be disabled or enabled by module stop control register D (MSTPCRD). After a reset, operation of the SRC is stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

49. SD Host Interface (SDHI)

This MCU incorporates an SD host interface (SDHI) which is compliant with the SD Specifications. When developing host devices that are compliant with the SD Specifications, the user must enter into the SD Host/Ancillary Product License Agreement (SD HALA).

49.1 Overview

Table 49.1 lists the SDHI specifications.

Table 49.1 SDHI Specifications

Item	Description
SD bus interface	• Compatible with SD memory card and SDIO card (**NOT** compatible with the SPI bus interface, embedded SDIO shared bus, 8-bit SD bus, or SDIO suspend/resume functions) • Transfer bus mode selectable from 4-bit wide bus mode or 1-bit default bus mode • Compatible with SD, SDHC, and SDXC formats
Transfer modes	Selectable from high-speed mode or default speed mode
SDHI clock	The SDHI clock is generated by dividing peripheral module clock B (PCLKB) by *n*, where *n* = 2, 4, 8, 16, 32, 64, 128, 256, or 512
Error check functions	• CRC7 (command/response) • CRC16 (transfer data)
Interrupt sources	• Card access interrupt (CACI) • SDIO access interrupt (SDACI) • Card detection interrupt (CDETI) • SD buffer access interrupt (SBFAI)
DMA transfer sources	• DMAC and DTC triggerable by the SBFAI interrupt • SD buffer is read and write accessible using the DMAC and DTC
Other functions	• Card detection • Write protection

Figure 49.1 shows a block diagram of the SDHI.

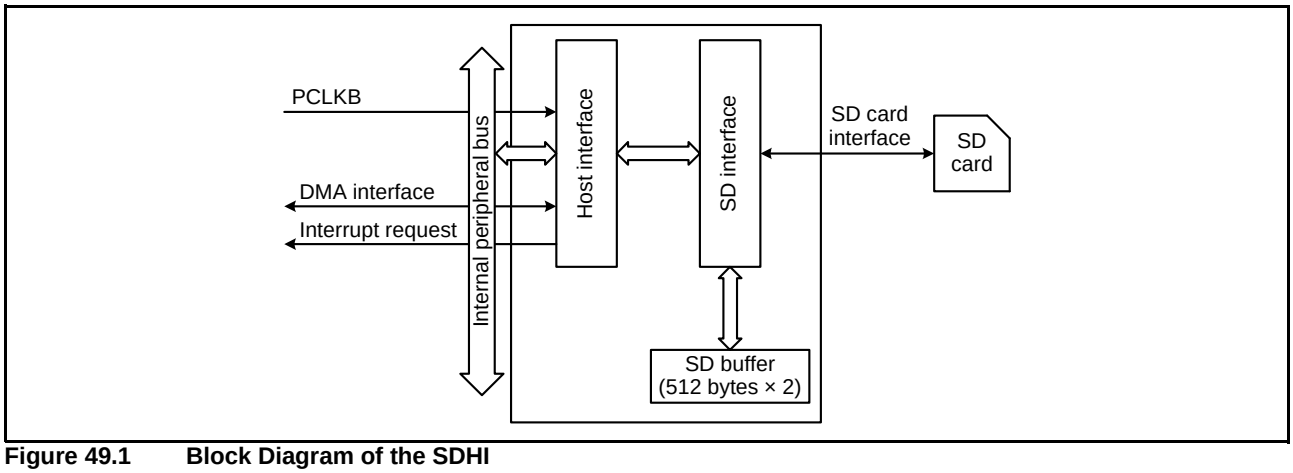


Figure 49.1 Block Diagram of the SDHI

Table 49.2 lists the pin configuration of the SDHI.

Table 49.2 Pin Configuration of the SDHI

Pin Name	I/O	Description
SDHI_CLK	Output	SDHI clock
SDHI_CMD	I/O	Command output, response input
SDHI_D0	I/O	Data 0 (DAT0)
SDHI_D1	I/O	Data 1 (DAT1), SDIO access interrupt
SDHI_D2	I/O	Data 2 (DAT2), read wait
SDHI_D3	I/O	Data 3 (DAT3), SD card detection
SDHI_CD	Input	SD card detection
SDHI_WP	Input	SD card write protection

49.2 Register Details

49.2.1 Command Register (SDCMD)

Address(es): SDHI.SDCMD 0008 AC00h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMD12AT[1:0]		TRSTP	CMDRW	CMDTP	RSPTP[2:0]			ACMD[1:0]		CMDIDX[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CMDIDX[5:0]	Command Index Field Value Select	These bits configure the command index field value. The examples below include the bit values for the ACMD[1:0] bits. b7 b0 0 0 0 0 0 1 1 0: CMD6 0 0 0 1 0 0 1 0: CMD18 0 1 0 0 1 1 0 1: ACMD13	R/W
b7, b6	ACMD[1:0]	Command Type Select	b7 b6 0 0: CMD 0 1: ACMD Only set the values listed above.	R/W
b10 to b8	RSPTP[2:0]	Response Type Select *1	b10 b8 0 0 0: Normal mode. Depending on the command, the response type and transfer method are selected by setting the ACMD[1:0] bits and CMDIDX[5:0] bits. At this time, the values for b15 to b11 in this register are invalid. 0 1 1: Expansion mode and no response 1 0 0: Expansion mode and R1, R5, R6, or R7 response 1 0 1: Expansion mode and R1b response 1 1 0: Expansion mode and R2 response 1 1 1: Expansion mode and R3 or R4 response Only set the values listed above.	R/W
b11	CMDTP	Data Transfer Select *2	0: Command does not include data transfer (bc, bcr, or ac) 1: Command includes data transfer (adtc)	R/W
b12	CMDRW	Data Transfer Direction Select *3	0: Write data to the SD card 1: Read data from the SD card	R/W
b13	TRSTP	Block Transfer Select *3	0: Single block transferred 1: Multiple blocks transferred	R/W
b15, b14	CMD12AT[1:0]	CMD12 Automatic Issue Select *4	b15 b14 0 0: CMD12 is automatically issued during multi-block transfer 0 1: CMD12 is not automatically issued during multi-block transfer Only set the values listed above.	R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Some commands cannot be used in normal mode. Refer to Table 49.3 and set the RSPTP[2:0] bits.
Note 2. The CMDTP bit is valid only when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b.
Note 3. Bits CMDRW and TRSTP are valid only when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b, and the CMDTP bit is 1.
Note 4. The CMD12AT[1:0] bits are valid only when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b, and the TRSTP bit is 1.

The command type and response type are set in the SDCMD register. The command type and transfer mode must be set when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b. The sequence starts when a value is written to this register. Refer to Table 49.3 for setting examples. Do not write to the SDCMD register when the SDSTS2.CBSY flag is 1.

Table 49.3 lists examples of SDCMD register settings.

Table 49.3 Examples of SDCMD Register Settings

Type	Command Symbol	SDCMD Register Setting	Remarks
CMD	CMD0	0000 0000h	
	CMD2	0000 0002h	
	CMD3	0000 0003h	
	CMD4	0000 0004h	
	CMD5	0000 0705h or 0000 0005h	
	CMD6	0000 1C06h or 0000 0006h	
	CMD7	0000 0007h	When the card is deselected, the SD card does not return a response, so the SDSTS2.RSPTO flag becomes 1.
	CMD8	0000 0408h or 0000 0008h	
	CMD9	0000 0009h	
	CMD10	0000 000Ah	
	CMD11	0000 040Bh or 0000 000Bh	
	CMD12	0000 000Ch	
	CMD13	0000 000Dh	
	CMD15	0000 000Fh	
	CMD16	0000 0010h	
	CMD17	0000 0011h	
	CMD18	0000 0012h	
	CMD20	0000 0514h or 0000 0014h	
	CMD24	0000 0018h	
	CMD25	0000 0019h	
	CMD27	0000 001Bh	
	CMD28	0000 001Ch	
	CMD29	0000 001Dh	
	CMD30	0000 001Eh	
	CMD32	0000 0020h	
	CMD33	0000 0021h	
	CMD38	0000 0026h	
	CMD42	0000 002Ah	
	CMD52	0000 0434h or 0000 0034h	
	CMD53	0000 1C35h	Single block read
		0000 0C35h	Single block write
		0000 7C35h	Multi-block read
		0000 6C35h	Multi-block write
		0000 0035h	The setting to the left can be used regardless of the transfer being single block or multi-block. However, the MSB in the SDARG register (RW flag) must be set to 0 when reading and 1 when writing.
	CMD55	0000 0037h	
	CMD56	0000 0038h	
ACMD	ACMD6	0000 0046h	
	ACMD13	0000 004Dh	
	ACMD22	0000 0056h	
	ACMD23	0000 0057h	
	ACMD41	0000 0069h	
	ACMD42	0000 006Ah	
	ACMD51	0000 0073h	

49.2.2 Argument Register (SDARG)

Address(es): SDHI.SDARG 0008 AC08h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The SDARG register is used for setting the argument field value. Set the SDARG register before setting the SDCMD register. The argument field value of the automatically issued CMD12 is 0000 0000h regardless of the SDARG register value.

49.2.3 Data Stop Register (SDSTOP)

Address(es): SDSTOP 0008 AC10h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	—	SDBLKCNTEN
Value after reset:	0	0	0	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	STP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STP	Transfer Stop	Data transfer stops when this bit is set to 1.	R/W
b7 to b1	—	Reserved	These bits are 0 when read and cannot be modified.	R
b8	SDBLKCNTEN	Block Count Register Value Select *1	0: SDBLKCNT register value is invalid 1: SDBLKCNT register value is valid	R/W
b31 to b9	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite this bit when the SDSTS2.CBSY flag is 1.

The SDSTOP register stops data transfer. During a multi-block transfer sequence, the SDBLKCNT register value (number of blocks to be transferred) can be set to valid or invalid by setting the SDSTOP register.

STP Bit (Transfer Stop)

When setting the STP bit to 1, set it after the SDSTS1.RSPEND flag becomes 1; when setting the STP bit to 0, set it after the SDSTS1.ACEND flag becomes 1. After a command sequence is complete, the SDHI does not issue CMD12 and the SDSTS1.ACEND flag does not become 1 even if the STP bit is set to 1. When the SDHI is in the busy state after receiving the R1b response, the SDHI does not issue CMD12 even if the STP bit is 1, and after the SDHI is released from

the busy state, the SDSTS1.ACEND flag becomes 1.

- Performing a multi-block transfer
  When the STP bit is set to 1, the SDHI issues CMD12, and the command sequence is stopped. The SD buffer can be accessed even after the STP bit is set to 1, but a buffer access error occurs and the SDSTS2.ILW flag or SDSTS2.ILR flag becomes 1. If the command sequence stops due to a communication error or a timeout, the SDHI does not issue CMD12.
- Performing a single block transfer
  When the STP bit is set to 1 during a single block write access, if there is no data in the SD buffer, the SDSTS1.ACEND flag becomes 1. If there is data in the SD buffer, after the SDHI is released from the busy state, the SDSTS1.ACEND flag becomes 1. When the STP bit is set to 1 during a single block read access, the SDSTS1.ACEND flag becomes 1. Also, CMD12 is not issued even if the STP bit is set to 1 during a single block read access or single block write access.

SDBLKCNTEN Bit (Block Count Register Value Select)

If the SDBLKCNTEN bit is 1 during a multi-block transfer sequence, the SDHI automatically issues CMD12. When the SDCMD.RSPTP[2:0] bits are set to 000b and CMD18 or CMD25 is issued, or if the SDCMD.RSPTP[2:0] bits are set to 011b, 100b, 101b, 110b, or 111b and the SDCMD.TRSTP bit is 1 (multiple blocks transferred), if the SDCMD.CMD12AT[1:0] bits are 00b (CMD12 is automatically issued during multi-block transfer), and the number of transfer blocks reaches the value set in the SDBLKCNT register, the SDHI automatically issues CMD12.
If the command sequence is stopped by a communication error or a timeout, CMD12 is not automatically issued.

49.2.4 Block Count Register (SDBLKCNT)

Address(es): SDHI.SDBLKCNT 0008 AC14h

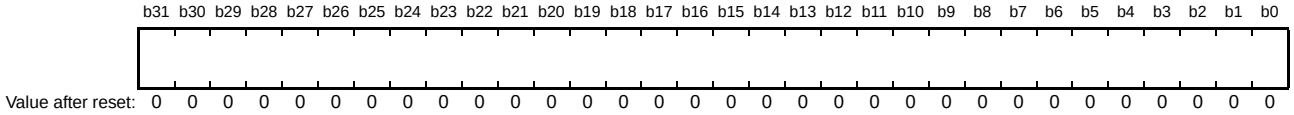
	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

When performing a multi-block transfer, SDBLKCNT is a readable/writable register used to set the number of blocks to be transferred. For example, when the register value is 0000 0001h, 1 block is transferred; when the register value is 0000 FFFFh, 65,535 blocks are transferred; and when the register value is FFFF FFFFh, 4,294,967,295 blocks are transferred. Do not set this register to 0000 0000h. Do not rewrite the SDBLKCNT register when the SDSTS2.CBSY flag is 1.

49.2.5 Response Register 10 (SDRSP10), Response Register 32 (SDRSP32), Response Register 54 (SDRSP54), Response Register 76 (SDRSP76)

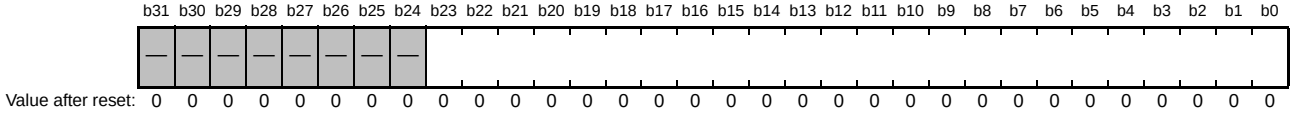
- SDRSP10, SDRSP32, SDRSP54

Address(es): SDHI.SDRSP10 0008 AC18h, SDHI.SDRSP32 0008 AC20h, SDHI.SDRSP54 0008 AC28h



- SDRSP76

Address(es): SDHI.SDRSP76 0008 AC30h



Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	—	This register stores the response from the SD card.	R
b31 to b24	—	Reserved	These bits are 0 when read.	R

Registers SDRSP10, SDRSP32, SDRSP54, and SDRSP76 are read-only registers that store the response from the SD card. Depending on the type of response from the SD card, the SDHI divides and stores the response among the four registers.

Table 49.4 lists the correspondence between the response type and its storage destination.

Table 49.4 Correspondence Between the Response Type and Its Storage Destination

Response Type	SDRSP10 Register	SDRSP32 Register	SDRSP54 Register	SDRSP76 Register
R1	[39:8]	—	[39:8] *1	—
R1b	[39:8]	—	[39:8] *1	—
R2	[39:8]	[71:40]	[103:72]	[127:104]
R3	[39:8]	—	—	—
R4	[39:8]	—	—	—
R5	[39:8]	—	—	—
R6	[39:8]	—	—	—
R7	[39:8]	—	—	—

Note 1. The response for CMD18 and CMD25 is stored in registers SDRSP10 and SDRSP54. Therefore, even if the SDRSP10 register is overwritten with the response for the automatically transmitted CMD12, the response for CMD18 or CMD25 can be confirmed by reading the SDRSP54 register.

49.2.6 SD Status Register 1 (SDSTS1)

Address(es): SDHI.SDSTS1 0008 AC38h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	SDD3MON	SDD3IN	SDD3RM
Value after reset:	0	0	0	0	0	x	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	SDWPMON	—	SDCDMON	SDCDIN	SDCDRM	ACEND	—	RSPEND
Value after reset:	x	0	x	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RSPEND	Response End Detection Flag	0: Response end is not detected 1: Response end is detected	R(/W) *1
b1	—	Reserved	This bit is 0 when read and cannot be modified.	R
b2	ACEND	Access End Detection Flag	0: Access end is not detected 1: Access end is detected	R(/W) *1
b3	SDCDRM	SDHI_CD Removal Flag	0: SD card removal not detected by the SDHI_CD pin 1: SD card removal detected by the SDHI_CD pin	R(/W) *1
b4	SDCDIN	SDHI_CD Insertion Flag	0: SD card insertion not detected by the SDHI_CD pin 1: SD card insertion detected by the SDHI_CD pin	R(/W) *1
b5	SDCDMON	SDHI_CD Pin Monitor Flag	0: SDHI_CD pin level is high *2 1: SDHI_CD pin level is low *2	R
b6	—	Reserved	This bit is 0 when read and cannot be modified.	R
b7	SDWPMON	SDHI_WP Pin Monitor Flag	0: SDHI_WP pin level is high 1: SDHI_WP pin level is low	R
b8	SDD3RM	SDHI_D3 Removal Flag	0: SD card removal not detected by the SDHI_D3 pin 1: SD card removal detected by the SDHI_D3 pin	R(/W) *1
b9	SDD3IN	SDHI_D3 Insertion Flag	0: SD card insertion not detected by the SDHI_D3 pin 1: SD card insertion detected by the SDHI_D3 pin	R(/W) *1
b10	SDD3MON	SDHI_D3 Pin Monitor Flag	0: SDHI_D3 pin level is low 1: SDHI_D3 pin level is high	R
b31 to b11	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. The flag does not change even if set to 1. Writing 0 changes the flag value to 0.
Note 2. The flag changes when the pin level continues for the period set in the SDOPT.CTOP[3:0] bits or longer.

The SDSTS1 register indicates the detection of a response end or access end for a command sequence. The SDSTS1 register also indicates the detection SD card insertion/removal, and indicates the write protection status.
During a multi-block transfer sequence, if CMD12 or CMD52 (SDIO abort) is issued, the ACEND flag becomes 1, but the RSPEND flag remains set to 0.
If the command sequence is stopped due to a communication error or timeout, the ACEND flag or RSPEND flag becomes 1.
After a reset is canceled, the SDD3MON bit, SDD3IN flag, and SDD3RM flag values are changed according to the status of the SDHI_D3 pin, and their values are changed when data is being transferred in wide bus mode.
Flags to be cleared should be set to 0; flags not being cleared should be set to 1.

RSPEND Flag (Response End Detection Flag)

— This flag becomes 1 under any of the following conditions:

- A response is received.
- A command that does not have a response is issued.
- After the R1b response is received, the SDHI is released from the busy state.
- During a multi-block transmission, after the SDIOMD.C52PUB bit is set to 1, the CMD52 response is received.
- A communication error or timeout causes the command sequence to abort.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

Note: When a command is issued that is absent of data transfer, the RSPEND flag becomes 1 after the command sequence ends.

ACEND Flag (Access End Detection Flag)

— This flag becomes 1 under any of the following conditions:

- During a single block read sequence, the SD buffer read access is completed.
- During a multi-block read sequence, the last block is read from the SD buffer.
- During a multi-block read sequence, if CMD12 is automatically issued, data is read from the SD buffer, and the response for CMD12 is received.
- During a single block write sequence, after a CRC status token is received, the SDHI is released from the busy state.
- During a multi-block write sequence, after a CRC status token is received for the last block, the SDHI is released from the busy state.
- During a multi-block write sequence, when CMD12 is automatically issued, a response busy of the automatically issued CMD12 is received.
- During a multi-block read sequence, when CMD12 is automatically issued, after setting the SDSTOP.STP bit to 1, a response of the automatically issued CMD12 is received.
- During a multi-block write sequence, when CMD12 is automatically issued, after setting the SDSTOP.STP bit to 1, a response busy of the automatically issued CMD12 is received.
- During a multi-block read sequence, after the SDIOMD.IOABT bit is set to 1, the response for CMD52 is received.
- During a multi-block write sequence, after the SDIOMD.IOABT bit is set to 1, the response for CMD52 is received.
- A communication error or timeout causes the command sequence to abort.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

Note: The ACEND flag becomes 1 after the command sequence ends.

SDCDRM Flag (SDHI_CD Removal Flag)

— This flag becomes 1 under the following condition:

- The SDHI_CD pin changes from low to high, and the high period is the period set in the SDOPT.CTOP[3:0] bits or longer.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

SDCDIN Flag (SDHI_CD Insertion Flag)

— This flag becomes 1 under the following condition:

- The SDHI_CD pin changes from high to low, and the low period is the period specified in the SDOPT.CTOP[3:0] bits or longer.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

SDD3RM Flag (SDHI_D3 Removal Flag)

— This flag becomes 1 under the following condition:

- The SDHI_D3 pin changes from high to low, and the low period is at least two PCLKB cycles.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

SDD3IN Flag (SDHI_D3 Insertion Flag)

— This flag becomes 1 under the following condition:

- The SDHI_D3 pin changes from low to high, and the high period is at least two PCLKB cycles.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

49.2.7 SD Status Register 2 (SDSTS2)

Address(es): SDHI.SDSTS2 0008 AC3Ch

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	ILA	CBSY	SDCLKCREN	—	—	—	BWE	BRE
Value after reset:	0	0	1	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	SDD0MON	RSPTO	ILR	ILW	DTO	ENDE	CRCE	CMDE
Value after reset:	x	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMDE	Command Error Detection Flag	0: Command error not detected 1: Command error detected	R(/W) *1
b1	CRCE	CRC Error Detection Flag	0: CRC error not detected 1: CRC error detected	R(/W) *1
b2	ENDE	End Bit Error Detection Flag	0: End bit error not detected 1: End bit error detected	R(/W) *1
b3	DTO	Data Timeout Detection Flag	0: Data timeout not detected 1: Data timeout detected	R(/W) *1
b4	ILW	SDBUFR Illegal Write Access Detection Flag	0: Illegal write access to the SDBUFR register not detected 1: Illegal write access to the SDBUFR register detected	R(/W) *1
b5	ILR	SDBUFR Illegal Read Access Detection Flag	0: Illegal read access to the SDBUFR register not detected 1: Illegal read access to the SDBUFR register detected	R(/W) *1
b6	RSPTO	Response Timeout Detection Flag	0: Response timeout not detected 1: Response timeout detected	R(/W) *1
b7	SDD0MON	SDHI_D0 Pin Status Flag	0: SDHI_D0 pin is low 1: SDHI_D0 pin is high	R
b8	BRE	SDBUFR Read Enable Flag	0: Read access to the SDBUFR register disabled 1: Read access to the SDBUFR register enabled	R(/W) *1
b9	BWE	SDBUFR Write Enable Flag	0: Write access to the SDBUFR register disabled 1: Write access to the SDBUFR register enabled	R(/W) *1
b10	—	Reserved	This bit is 0 when read and cannot be modified.	R
b11	—	Reserved	This bit is 0 when read. Set it to 1 when writing.	R/W
b12	—	Reserved	This bit is 0 when read and cannot be modified.	R
b13	SDCLKCREN	SDCLKCR Write Enable Flag	0: SD bus (CMD and DAT lines) is busy, so write access to the SDCLKCR.CLKEN bit and CLKSEL[7:0] bits is disabled. 1: SD bus (CMD and DAT lines) is not busy, so write access to the SDCLKCR.CLKEN bit and CLKSEL[7:0] bits is enabled.	R
b14	CBSY	Command Sequence Status Flag	0: Command sequence completed 1: Command sequence in progress (busy)	R
b15	ILA	Illegal Access Error Detection Flag	0: Illegal access error not detected 1: Illegal access error detected	R(/W) *1
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. The flag does not change even if set to 1. Writing 0 changes the flag value to 0.

The SDSTS2 register indicates the status of the SD buffer and the status of the SD card. Flags to be cleared should be set to 0; flags not being cleared should be set to 1.

CMDE Flag (Command Error Detection Flag)

The command sequence is stopped when a command error occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 49.3.6.8 or section 49.3.6.9 and complete the command sequence.

— This flag becomes 1 under any of the following conditions:

- The command index field value for the command transmitted differs from the command index field value for the response received.
- The command index field value for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer) differs from the command index field value for the response received.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

CRCE Flag (CRC Error Detection Flag)

The command sequence is stopped when a CRC error occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 49.3.6.8 or section 49.3.6.9 and complete the command sequence.

— This flag becomes 1 under any of the following conditions:

- The received CRC status token is in error (the value of the CRC status is a value other than 010b).
- The read data contains a CRC error.
- The response contains a CRC error.
- The response for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer) contains a CRC error.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

ENDE Flag (End Bit Error Detection Flag)

The command sequence is stopped when an end bit error occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 49.3.6.8 or section 49.3.6.9 and complete the command sequence.

— This flag becomes 1 under any of the following conditions:

- The response length is in error (the end bit could not be detected).
- The read data length is in error (the end bit of the enabled bit could not be detected).
- The CRC status token length is in error (the end bit could not be detected).
- The response length for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer) contains an error (the end bit could not be detected).

— This flag becomes 0 under the following condition:

- The flag is set to 0.

DTO Flag (Data Timeout Detection Flag)

This flag indicates that the data expected to be received during the period specified (set in the SDOPT.TOP[3:0] bits) was not received. However, response timeouts are excluded. The command sequence stops when a data timeout occurs.

— This flag becomes 1 under any of the following conditions:

- After the R1b response is received, the SDHI is busy for the period specified or longer.
- After the CRC status token is received, the SDHI is busy for the period specified or longer.
- After data is written, the CRC status token is not received even after the period specified elapsed.
- After a read command is issued, the read data is not received even after the period specified elapsed.
- After CMD12 is issued during a command sequence, the SDHI is busy for the period specified or longer.
- After the read data is received, the next read data is not received even after the period specified elapsed.
- After the SDHI exits the read wait state, the next read data is not received even after the period specified elapsed.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

ILW Flag (SDBUFR Illegal Write Access Detection Flag)

— This flag becomes 1 under any of the following conditions:

- A value is written to the SDBUFR register while the SDHI is not in the data read or data write command state.
- A value is written to the SDBUFR register while the SD buffer is full.
- A value is written to the SDBUFR register while the CRC status token or CRC status token length is in error.
- After the CRC status token is received, a value is written to the SDBUFR register if the SDHI is busy for the period set in bits SDOPT.TOP[3:0] or longer.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

ILR Flag (SDBUFR Illegal Read Access Detection Flag)

— This flag becomes 1 under any of the following conditions:

- A value is set to the SDBUFR register while the SD buffer is empty.
- The value read from the SDBUFR register includes a CRC error or end error.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

RSPTO Flag (Response Timeout Detection Flag)

The command sequence is stopped when a response timeout occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 49.3.6.8 or section 49.3.6.9 and complete the command sequence.

— This flag becomes 1 under the following condition:

- A response is not received even after 640 SDHI clock cycles or more have elapsed (including the response for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer)).

— This flag becomes 0 under the following condition:

- The flag is set to 0.

SDD0MON Flag (SDHI_D0 Pin Status Flag)

This flag indicates the status of the SDHI_D0 pin. After an erase command is issued, if the DTO flag is 1 and the RSPTO flag is 0, polling can be used to monitor the SDD0MON flag change from 0 to 1, and check that the erase command sequence is complete. If a communication error or timeout occurs during the write sequence, the SDHI_D0 pin may remain low. When the SDHI clock is stopped, the value before the SDHI clock was stopped is retained.

BRE Flag (SDBUFR Read Enable Flag)

— This flag becomes 1 under any of the following conditions:

- During a single block transfer, the data size set in the SDSIZE.LEN[9:0] bits is stored in the SD buffer.
- During a multi-block transfer, the data size set in the SDSIZE.LEN[9:0] bits is stored in one of the two SD buffers.

— This flag becomes 0 under any of the following conditions:

- The bit is set to 0.
- DMA transfer is used to read 1 block of data from the SD buffer.

If the CPU is used to read data from the SDBUFR register, set the BRE flag to 0 before reading the data size *1 set in the SDSIZE.LEN[9:0] bits. Even if the block of data read contains a CRC error or end bit error, the data is stored in the SD buffer and the BRE flag becomes 1.

Note 1. If the transfer data size set in the SDSIZE.LEN[9:0] bits is an odd number, the odd numbered byte is ignored. Refer to section 49.5.2, SDBUFR Register Illegal Write Error for details.

BWE Flag (SDBUFR Write Enable Flag)

— This flag becomes 1 under any of the following conditions:

- During a single block transfer, the SD buffer is empty.
- During a multi-block transfer, bank 1 or bank 2 of the SD buffer is empty.

— This flag becomes 0 under any of the following conditions:

- The flag is set to 0.
- A DMA transfer is used to write 1 block of data from the SD buffer.

If the CPU is used to write data to the SDBUFR register, set the BWE flag to 0 before writing the data size *1 set in the SDSIZE.LEN[9:0] bits.

Note 1. If the transfer data size set in the SDSIZE.LEN[9:0] bits is an odd number, the odd numbered byte is ignored. Refer to section 49.5.2, SDBUFR Register Illegal Write Error for details.

SDCLKCREN Flag (SDCLKCR Write Enable Flag)

When a value is written to the SDCMD register, the SDHI starts the command sequence, the SDSTS2.CBSY flag becomes 1, and the SDSTS2.SDCLKCREN flag becomes 0. When the command sequence is complete, after the SDSTS2.CBSY flag becomes 0, eight cycles of the SDHI clock elapse and then the SDSTS2.SDCLKCREN flag becomes 1.

ILA Flag (Illegal Access Error Detection Flag)

— This flag becomes 1 under any of the following conditions:

- A value is written to the SDCMD register when the SDSTS2.CBSY flag is 1.
- The SDCMD.CMDTP bit is set to 1 (command accompanying data transfer), the SDCMD.ACMD[1:0] bits are set to 00b, and the SDCMD.CMDIDX[5:0] bits are set to 001100b (CMD12).

— This flag becomes 0 under the following condition:

- The flag is set to 0.

49.2.8 SD Interrupt Mask Register 1 (SDIMSK1)

Address(es): SDHI.SDIMSK1 0008 AC40h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	SDD3INM	SDD3RMM
Value after reset:	0	0	0	0	0	0	1	1

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SDCDINM	SDCDRMM	ACENDM	—	RSPENDM
Value after reset:	0	0	0	1	1	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RSPENDM	Response End Interrupt Request Mask	0: Response end interrupt request is not masked 1: Response end interrupt request is masked	R/W
b1	—	Reserved	This bit is 0 when read and cannot be modified.	R
b2	ACENDM	Access End Interrupt Request Mask	0: Access end interrupt request is not masked 1: Access end interrupt request is masked	R/W
b3	SDCDRMM	SDHI_CD Removal Interrupt Request Mask	0: SD card removal interrupt request by the SDHI_CD pin not masked 1: SD card removal interrupt request by the SDHI_CD pin masked	R/W
b4	SDCDINM	SDHI_CD Insertion Interrupt Request Mask	0: SD card insertion interrupt request by the SDHI_CD pin not masked 1: SD card insertion interrupt request by the SDHI_CD pin masked	R/W
b7 to b5	—	Reserved	These bits are 0 when read and cannot be modified.	R
b8	SDD3RMM	SDHI_D3 Removal Interrupt Request Mask	0: SD card removal interrupt request by the SDHI_D3 pin not masked 1: SD card removal interrupt request by the SDHI_D3 pin masked	R/W
b9	SDD3INM	SDHI_D3 Insertion Interrupt Request Mask	0: SD card insertion interrupt request by the SDHI_D3 pin not masked 1: SD card insertion interrupt request by the SDHI_D3 pin masked	R/W
b31 to b10	—	Reserved	These bits are 0 when read and cannot be modified.	R

The SDIMSK1 register enables and disables the interrupt requests from the status flags in the SDSTS1 register. Refer to Table 49.8, Interrupt Sources for details on the relationship between the status flags and the requested interrupt source.

49.2.9 SD Interrupt Mask Register 2 (SDIMSK2)

Address(es): SDHI.SDIMSK2 0008 AC44h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	ILAM	—	—	—	—	—	BWEM	BREM
Value after reset:	1	0	0	0	1	0	1	1

	b7	b6	b5	b4	b3	b2	b1	b0
	—	RSPTOM	ILRM	ILWM	DTTOM	ENDEM	CRCEM	CMDEM
Value after reset:	0	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	CMDEM	Command Error Interrupt Request Mask	0: Command error interrupt request not masked 1: Command error interrupt request masked	R/W
b1	CRCEM	CRC Error Interrupt Request Mask	0: CRC error interrupt request not masked 1: CRC error interrupt request masked	R/W
b2	ENDEM	End Bit Error Interrupt Request Mask	0: End bit detection error interrupt request not masked 1: End bit detection error interrupt request masked	R/W
b3	DTTOM	Data Timeout Interrupt Request Mask	0: Data timeout interrupt request not masked 1: Data timeout interrupt request masked	R/W
b4	ILWM	SDBUFR Register Illegal Write Interrupt Request Mask	0: Illegal write detection interrupt request for the SDBUFR register not masked 1: Illegal write detection interrupt request for the SDBUFR register masked	R/W
b5	ILRM	SDBUFR Register Illegal Read Interrupt Request Mask	0: Illegal read detection interrupt request for the SDBUFR register not masked 1: Illegal read detection interrupt request for the SDBUFR register masked	R/W
b6	RSPTOM	Response Timeout Interrupt Request Mask	0: Response timeout interrupt request not masked 1: Response timeout interrupt request masked	R/W
b7	—	Reserved	This bit is 0 when read and cannot be modified.	R
b8	BREM	BRE Interrupt Request Mask	0: Read enable interrupt request for the SD buffer not masked 1: Read enable interrupt request for the SD buffer masked	R/W
b9	BWEM	BWE Interrupt Request Mask	0: Write enable interrupt request for the SDBUFR register not masked 1: Write enable interrupt request for the SDBUFR register masked	R/W
b10	—	Reserved	This bit is 0 when read and cannot be modified.	R
b11	—	Reserved	This bit is 1 when read and cannot be modified.	R
b14 to b12	—	Reserved	These bits are 0 when read and cannot be modified.	R
b15	ILAM	Illegal Access Error Interrupt Request Mask	0: Illegal access error interrupt request not masked 1: Illegal access error interrupt request masked	R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. When the SDIMSK2.BWEM bit is 0 or the SDIMSK2.BREM bit is 0, set the SDDMAEN.DMAEN bit to 0. When the SDDMAEN.DMAEN bit is 1, set the SDIMSK2.BWEM bit to 1 and the SDIMSK2.BREM bit to 1.

The SDIMSK2 register enables and disables the interrupt requests from the status flags in the SDSTS2 register. Refer to Table 49.8 for details on the relationship between the status flags and the requested interrupt source.

49.2.10 SDHI Clock Control Register (SDCLKCR)

Address(es): SDHI.SDCLKCR 0008 AC48h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	CLKCTRLEN	CLKEN
Value after reset:	0	0	0	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	CLKSEL[7:0]							
Value after reset:	0	0	1	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CLKSEL[7:0]	SDHI Clock Frequency Select *1	b7 b0 0 0 0 0 0 0 0 0: PCLKB divided by 2 0 0 0 0 0 0 0 1: PCLKB divided by 4 0 0 0 0 0 0 1 0: PCLKB divided by 8 0 0 0 0 0 1 0 0: PCLKB divided by 16 0 0 0 0 1 0 0 0: PCLKB divided by 32 0 0 0 1 0 0 0 0: PCLKB divided by 64 0 0 1 0 0 0 0 0: PCLKB divided by 128 0 1 0 0 0 0 0 0: PCLKB divided by 256 1 0 0 0 0 0 0 0: PCLKB divided by 512 Only set the values listed above.	R/W
b8	CLKEN	SDHI Clock Output Control *1	0: SDHI clock output is disabled (SDHI_CLK signal fixed low) 1: SDHI clock output enabled	R/W
b9	CLKCTRLEN	SDHI Clock Output Automatic Control Select	0: Automatic control of SDHI clock output disabled 1: Automatic control of SDHI clock output enabled	R/W
b31 to b10	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Bits CLKSEL[7:0] and CLKEN cannot be write accessed when the SDSTS2.SDCLKCREN flag is 0.

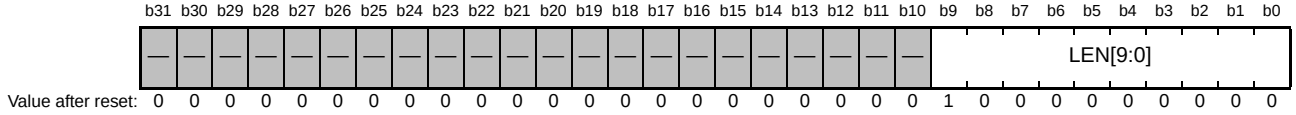
The SDCLKCR register controls the SDHI clock frequency settings and output. Set the CLKEN bit to 1 before writing to the SDCMD register to start a command sequence. Do not write access the SDCLKCR register when the SDSTS2.SDCLKCREN flag is 0.

CLKCTRLEN Bit (SDHI Clock Output Automatic Control Select)

The SDHI clock output automatic control is a function for starting and stopping SDHI clock output only during a command sequence. When this function is enabled, the SDHI starts outputting the SDHI clock after a value is set to the SDCMD register. After the command sequence is complete and eight cycles of the SDHI clock elapse, the SDHI stops outputting the SDHI clock. When the SDCLKCR.CLKEN bit is 0, output from the SDHI_CLK pin becomes low regardless of the CLKCTRLEN bit setting.

49.2.11 Transfer Data Size Register (SDSIZE)

Address(es): SDHI.SDSIZE 0008 AC4Ch



Bit	Symbol	Bit Name	Description	R/W
b9 to b0	LEN[9:0]	Transfer Data Size Setting	Set the transfer data size. *1	R/W
b11 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R
b31 to b12	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite these bits when the SDSTS2.CBSY flag is 1.

The SDSIZE register is used to set the transfer data size.

LEN[9:0] Bits (Transfer Data Size Setting)

When using single block transfer, the transfer data size can be set from 1 byte to 512 bytes. When CMD12 is automatically issued during a multi-block transfer sequence (CMD18 and CMD25), the transfer data size can only be set to 512 bytes. When CMD12 is not automatically issued during a multi-block transfer sequence, the transfer data size can be set to 32, 64, 128, 256, or 512 bytes. However, a 32-, 64-, 128-, or 256-byte multi-block read transfer can only be performed during an SDIO multi-block transfer (CMD53). Do not set these bits to 0 when using a command that includes data transfer.

49.2.12 Card Access Option Register (SDOPT)

Address SDHI.SDOPT 0008 AC50h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WIDTH	—	—	—	—	—	—	—	TOP[3:0]				CTOP[3:0]			
Value after reset	0	1	0	0	0	0	0	0	1	1	1	0	1	1	1	0

Bit	Symbol	Bit Name	Description		R/W
b3 to b0	CTOP[3:0]	Card Detection Time Counter *1	b3 b0 0 0 0 0: PCLKB × 2^{10} 0 0 0 1: PCLKB × 2^{11} 0 0 1 0: PCLKB × 2^{12} 0 0 1 1: PCLKB × 2^{13} 0 1 0 0: PCLKB × 2^{14} 0 1 0 1: PCLKB × 2^{15} 0 1 1 0: PCLKB × 2^{16} 0 1 1 1: PCLKB × 2^{17}	b3 b0 1 0 0 0: PCLKB × 2^{18} 1 0 0 1: PCLKB × 2^{19} 1 0 1 0: PCLKB × 2^{20} 1 0 1 1: PCLKB × 2^{21} 1 1 0 0: PCLKB × 2^{22} 1 1 0 1: PCLKB × 2^{23} 1 1 1 0: PCLKB × 2^{24} 1 1 1 1: Do not set this value.	R/W
b7 to b4	TOP[3:0]	Timeout Counter *1	b7 b4 0 0 0 0: SDHI clock × 2^{13} 0 0 0 1: SDHI clock × 2^{14} 0 0 1 0: SDHI clock × 2^{15} 0 0 1 1: SDHI clock × 2^{16} 0 1 0 0: SDHI clock × 2^{17} 0 1 0 1: SDHI clock × 2^{18} 0 1 1 0: SDHI clock × 2^{19} 0 1 1 1: SDHI clock × 2^{20}	b7 b4 1 0 0 0: SDHI clock × 2^{21} 1 0 0 1: SDHI clock × 2^{22} 1 0 1 0: SDHI clock × 2^{23} 1 0 1 1: SDHI clock × 2^{24} 1 1 0 0: SDHI clock × 2^{25} 1 1 0 1: SDHI clock × 2^{26} 1 1 1 0: SDHI clock × 2^{27} 1 1 1 1: Do not set this value.	R/W
b8	—	Reserved	This bit is 0 when read. Set it to 0 when writing.		R/W
b12 to b9	—	Reserved	These bits are 0 when read and cannot be modified.		R
b13	—	Reserved	This bit is 0 when read. Set it to 0 when writing.		R/W
b14	—	Reserved	This bit is 1 when read and cannot be modified.		R
b15	WIDTH	SD Bus Width Select *1	0: Wide bus mode (4 bits) 1: Default bus mode (1 bit)		R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.		R

Note 1. Do not rewrite these bits when the SDSTS2.CBSY flag is 1.

The SD bus width and timeout counter are set in the SDOPT register.

49.2.13 SD Error Status Register 1 (SDERSTS1)

Address(es): SDHI.SDERSTS1 0008 AC58h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	CRCTK[2:0]			CRCTKE	RDCRCE	RSPCRCE1	RSPCRCE0
Value after reset:	0	0	1	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CRCLENE	RDLENE	RSPLENE1	RSPLENE0	CMDE1	CMDE0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMDE0	Command Error Flag 0	0: Command index field value for a command *1 response is error free 1: Command index field value for a command *1 response is in error	R
b1	CMDE1	Command Error Flag 1	0: Command index field value for a command *2 response is error free 1: Command index field value for a command *2 response is in error (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the CMDE0 flag)	R
b2	RSPLENE0	Response Length Error Flag 0	0: Command *1 response length is error free 1: Command *1 response length is in error	R
b3	RSPLENE1	Response Length Error Flag 1	0: Command *2 response length is error free 1: Command *2 response length is in error (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the RSPLENE0 flag)	R
b4	RDLENE	Read Data Length Error Flag	0: Read data length error did not occur 1: Read data length error occurred	R
b5	CRCLENE	CRC Status Token Length Error Flag	0: CRC status token length error did not occur 1: CRC status token length error occurred	R
b7, b6	—	Reserved	These bits are 0 when read.	R
b8	RSPCRCE0	Response CRC Error Flag 0	0: No CRC error detected in command *1 response 1: CRC error detected in command *1 response	R
b9	RSPLENE1	Response CRC Error Flag 1	0: No CRC error detected in command *2 response 1: CRC error detected in command *2 response (the error that occurs for CMD12 with the setting of the SDCMD.CMDIDX[5:0] bits is indicated by the RSPCRCE0 flag)	R
b10	RDCRCE	Read Data CRC Error Flag	0: No CRC error detected in read data 1: CRC error detected in read data	R
b11	CRCTKE	CRC Status Token Error Flag	0: No error detected in CRC status token 1: Error detected in CRC status token	R
b14 to b12	CRCTK[2:0]	CRC Status Token	Store the CRC status token value (normal value is 010b)	R
b31 to b15	—	Reserved	These bits are 0 when read.	R

Note 1. Command other than CMD12 or CMD52 which are automatically issued to stop data transfer.
Note 2. CMD12 or CMD52 which are automatically issued to stop data transfer.

The SDERSTS1 register indicates the CRC status token, CRC error, end bit error, and command error.

49.2.14 SD Error Status Register 2 (SDERSTS2)

Address(es): SDHI.SDERSTS2 0008 AC5Ch

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	CRCBSYTO	CRCTO	RDTO	BSYTO1	BSYTO0	RSPTO1	RSPTO0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RSPTO0	Response Timeout Flag 0	0: After a command *1 was issued, a response was received in less than 640 cycles of the SDHI clock. 1: After a command *1 was issued, a response was not received even after 640 cycles or more of the SDHI clock elapsed.	R
b1	RSPTO1	Response Timeout Flag 1	0: After a command *2 was issued, a response was received in less than 640 cycles of the SDHI clock. 1: After a command *2 was issued, a response was not received even after 640 cycles or more of the SDHI clock elapsed (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the RSPTO0 flag).	R
b2	BSYTO0	Busy Timeout Flag 0	0: After the R1b response was received, the SDHI was released from the busy state during the specified period *3. 1: After the R1b response was received, the SDHI was in the busy state even after the specified period *3 elapsed.	R
b3	BSYTO1	Busy Timeout Flag 1	0: After CMD12 was automatically issued, the SDHI was released from the busy state during the specified period *3. 1: After CMD12 was automatically issued, the SDHI was in the busy state even after the specified period *3 elapsed (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the BSYTO0 flag).	R
b4	RDTO	Read Data Timeout Flag	After a read command is issued, this flag becomes 1 when read data is not received even after the specified period *3 elapses. After read data is received, this flag becomes 1 when the next block of read data is not received even after the specified period *3 elapses. After the SDHI exits the read wait state, this flag becomes 1 when the next block of read data is not received even after the specified period *3 elapses.	R
b5	CRCTO	CRC Status Token Timeout Flag	0: After data was written to the SD card, a CRC status token was received during the specified period *3. 1: After CRC data was written to the SD card, a CRC status token was not received even after the specified period *3 elapsed.	R
b6	CRCBSYTO	CRC Status Token Busy Timeout Flag	0: After a CRC status token was received, the SDHI was released from the busy state during the specified period *3. 1: After a CRC status token was received, the SDHI is in the busy state even after the specified period *3 elapsed.	R
b31 to b7	—	Reserved	These bits are 0 when read.	R

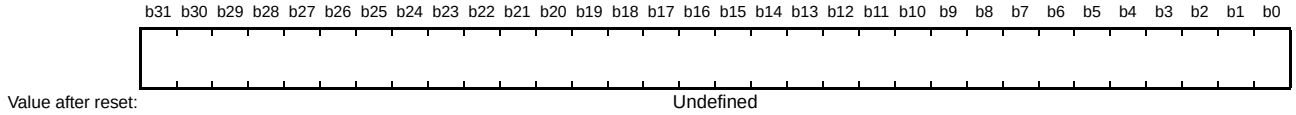
Note 1. Command other than CMD12 or CMD52 which are automatically issued to stop data transfer.
Note 2. CMD12 or CMD52 which are automatically issued to stop transfer.

Note 3. Set the SDOPT.TOP[3:0] bits to select the number of *n* cycles.

The SDERSTS2 register indicates the timeout status.

49.2.15 SD Buffer Register (SDBUFR)

Address(es): SDHI.SDBUFR 0008 AC60h



The SDBUFR register is used when writing data to the SD card and when reading data from the SD card. The SDBUFR register is connected to the SDHI's internal SD buffer. Refer to section 49.3.1, Data Block Format of the SD Card for details on the configuration of the SDBUFR register and the SD buffer.

49.2.16 SDIO Mode Control Register (SDIOMD)

Address(es): SDHI.SDIOMD 0008 AC68h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	C52PUB	IOABT	—	—	—	—	—	RWREQ	—	INTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	INTEN	SDIO Interrupt Acceptance Enable *1	0: SDIO interrupt accept disabled 1: SDIO interrupt accept enabled	R/W
b1	—	Reserved	This bit is 0 when read and cannot be modified.	R
b2	RWREQ	Read Wait Request	0: SDHI exits read wait state 1: Request for SDHI to enter read wait state	R/W
b7 to b3	—	Reserved	These bits are 0 when read and cannot be modified.	R
b8	IOABT	SDIO Abort	If this bit is set to 1 during multi-block transfer triggered by CMD53, CMD52 is immediately issued, and the command sequence is aborted.	R/W
b9	C52PUB	SDIO None Abort	If this bit is set to 1 during multi-block transfer triggered by CMD53, CMD52 is issued before the transfer process is complete, and the command sequence is completed.	R/W
b31 to b10	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite this bit when the SDSTS2.CBSY flag is 1.

The SDIOMD register controls reception of the SDIO interrupt, controls CMD52 issuance during multi-block transfer, and controls the read wait request. Do not set bits C52PUB and IOABT to 1 at the same time.

RWREQ Bit (Read Wait Request)

If the RWREQ bit is set to 1 during a multi-block read sequence triggered by issuing CMD53, when the current block is done being read, the SDHI enters the read wait state. The method for exiting the read wait state is as follows.

- If the RWREQ bit is set to 0 while the SDHI is in the read wait state, the SDHI exits the read wait state.
- If the IOABT bit is set to 1 while the SDHI is in the read wait state, after CMD52 is issued, the RWREQ bit becomes 0 and the SDHI exits the read wait state.
- If bits C52PUB and RWREQ are simultaneously set to 1 during a multi-block read sequence triggered by issuing CMD53 *1, the SDHI does not automatically exit the read wait state, so after receiving the CMD52 response, set the RWREQ bit to 0.

Note 1. Set bits RWREQ and C52PUB to 1 simultaneously.

If the RWREQ bit is set to 1 while the last block is being transferred during a multi-block read sequence triggered by issuing CMD53, the SDHI will not enter the read wait state, the SDSTS1.ACEND flag becomes 1, and the RWREQ bit becomes 0. Set the RWREQ bit to 1 after the SDSTS1.RSPEND flag becomes 1.

IOABT Bit (SDIO Abort)

- If the IOABT bit is set to 1 during a multi-block transfer sequence triggered by issuing CMD53, the SDHI stops the CMD53 command sequence, and CMD52 is issued. If the command sequence is stopped due to a communication error or timeout, the SDHI does not issue CMD52. The SD buffer can be accessed even after the IOABT bit is set to 1, but the SDSTS2.ILR flag or ILW flag becomes 1, and a buffer access error occurs. Write a value to the SDARG register before setting the IOABT bit to 1.
- During a single block write, if there is no data in the SD buffer when the IOABT bit is set to 1, the SDHI does not issue CMD52, and the SDSTS1.ACEND flag becomes 1. If there is data in the SD buffer when the IOABT bit is set to 1, the SDHI does not issue CMD52, and after the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1.
- If the IOABT bit is set to 1 during a single block read, the SDHI does not issue CMD52, and the SDSTS1.ACEND flag immediately becomes 1.
- If the SDHI is in the busy state after the R1b response is received and the IOABT bit is set to 1, the SDHI does not issue CMD52, and after the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1.
- If the IOABT bit is set to 1 after the command sequence is completed, the SDHI does not issue CMD52, and the SDSTS1.ACEND flag does not become 1.
- Set the IOABT bit to 1 after the SDSTS1.RSPEND flag becomes 1.
- Set the IOABT bit to 0 after the SDSTS1.ACEND flag becomes 1.

C52PUB Bit (SDIO None Abort)

- If the C52PUB bit is set to 1 during a multi-block write sequence triggered by issuing CMD53, CMD52 is automatically issued when the SD buffer is empty and the current block write access is complete. The C52PUB bit becomes 0 after the response for CMD52 is received. If the C52PUB bit is 1 while the last block is being transferred, the SDHI does not issue CMD52, and after the SDSTS1.RSPEND flag becomes 1, the C52PUB bit is set to 0.
- If the C52PUB bit and RWREQ bit are set to 1 during a multi-block read sequence triggered by issuing CMD53, the SDHI enters the read wait state after the current block read access is complete, and the SDHI automatically issues CMD52. The C52PUB bit becomes 0 after the response for CMD52 is received. If the C52PUB bit is set to 1 while the last block is being transferred, the SDHI does not issue CMD52, and after the SDSTS1.RSPEND flag becomes 1, the C52PUB bit is set to 0.
- During a multi-block read sequence triggered by issuing CMD53, if the C52PUB bit is set to 1, also set the RWREQ bit to 1.
- Write a value to the SDARG register before setting the C52PUB bit to 1.
- Set the C52PUB bit to 1 after the SDSTS1.RSPEND flag becomes 1.

49.2.17 SDIO Status Register (SDIOSTS)

Address(es): SDHI.SDIOSTS 0008 AC6Ch

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	EXWT	EXPUB52	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IOIRQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IOIRQ	SDIO Interrupt Status Flag	0: SDIO interrupt not accepted 1: SDIO interrupt accepted	R(/W) *1
b2, b1	—	Reserved	These bits are undefined when read. Set them to 1 when writing.	R/W
b13 to b3	—	Reserved	These bits are 0 when read and cannot be modified.	R
b14	EXPUB52	EXPUB52 Status Flag	Indicates the status of the EXPUB52	R(/W) *1
b15	EXWT	EXWT Status Flag	Indicates the status of the EXWT	R(/W) *1
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. The flag value does not change even when set to 1. If 0 is written to this flag, it becomes 0.

The SDIOSTS register indicates the status of the SDIO card access. When clearing a flag, bits to be cleared should be set to 0; bits not being cleared should be set to 1.

IOIRQ Flag (SDIO Interrupt Status Flag)

— This flag becomes 1 under the following condition:

- The SDIO interrupt from the SDIO card is accepted while the SDIOMD.INTEN bit is 1.

— This flag becomes 0 under the following condition:

- The flag is set to 0. *1

Note 1. Access the SDIO card, negate the SDIO interrupt from the SDIO card, and then set the IOIRQ flag to 0. If the SDIO interrupt from the SDIO card is not negated, the IOIRQ flag might become 1 again.

EXPUB52 Flag (EXPUB52 Status Flag)

— This flag becomes 1 under any of the following conditions:

- When multi-block transfer is triggered by CMD53 being issued, the SDIOMD.C52PUB bit is set to 1 while the last block is being transferred.
- When multi-block write is triggered by CMD53 being issued, the SDIOMD.C52PUB bit remains set to 1 while the last block is being transferred.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

EXWT Flag (EXWT Status Flag)

— This flag becomes 1 under the following condition:

- During a multi-block read sequence triggered by CMD53 being issued, the SDIOMD.RWREQ bit is set to 1 while the last block is being transferred.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

49.2.18 SDIO Interrupt Mask Register (SDIOIMSK)

Address(es): SDHI.SDIOIMSK 0008 AC70h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	EXWTM	EXPUB52M	—	—	—	—	—	—
Value after reset:	1	1	0	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IOIRQM
Value after reset:	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	IOIRQM	IOIRQ Interrupt Mask Control	0: IOIRQ interrupt not masked 1: IOIRQ interrupt masked	R/W
b2, b1	—	Reserved	These bits are 1 when read. Set them to 1 when writing.	R/W
b13 to b3	—	Reserved	These bits are 0 when read and cannot be modified.	R
b14	EXPUB52M	EXPUB52 Interrupt Request Mask Control	0: EXPUB52 interrupt request not masked 1: EXPUB52 interrupt request masked	R/W
b15	EXWTM	EXWT Interrupt Request Mask Control	0: EXWT interrupt request not masked 1: EXWT interrupt request masked	R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

The SDIOIMSK register enables and disables the interrupt requests from the status flags in the SDIOSTS register. Refer to Table 49.8, Interrupt Sources for details on the relationship between the status flags and the requested interrupt source.

49.2.19 DMA Transfer Enable Register (SDDMAEN)

Address(es): SDHI.SDDMAEN 0008 ADB0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMAEN	—
Value after reset:	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b1	DMAEN	DMA Transfer Enable *1 *2	0: Using DMAC and DTC to access the SDBUFR register is disabled 1: Using DMAC and DTC to access the SDBUFR register is enabled	R/W
b3, b2	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R
b4	—	Reserved	This bit is 1 when read. Set it to 1 when writing.	R
b5	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b7, b6	—	Reserved	These bits are 0 when read and cannot be modified.	R
b9, b8	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b11, b10	—	Reserved	These bits are 0 when read and cannot be modified.	R
b12	—	Reserved	This bit is 1 when read. Set it to 1 when writing.	R
b31 to b13	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite this bit when the SDSTS2.CBSY bit is 1.
Note 2. When the SDIMSK2.BWEM bit is 0 or the SDIMSK2.BREM bit is 0, set the SDDMAEN.DMAEN bit to 0. When the SDDMAEN.DMAEN bit is 1, set the SDIMSK2.BWEM bit to 1 and the SDIMSK2.BREM bit to 1.

The SDDMAEN register enables and disables DMA transfer.

DMAEN Bit (DMA Transfer Enable)

When using DMA transfer to access the SD buffer, set the DMAEN bit to 1 before setting the SDCMD register.

49.2.20 SDHI Software Reset Register (SDRST)

Address(es): SDHI.SDRST 0008 ADC0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SDRST	SDHI Software Reset Control	0: SDHI software reset 1: SDHI software reset canceled	R/W
b2, b1	—	Reserved	These bits are 1 when read. Set them to 1 when writing.	R
b31 to b3	—	Reserved	These bits are 0 when read and cannot be modified.	R

Table 49.5 lists the bits and flags initialized by the SDHI software reset.

Table 49.5 Bits and Flags Initialized by the SDHI Software Reset

Register	Bit/Flag
SDSTOP	SDBLKCNTEN
SDSTS1	RSPEND, ACEND
SDSTS2	CMDE, CRCE, ENDE, DTO, ILW, ILR, RSPTO, SDD0MON, BRE, BWE, SDCLKCREN, ILA
SDCLKCR	CLKEN
SDOPT	CTOP[3:0], TOP[3:0], WIDTH Bits b8 and b13 in the SDOPT register are also initialized by the SDHI software reset.
SDERSTS1	CMDE0, CMDE1, RSPLENE0, RSPLENE1, RDLENE, CRCLENE, RSPCRCE0, RSPCRCE1, RDCRCE, CRCTKE, CRCTK[2:0]
SDERSTS2	RSPTO0, RSPTO1, BSYTO0, BSYTO1, RDTO, CRCTO, CRCBSYTO
SDIOSTS	IOIRQ, EXPUB52, EXWT

49.2.21 Version Register (SDVER)

Address(es): SDHI.SDVER 0008 ADC4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CPRM	CLKRAT	—	—	IP2[3:0]				IP1[7:0]							
Value after reset:	1	0	0	0	1	0	1	0	0	0	0	0	1	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	IP1[7:0]	IP Version 1	IP version 1	R
b11 to b8	IP2[3:0]	IP Version 2	IP version 2	R
b13, b12	—	Reserved	These bits are 0 when read.	R
b14	CLKRAT	Operating Clock Condition	0: SDHI clock frequency = PCLKB frequency not supported 1: SDHI clock frequency = PCLKB frequency supported	R
b15	CPRM	CPRM Function Select	0: CPRM function is enabled 1: CPRM function is disabled	R
b31 to b16	—	Reserved	These bits are 0 when read.	R

The SDVER register indicates the SDHI version.

49.2.22 Swap Control Register (SDSWAP)

Address(es): SDHI.SDSWAP 0008 ADE0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	BRSWP	BWSWP	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is 0 when read and cannot be modified.	R
b1	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b2	—	Reserved	This bit is 0 when read and cannot be modified.	R
b4, b3	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b5	—	Reserved	This bit is 0 when read and cannot be modified.	R
b6	BWSWP	SDBUFR Swap Write *1	0: Normal write operation 1: Swap the byte endian before writing to the SDBUFR register	R/W
b7	BRSWP	SDBUFR Swap Read *1	0: Normal read operation 1: Swap the byte endian before reading the SDBUFR register	R/W
b10 to b8	—	Reserved	These bits are 0 when read and cannot be modified.	R
b12, b11	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b14, b13	—	Reserved	These bits are 0 when read and cannot be modified.	R
b15	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite this bit when the SDSTS2.CBSY flag is 1.

The SDSWAP register is used to select whether or not the byte endian is swapped when accessing the SDBUFR register. Refer to section 49.3.1 for details on the differences in accessing the SDBUFR register based on the SDSWAP register value.

49.3 SDHI Operation

49.3.1 Data Block Format of the SD Card

The SDHI has a default bus mode (1-bit width) that uses just the SDHI_D0 pin as a data line and a wide bus mode (4-bit width) that uses pins SDHI_D0 to SDHI_D3. Figure 49.2 shows the transfer format when the SDOPT.WIDTH bit is 1 (default bus mode), and Figure 49.3 shows the transfer format when the SDOPT.WIDTH bit is 0 (wide bus mode).

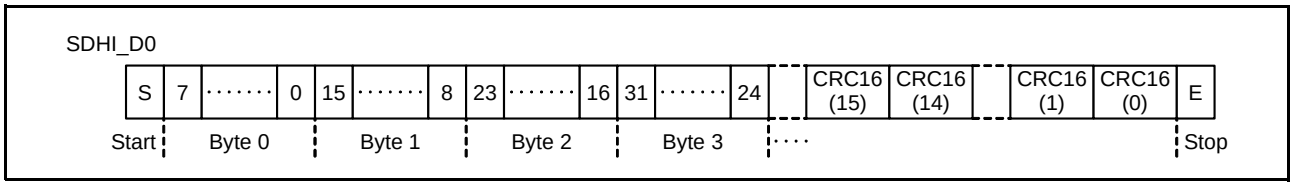


Figure 49.2 Transfer Format in Default Bus Mode

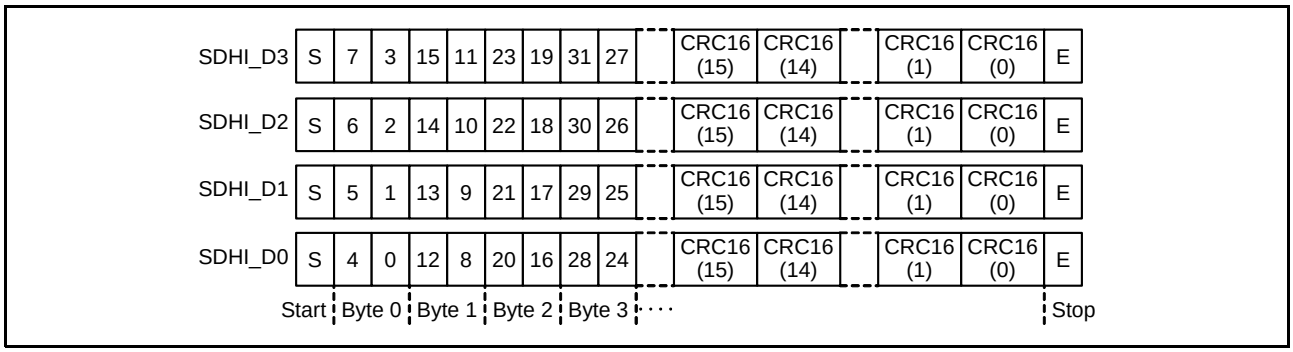


Figure 49.3 Transfer Format in Wide Bus Mode

49.3.2 SD Buffer and the SDBUFR Register

The SDHI transfers data to an SD card via its internal SD buffer. The SD buffer is comprised of a double buffer, and each buffer is 512 bytes. Figure 49.4 shows the data configuration of a single buffer of the SD buffer's double buffer.

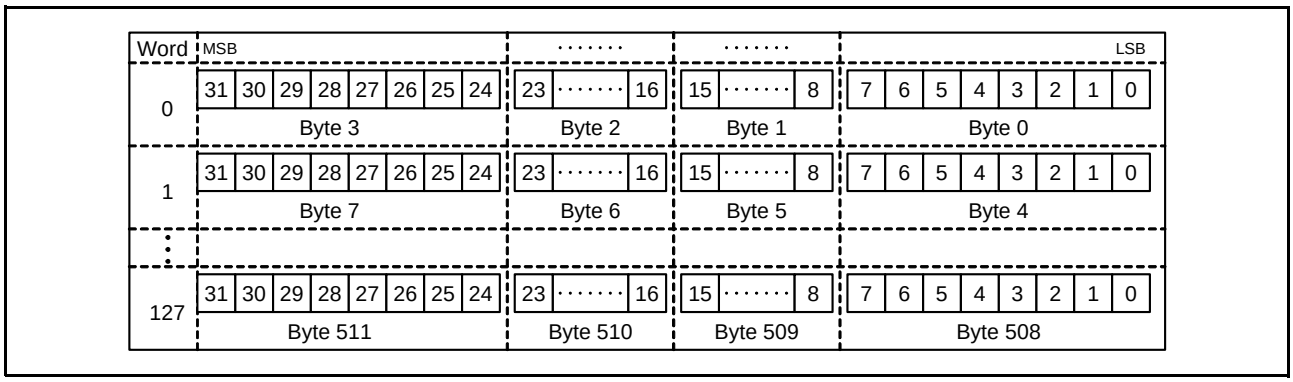


Figure 49.4 Data Configuration of Single Buffer in the SD Buffer

Access to the SD buffer is done via the SDBUFR register. If data is written to the SDBUFR register while the SDSWAP.BWSWP bit is 1, the SDHI swaps the endian for the byte, and stores the data in the SDBUFR register. If data is read from the SDBUFR register while the SDSWAP.BRSWP bit is 1, the data of the endian for the byte that was swapped can be read. Figure 49.5 shows the data alignment when reading the SDBUFR register.

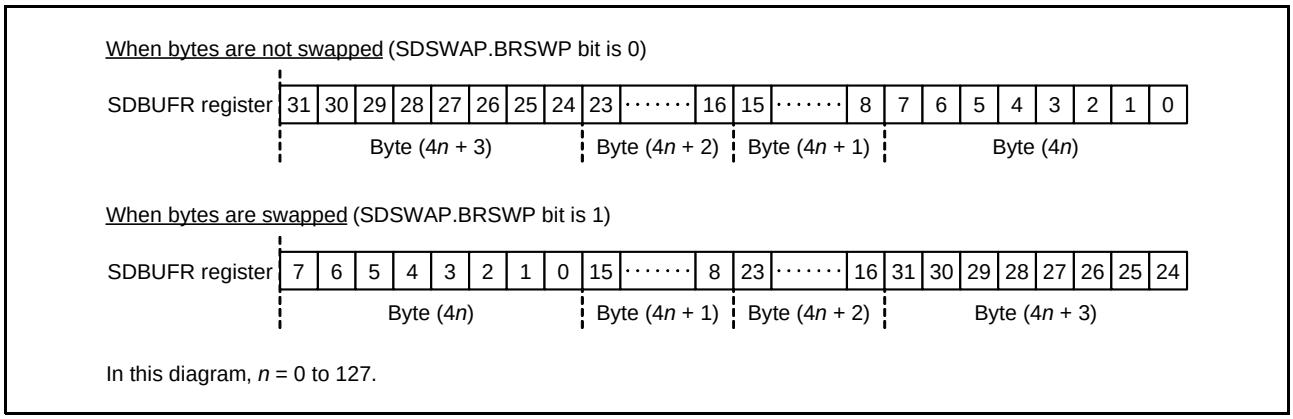


Figure 49.5 Data Alignment When Reading the SDBUFR Register

49.3.3 SD Card Detection

The SDHI can detect an SD card using either the SDHI_CD pin or SDHI_D3 pin.

49.3.3.1 Using the SDHI_CD Pin to Detect an SD Card

Figure 49.6 shows the timing chart for SD card detection using the SDHI_CD pin. The SDHI_CD pin is connected to the card detection switch of the SD card connector, and is pulled-up by the MCU. The pull-up resistance value is determined by the specifications of the host device. Note that there are some SD card sockets whose card detection switches become open when the SD card is inserted.

- Detecting SD card insertion
  The signal from the SDHI_CD pin becomes low when an SD card is inserted. This causes the SDSTS1.SDCDIN flag to become 1 if the SDHI_CD pin is low for the number of cycles set in the SDOPT.CTOP[3:0] bits. The SDSTS1.SDCDIN flag is cleared by setting it to 0.
- Detecting SD card removal
  The signal from the SDHI_CD pin becomes high when the SD card is removed. This causes the SDSTS1.SDCDRM flag to become 1 if the SDHI_CD pin is high for the number of cycles set in the SDOPT.CTOP[3:0] bits. The SDSTS1.SDCDRM flag is cleared by setting it to 0.

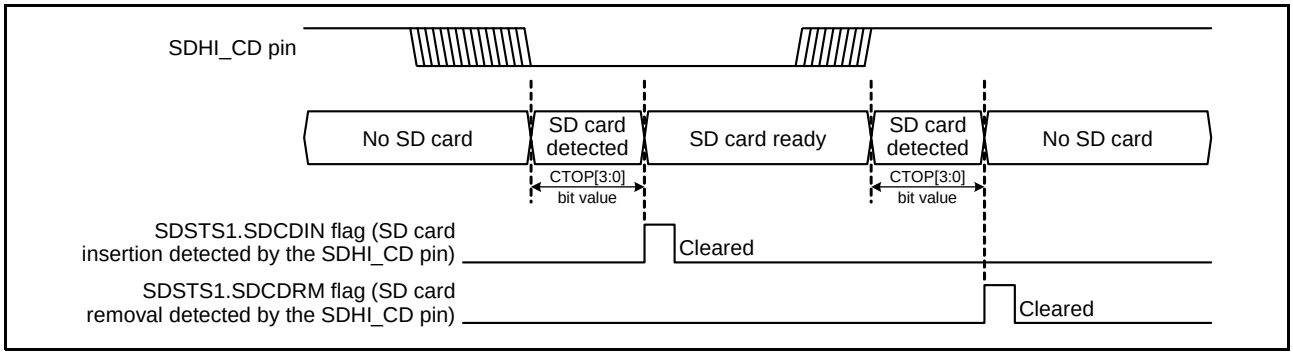


Figure 49.6 SD Card Detection Using the SDHI_CD Pin

49.3.3.2 Using the SDHI_D3 Pin to Detect an SD Card

Figure 49.7 shows the timing chart for SD card detection using the SDHI_D3 pin. The SDHI_D3 pin is pulled-down by the MCU. The pull-down resistance value is determined by the specifications of the host device.

- Detecting SD card insertion
  The signal from the SDHI_D3 pin becomes high when an SD card is inserted. This causes the SDSTS1.SDD3IN flag to become 1. The SDSTS1.SDD3IN flag is cleared by setting it to 0.
- Detecting SD card removal
  The signal from the SDHI_D3 pin becomes low when the SD card is removed. This causes the SDSTS1.SDD3RM flag to become 1. The SDSTS1.SDD3RM flag is cleared by setting it to 0.

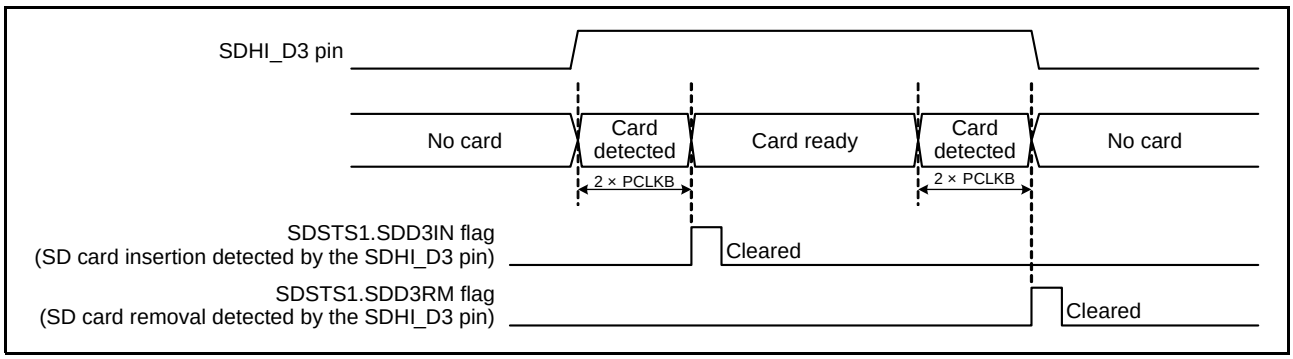


Figure 49.7 SD Card Detection Using the SDHI_D3 Pin

49.3.4 SD Card Write Protection

The SDHI can disable writing to an SD card via the SDHI_WP pin or a command.

49.3.4.1 Using the SDHI_WP Pin to Enable Write Protection

The SDHI_WP pin is connected to the WP detection switch of the SD card connector, and the SDHI_WP pin is pulled-down or pulled-up when an SD card is inserted. The resistance value and whether the SDHI_WP pin is pulled-up or pulled-down are determined by the specifications of the host device. The status of the SDHI_WP pin is reflected in the SDSTS1.SDWPMON flag. After an SD card is inserted, read the SDSTS1.SDWPMON flag to check if write protection is enabled or disabled.

49.3.4.2 Using a Command to Enable Write Protection

The SDHI uses the write protect command or the SD card lock command to disable writing to the SD card.

49.3.5 Communication Errors and Timeouts

When a communication error or timeout error occurs, depending on the type of error, the corresponding status flag in the SDSTS2 register becomes 1. Also, depending on the source of the error, the corresponding flag in the SDERSTS1 or SDERSTS2 register becomes 1.

The status flags in registers SDERSTS1 and SDERSTS2 become 0 by writing to the SDCMD register, or by setting the SDRST.SDRST bit to 0.

Table 49.6 Communication Errors

Communication Error	Interrupt Flag Register		Error Status Register		This Occurs When...
	Register symbol	Bit symbol	Register symbol	Bit symbol	
End bit error	SDSTS2	ENDE	SDERSTS1	CRCLENE	The CRC status token length is in error
				RDLENE	The read data length is in error
				RSPLENE1	The response length is in error *1
				RSPLENE0	The response length is in error *2
CRC error		CRCE		CRCTKE	The CRC status token is in error
				RDCRCE	There is a CRC error in the read data
				RSPCRCE1	There is a CRC error in the response *1
				RSPCRCE0	There is a CRC error in the response *2
Command error		CMDE		CMDE1	The command index field value for the transmitted command and received response do not match *1
				CMDE0	The command index field value for the transmitted command and received response do not match *2

Note 1. CMD12 or CMD52 which are automatically issued to stop transfer.
Note 2. A command other than CMD12 or CMD52 which are automatically issued to stop transfer.

Table 49.7 Timeouts

Timeout	Interrupt Flag Register		Error Status Register		This Occurs When...
	Register symbol	Bit symbol	Register symbol	Bit symbol	
Response timeout	SDSTS2	RSPTO	SDERSTS2	RSPTO1	A response is not received even after a minimum of 640 SDHI clock cycles elapse *1
				RSPTO0	A response is not received even after a minimum of 640 SDHI clock cycles elapse *2
Data timeout (excluding response timeout)		DTO		CRCBSYTO	After the CRC status token is received, the SDHI is busy for at least the period set *3
				CRCTO	After the write data is transmitted, the CRC status token is not received even after at least the period set *3 elapses
				RDTO	After the read command is issued, the read data is not received even after at least the period set *3 elapses
					After the read data is received, the next block read data is not received even after at least the period set *3 elapses
					After the SDHI exits the read wait state, the next block read data is not received even after at least the period set *3 elapses
				BSYTO1	After CMD12 is issued during the command sequence, the SDHI is busy for at least the period set *3
				BSYTO0	After the R1b response is received, the SDHI is busy for at least the period set *3 (a command other than CMD12 is issued during the command sequence)

Note 1. CMD12 or CMD52 which are automatically issued to stop transfer.
Note 2. A command other than CMD12 or CMD52 which are automatically issued to stop transfer.
Note 3. The period is set in the SDOPT.TOP[3:0] bits.

49.3.6 Examples of Issuing a Command

49.3.6.1 Command Absent of Response Reception and Data Transfer

Figure 49.8 shows an example of no response being received and no data being transferred after the SDHI issues a command.

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 49.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value to the SDARG register, write the command information to be sent to the SDCMD register. The SDHI issues a command when a value is written to the SDCMD register.
4. After a command is issued, the SDSTS1.RSPEND flag becomes 1, and a response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0.

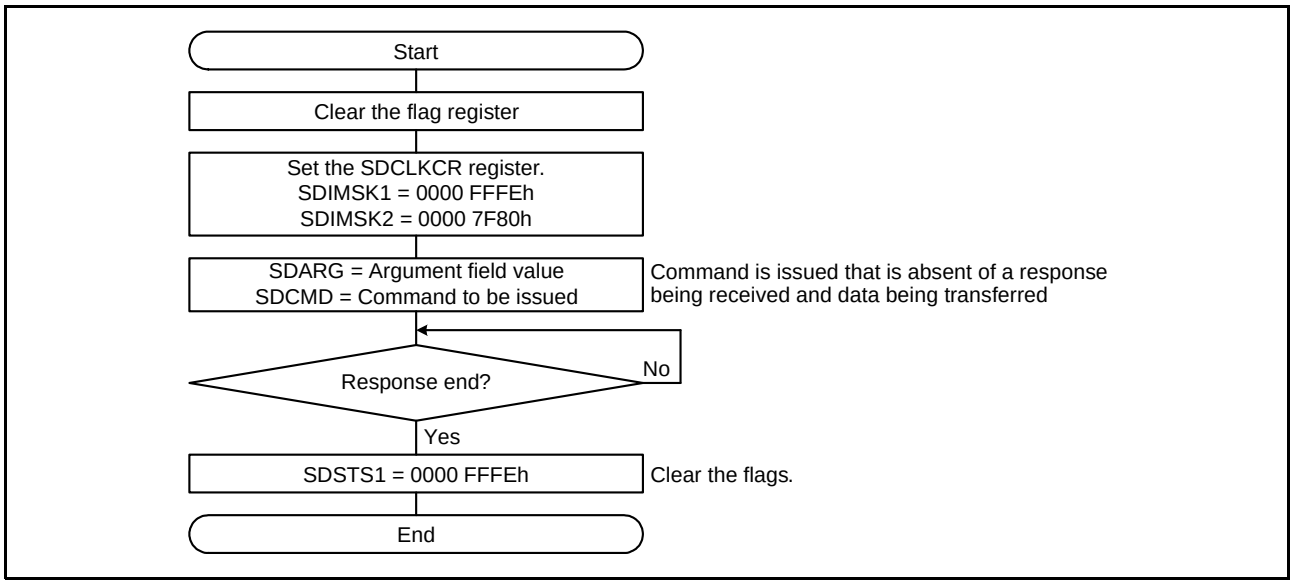


Figure 49.8 Command Issued That Is Absent of Response Reception and Data Transfer

49.3.6.2 Command Absent of Data Transfer

Figure 49.9 shows an example of no data being transferred after the SDHI issues a command.

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 49.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value in the SDARG register, write the information of the command to be issued to the SDCMD register. The SDHI issues a command when a value is written to the SDCMD register.
4. After a response is received, the SDSTS1.RSPEND flag becomes 1, and a response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0, and read the response stored in the SDRSP10 register.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

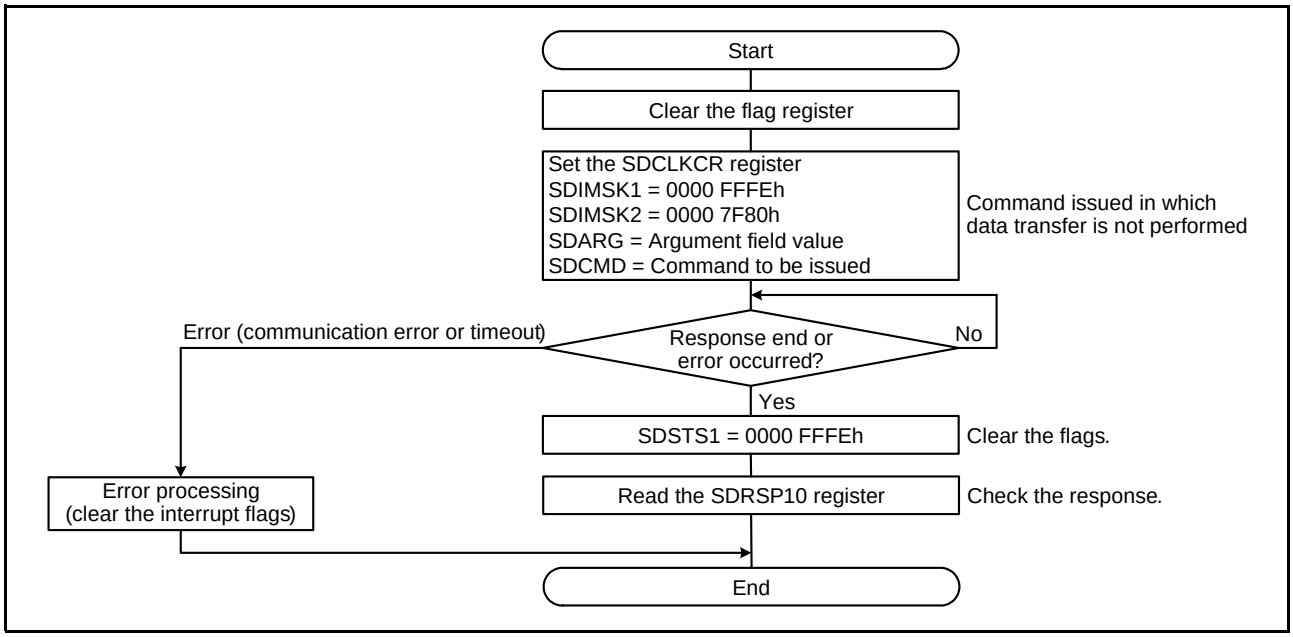


Figure 49.9 Command Issued That Is Absent of Data Transfer

49.3.6.3 Single Block Read Command (CMD17)

Figure 49.10 shows an example of issuing the single block read command (CMD17).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 49.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value for CMD17 to the SDARG register, write 0000 0011h to the SDCMD register. The SDHI issues CMD17 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and a response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0, and read the response stored in the SDRSP10 register. If the response read is in error, set the SDSTOP.STP bit or SDIOMD.IOABT bit to 1, and the command sequence can be stopped. When the command sequence is stopped, the SDSTS1.ACEND flag becomes 1. Note that CMD12 and CMD52 are not automatically issued by stopping this command sequence.
6. After the response is received, set the SDIMSK1.ACENDM bit to 0 and set the SDIMSK2.BREM bit to 0.
7. After the amount of data set in the SDSIZE.LEN[9:0] bits is received from the SD card, the SDSTS2.BRE flag becomes 1, and the BRE interrupt request is generated.
8. Set the SDSTS2.BRE flag to 0, and read the amount of data set in the SDSIZE.LEN[9:0] bits from the SDBUFR register.
9. After data has been read from the SDBUFR register, the SDSTS1.ACEND flag becomes 1, and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

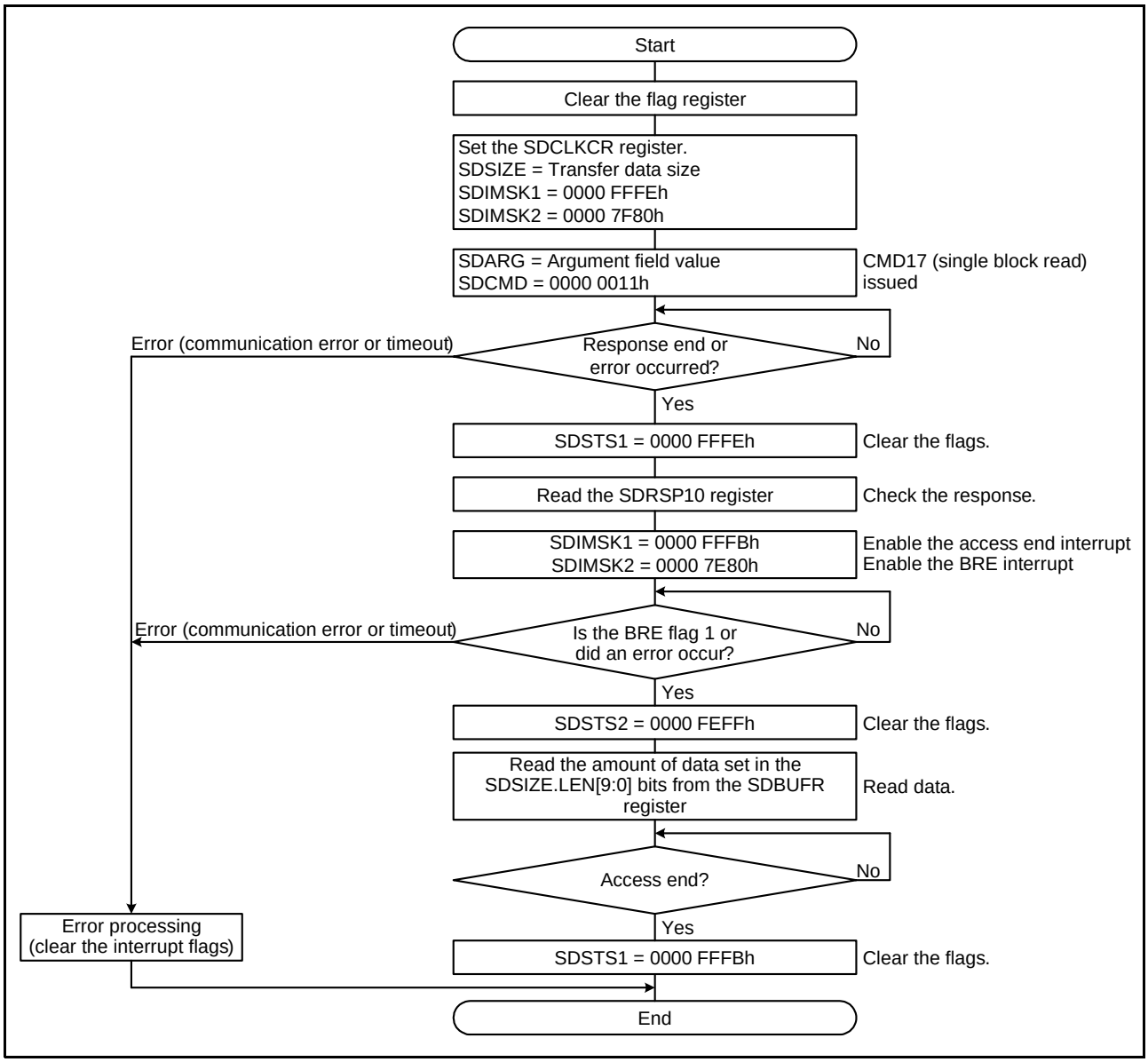


Figure 49.10 Issuing the Single Block Read Command

49.3.6.4 Single Block Write Command (CMD24)

Figure 49.11 shows an example of issuing the single block write command (CMD24).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 49.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value for CMD24 to the SDARG register, write 0000 0018h to the SDCMD register. The SDHI issues CMD24 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and the response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0 and read the response stored in the SDRSP10 register. If the read response is in error, set the SDSTOP.STP bit and SDIOMD.IOABT bit to 1, and the command sequence can be stopped. When the command sequence is stopped, the SDSTS1.ACEND flag becomes 1. Note that when this command sequence is stopped, CMD12 and CMD52 are not automatically issued.
6. After the response is received, set the SDIMSK1.ACENDM bit to 0, and set the SDIMSK2.BWEM bit to 0.
7. When the SDBUFR register becomes write accessible, the SDSTS2.BWE flag becomes 1, and the BWE interrupt request is generated.
8. Set the SDSTS2.BWE flag to 0, and write the amount of data set in the SDSIZE.LEN[9:0] bits to the SDBUFR register. After writing to the SDBUFR register, the SDHI transmits write data to the SD card. Also, after writing to the SDBUFR register, data transmission may cause a communication error or timeout to occur.
9. After all data has been written to the SD card, the SDHI receives the CRC status token, and the SDHI_D0 pin line becomes busy (low). Then, when the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1, and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

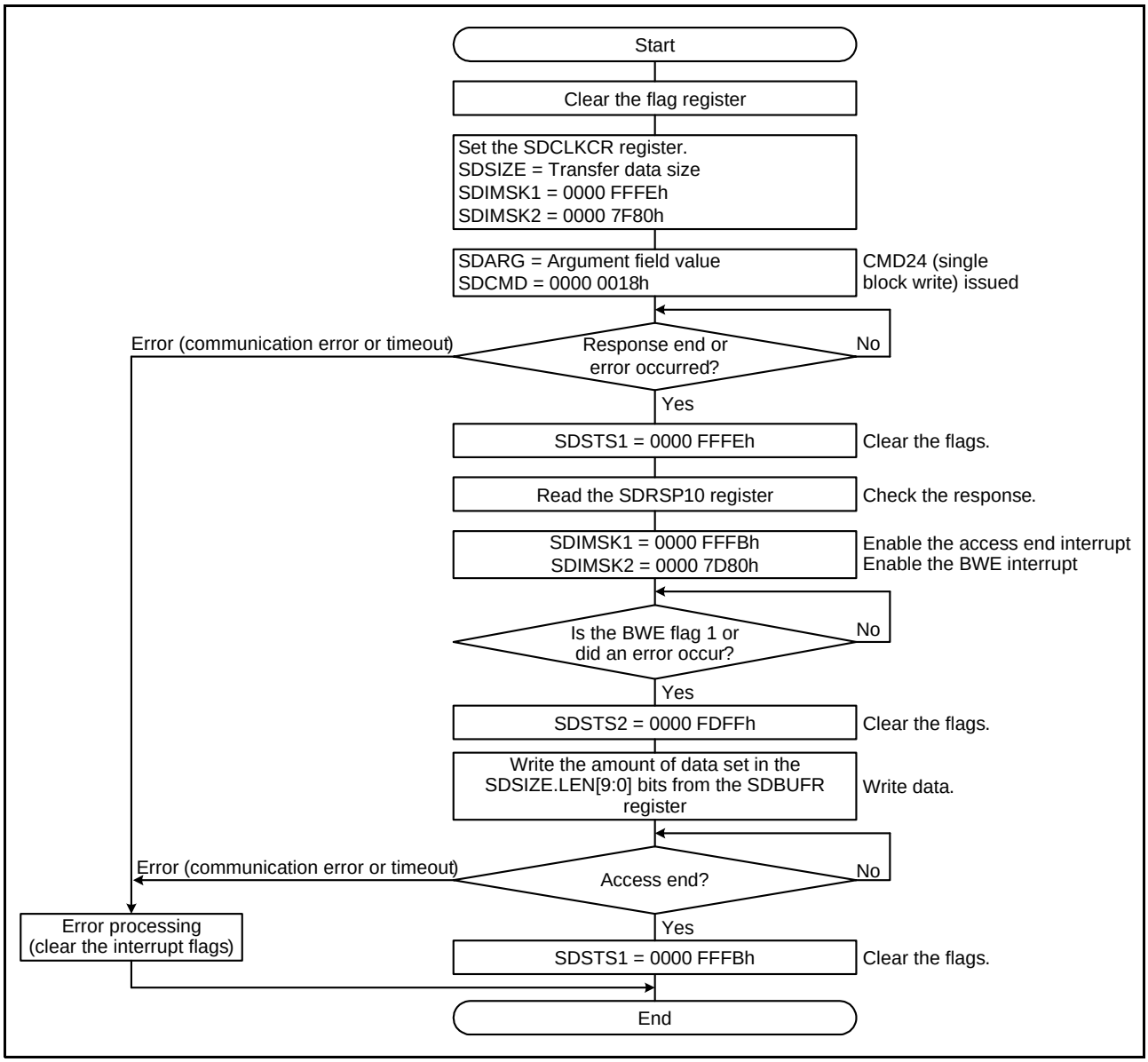


Figure 49.11 Issuing the Single Block Write Command

49.3.6.5 Multi-Block Read Command (CMD18)

Figure 49.12 shows an example of issuing the multi-block read command (CMD18).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 49.5.5 for details on setting the SDCLKCR register. Set the SDSTOP.SDBLKCNTEN bit to 1, and set the number of transfer blocks in the SDBLKCNT register.
3. After setting the argument field value for CMD18 to the SDARG register, write 0000 0012h to the SDCMD register. The SDHI issues CMD18 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and the response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0 and read the response stored in the SDRSP54 register. If the read response is in error, set the SDSTOP.STP bit to 1, and the command sequence can be stopped. When the SDSTOP.STP bit is set to 1, the SDHI automatically issues CMD12, and the response is received. At this point, the SDSTS1.ACEND flag becomes 1, and if the access end interrupt request is enabled, the access end interrupt request is generated. Next, set the SDSTS1.ACEND flag to 0 and read the response.
6. After the response is received, set the SDIMSK1.ACENDM bit to 0, and set the SDIMSK2.BREM bit to 0.
7. After receiving one block of data from the SD card, the SDSTS2.BRE bit becomes 1, and the BRE interrupt request is generated.
8. Set the SDSTS2.BRE flag to 0, and read the amount of data set in the SDSIZE.LEN[9:0] bits from the SDBUFR register. The read access to the SDBUFR register repeats for the amount of transfer blocks set in the SDBLKCNT register. Also, while reading the SDBUFR register, data reception may cause a communication error or timeout to occur. After the amount of transfer blocks set in the SDBLKCNT register have been read, the SDHI automatically issues CMD12, and the response is received. At this time, the SDHI automatically writes 0000 0000h to the SDARG register.
9. When all blocks have been received and the CMD12 response is received, the SDSTS1.ACEND flag becomes 1 and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0 and read the response.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

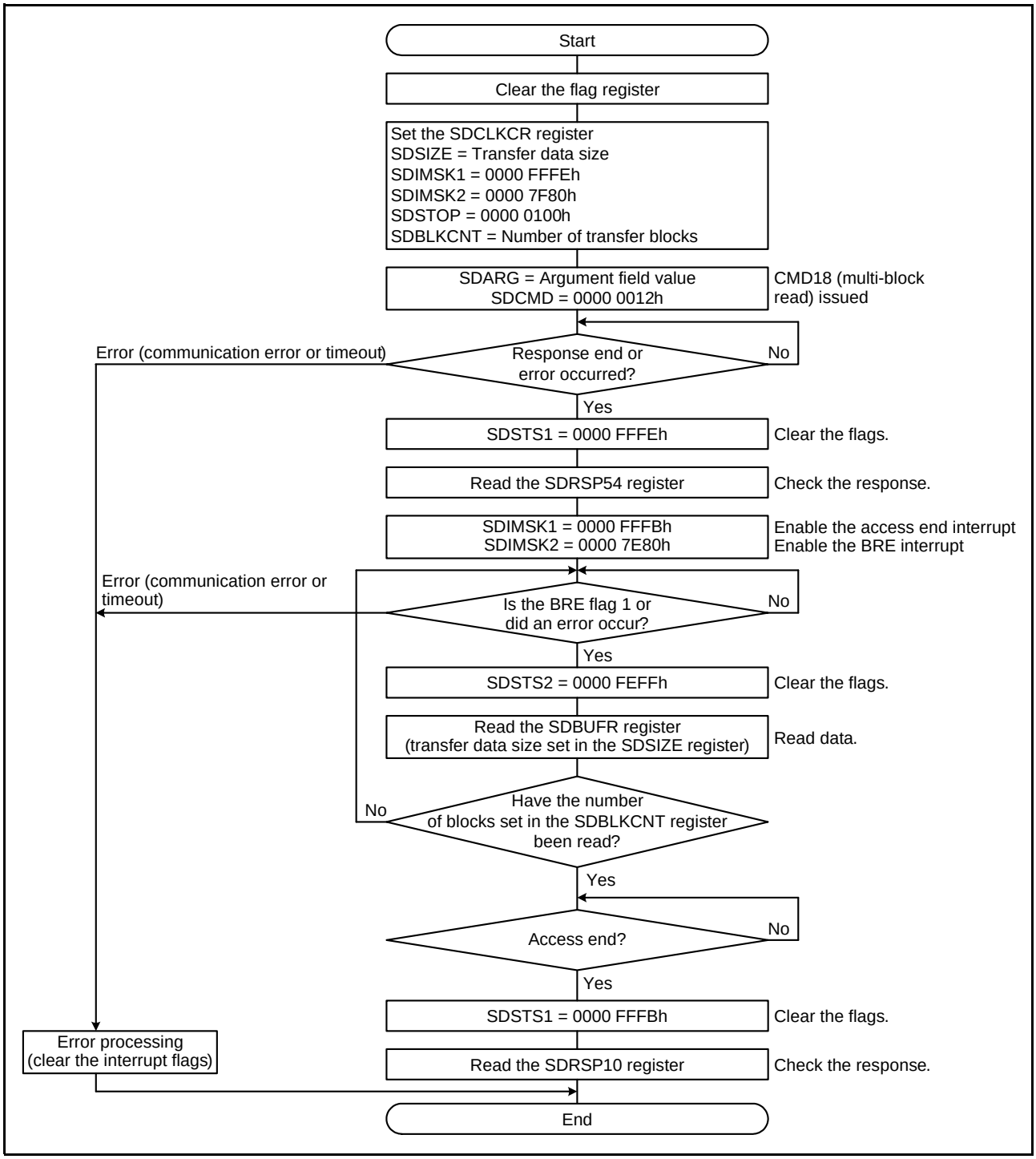


Figure 49.12 Issuing the Multi-Block Read Command

49.3.6.6 Multi-Block Write Command (CMD25)

Figure 49.13 shows an example of issuing the multi-block write command (CMD25).

1. Initialize the flags in registers SDSTS1 and SDSTS2.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 49.5.5 for details on setting the SDCLKCR register. Set the SDSTOP.SDBLKCNTEN bit to 1, and set the number of transfer blocks in the SDBLKCNT register.
3. After setting the argument field value for CMD25 to the SDARG register, write 0000 0019h to the SDCMD register. The SDHI issues CMD25 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and the response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0 and read the response stored in the SDRSP54 register. If the read response is in error, set the SDSTOP.STP bit to 1, and the command sequence can be stopped. When the SDSTOP.STP bit is set to 1, the SDHI automatically issues CMD12, and the response is received. At this point, the SDSTS1.ACEND flag becomes 1, and if the access end interrupt request is enabled, the access end interrupt request is generated. Next, set the SDSTS1.ACEND flag to 0 and read the response.
6. After the response is received, configure the SDIMSK1 register to enable the access end interrupt request, and configure the SDIMSK2 register to enable the BWE interrupt request.
7. When the SDBUFR register becomes write accessible, the SDSTS2.BWE flag becomes 1, and the BWE interrupt request is generated.
8. Set the SDSTS2.BWE flag to 0, and write the amount of data set in the SDSIZE.LEN[9:0] bits to the SDBUFR register. After writing to the SDBUFR register, and after the SDHI transmits write data to the SD card, the CRC status token is received, and the SDHI_D0 pin line becomes busy (low). The write access to the SDBUFR register and CRC status token reception repeat for the amount of transfer blocks set in the SDBLKCNT register. Also, after writing to the SDBUFR register, data transmission may cause a communication error or timeout to occur. After the amount of transfer blocks set in the SDBLKCNT register have been written, the SDHI automatically issues CMD12, and the response is received. At that time, the SDHI automatically writes 0000 0000h to the SDARG register.
9. When all blocks have been transmitted and the CRC status token is received, the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1, and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0 and read the response.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

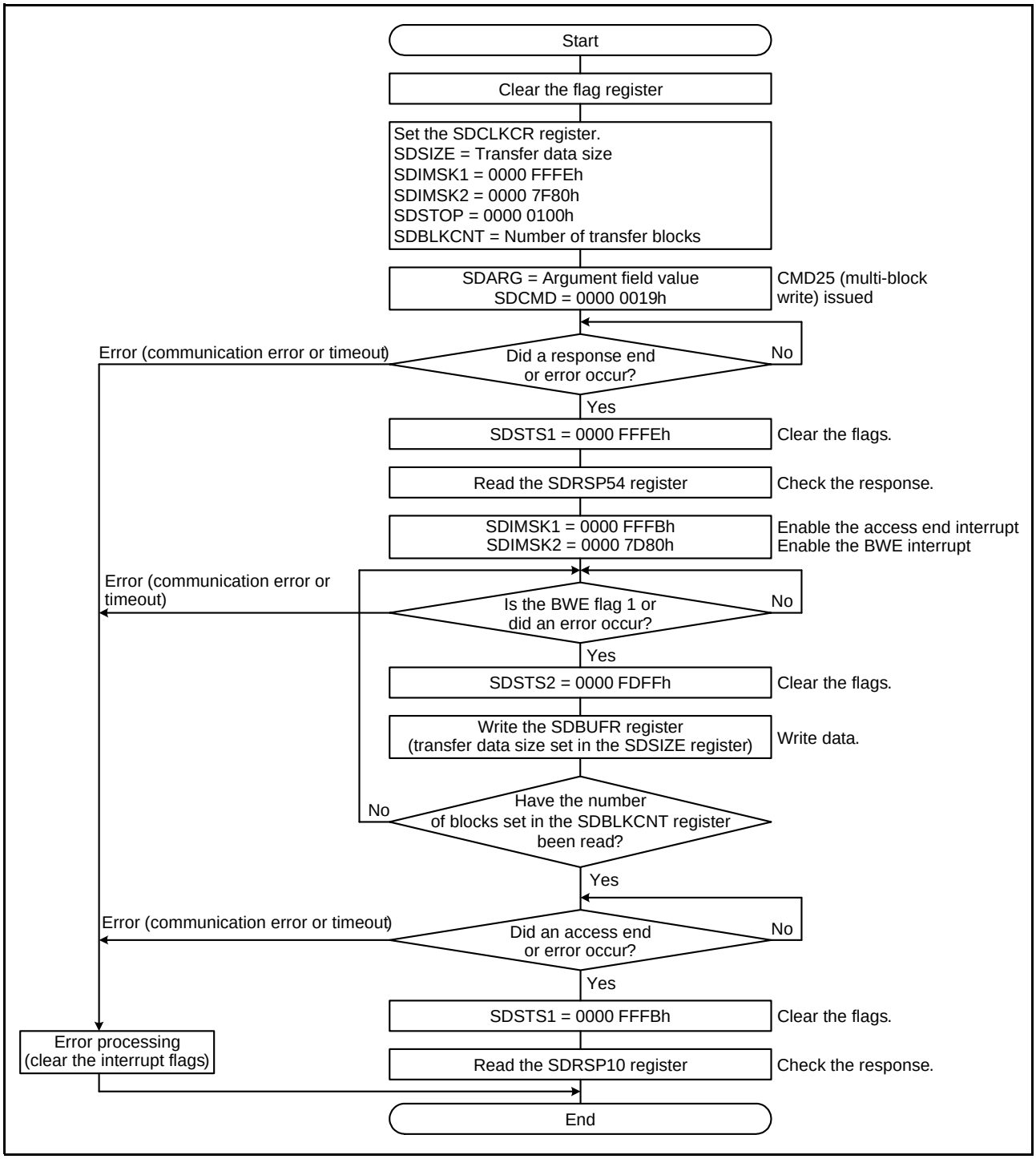


Figure 49.13 Issuing the Multi-Block Write Command

49.3.6.7 IO_RW_DIRECT Command (CMD52)

Figure 49.14 shows an example of issuing the IO_RW_DIRECT command (CMD52).

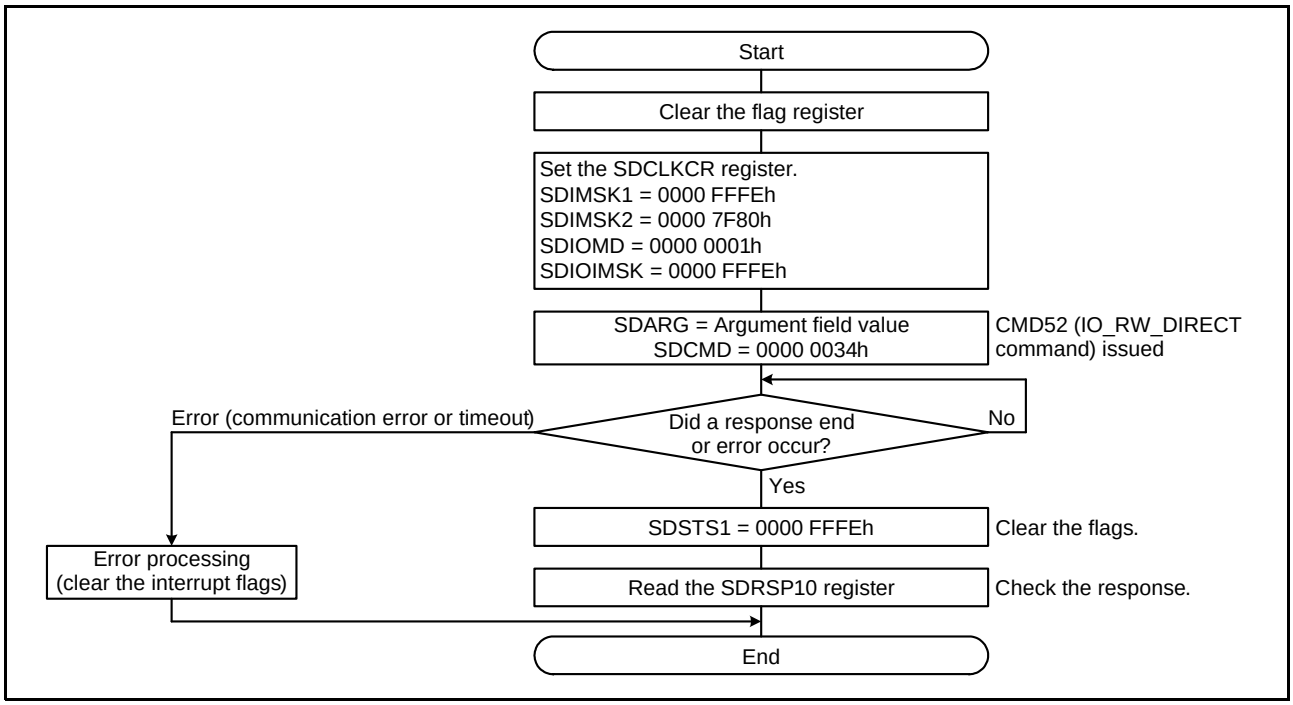


Figure 49.14 Issuing the IO_RW_DIRECT Command

49.3.6.8 IO_RW_EXTENDED Command (CMD53 (Multi-Block Read))

Figure 49.15 shows an example of issuing the IO_RW_EXTENDED command (CMD53/Multi-block read).

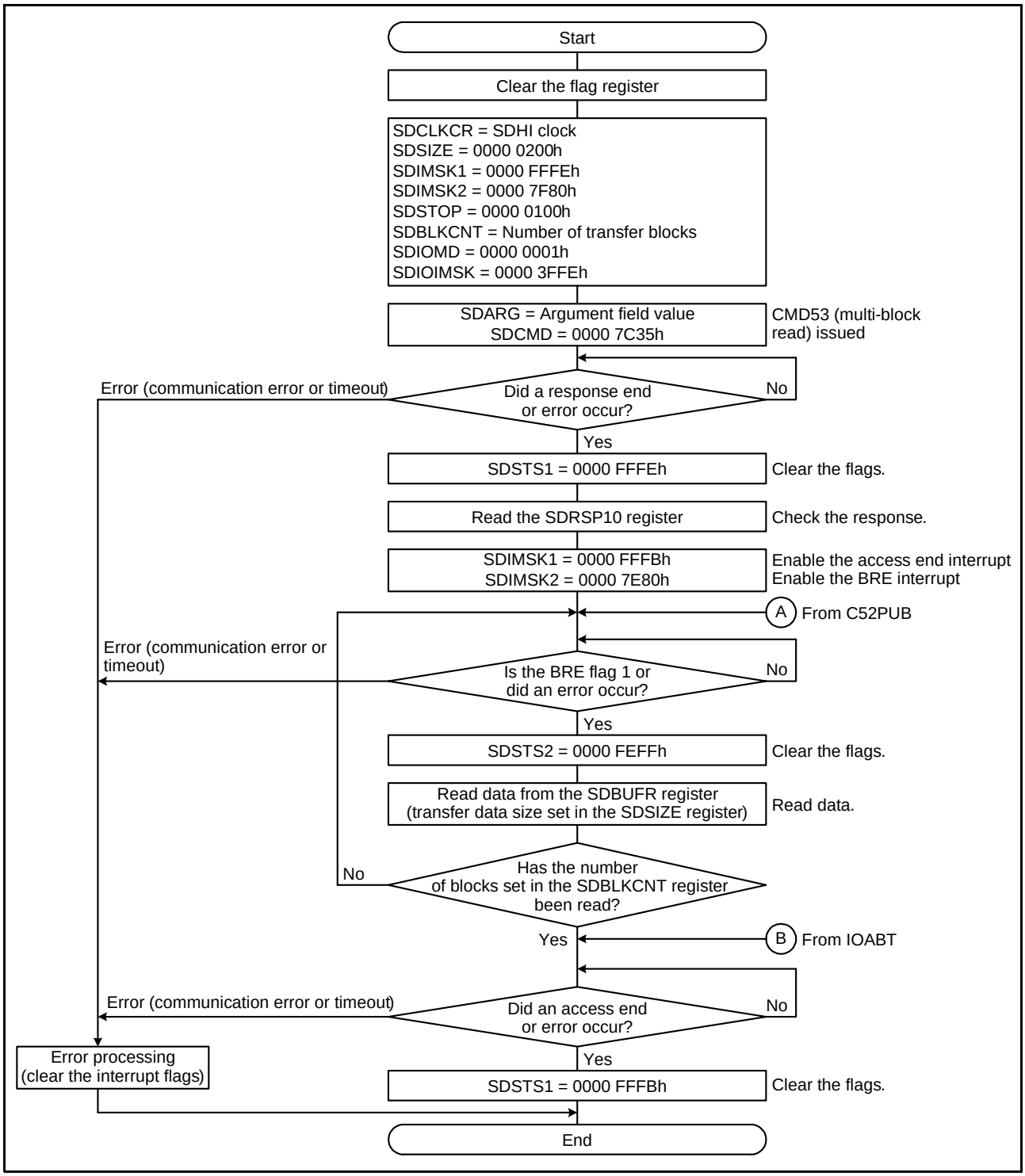


Figure 49.15 Issuing the IO_RW_EXTENDED Command (CMD53/Multi-Block Read)

Figure 49.16 shows an example of entering the read wait state and then issuing the SDIO none abort command (CMD52) during the IO_RW_EXTENDED command (CMD53/Multi-block read).

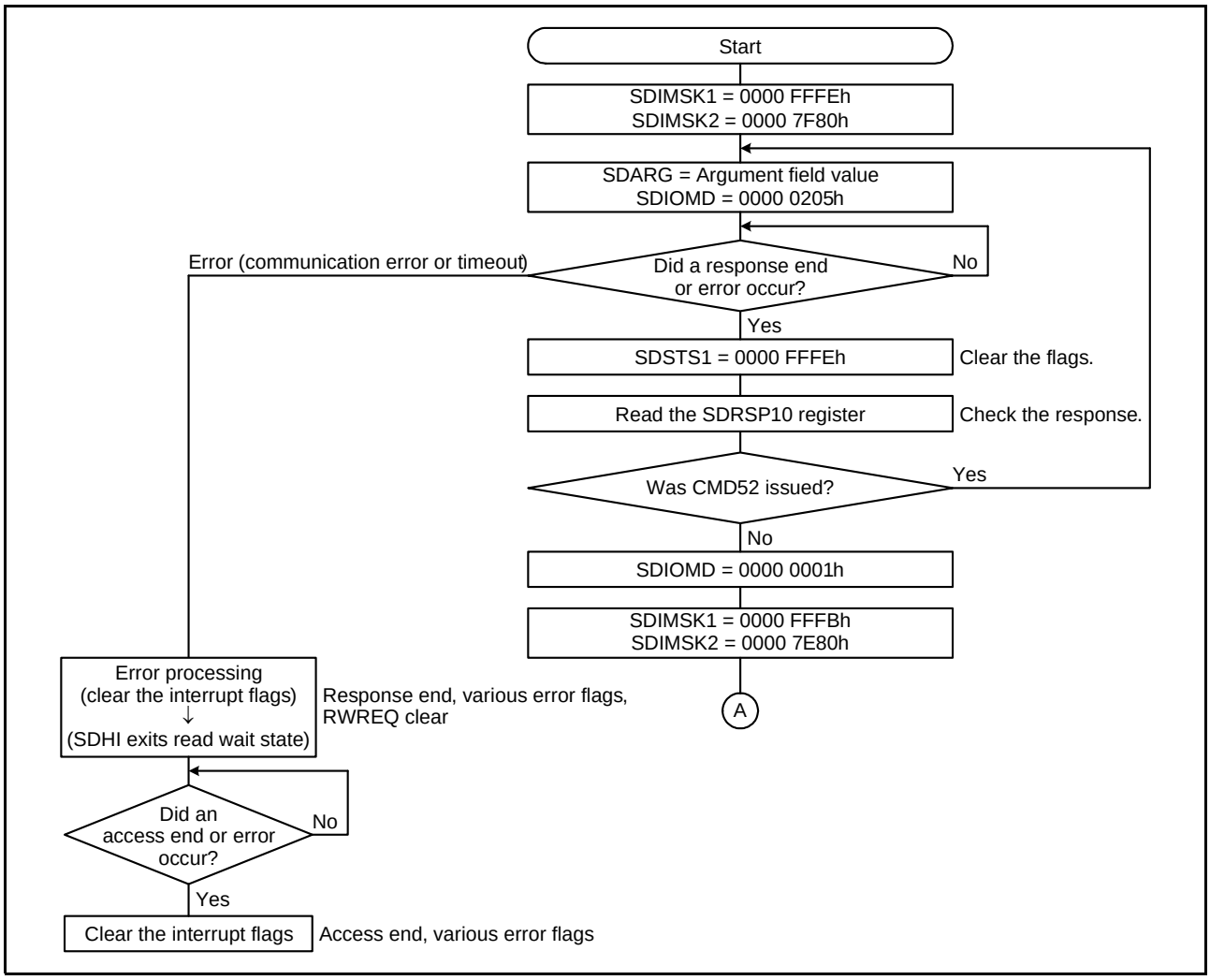


Figure 49.16 Entering the Read Wait State and Then Issuing the SDIO None Abort Command (CMD52) During the IO_RW_EXTENDED Command (CMD53/Multi-Block Read)

Figure 49.17 shows an example of SDIO abort (CMD52) issued during IO_RW_EXTENDED command (CMD53/ Multi-block read) sequence.

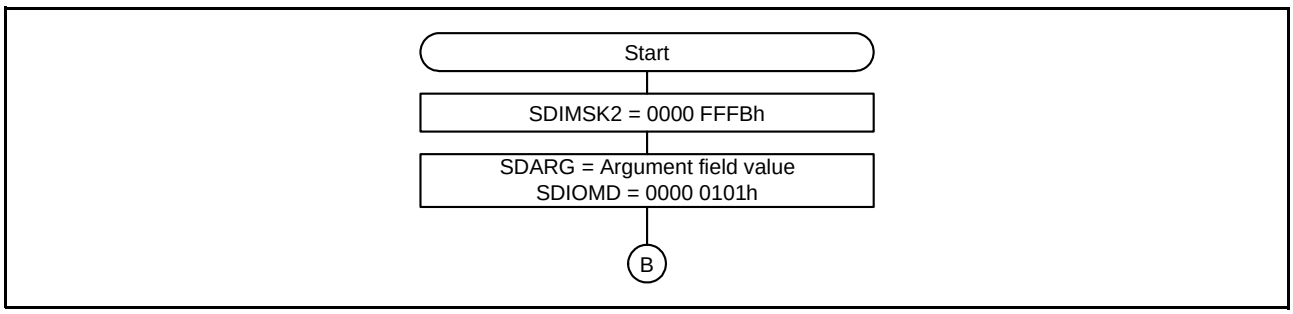


Figure 49.17 SDIO Abort (CMD52) Issued During IO_RW_EXTENDED Command (CMD53/Multi-Block Read) Sequence

49.3.6.9 IO_RW_EXTENDED (CMD53 Multi-Block Write)

Figure 49.18 shows an example of issuing the IO_RW_EXTENDED command (CMD53/Multi-block write).

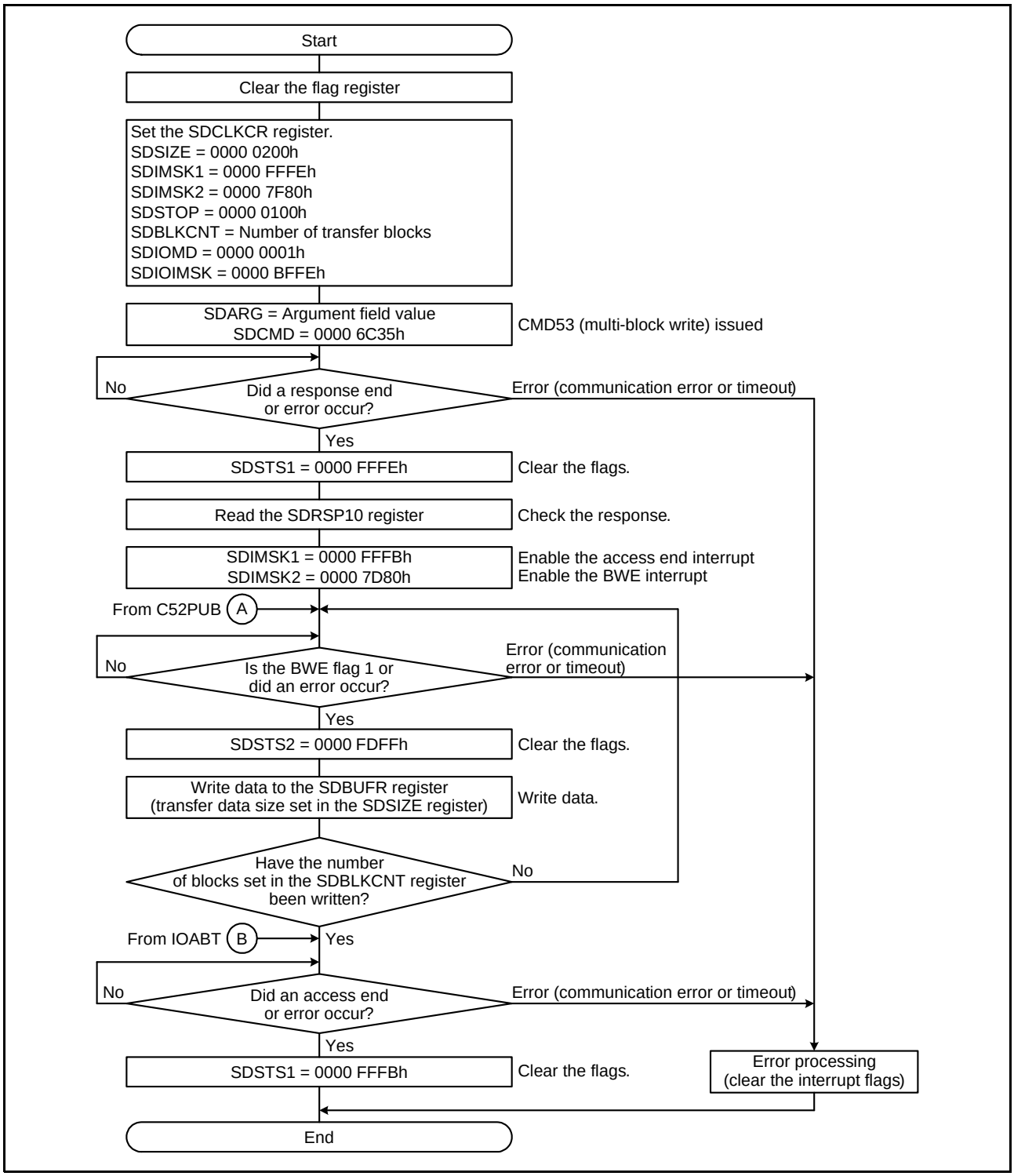


Figure 49.18 Issuing the IO_RW_EXTENDED Command (CMD53/Multi-Block Write)

Figure 49.19 shows SDIO none abort (CMD52) issued during IO_RW_EXTENDED command (CMD53/Multi-block write) sequence.

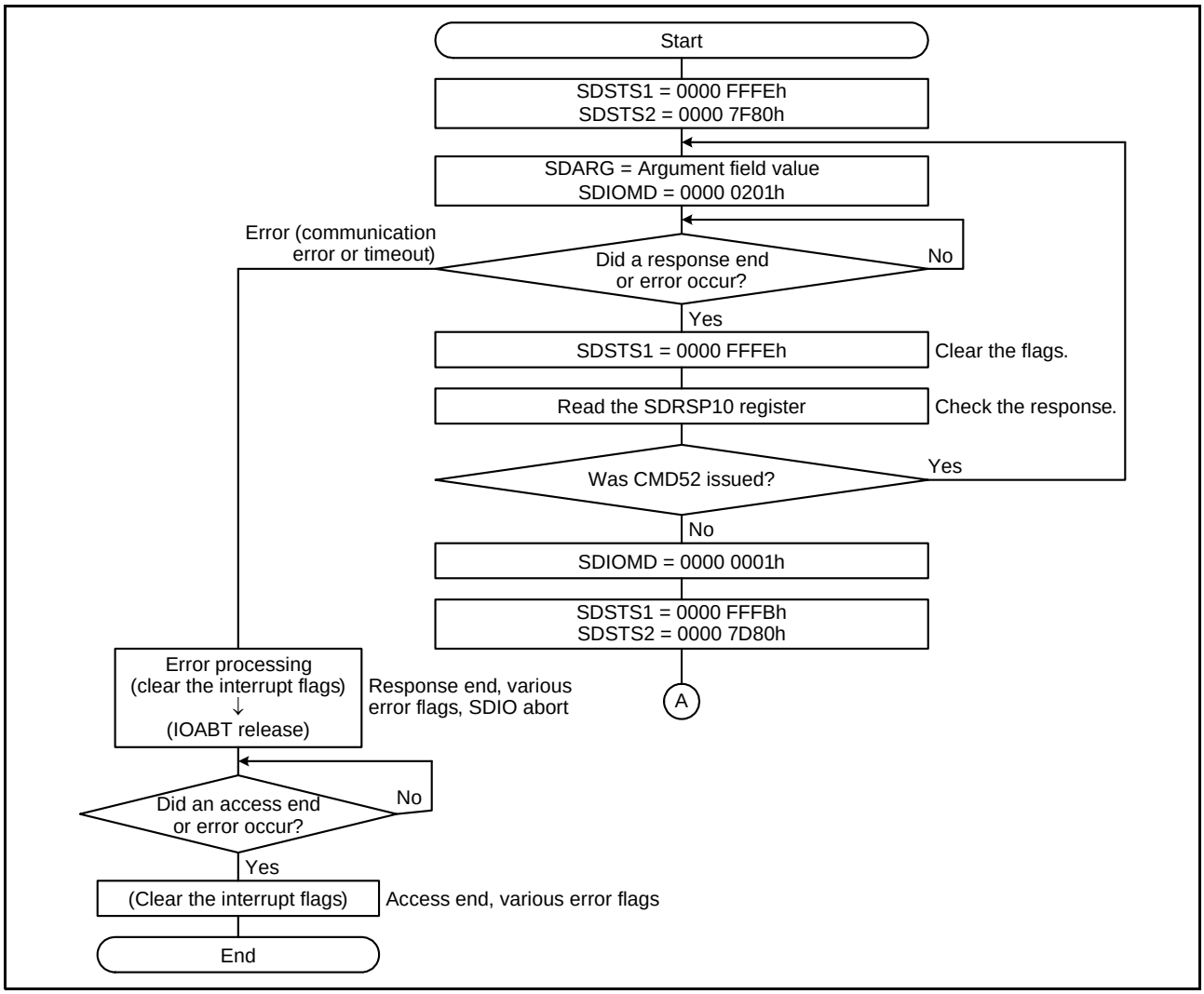


Figure 49.19 SDIO None Abort (CMD52) Issued During IO_RW_EXTENDED Command (CMD53/Multi-Block Write) Sequence

Figure 49.20 shows SDIO abort (CMD52) issued during IO_RW_EXTENDED command (CMD53) sequence.

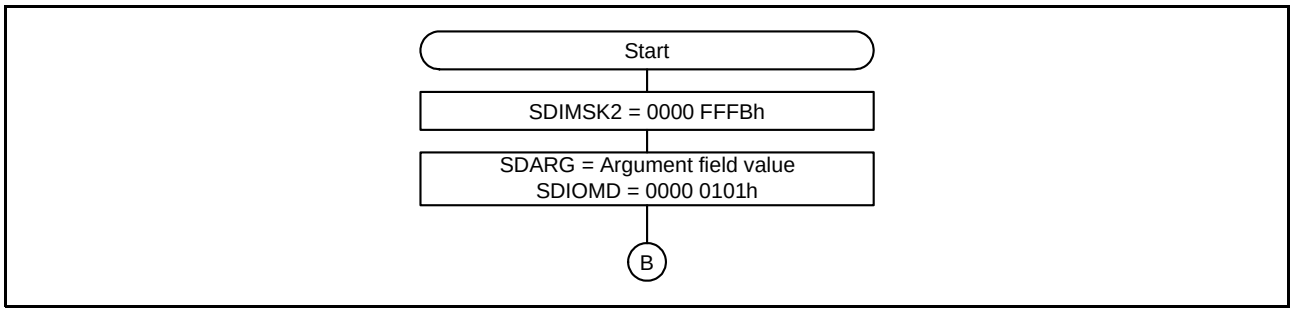


Figure 49.20 SDIO Abort (CMD52) Issued During IO_RW_EXTENDED Command (CMD53) Sequence

49.3.6.10 DMA Transfer

Figure 49.21 shows an example of data being transferred from the SDBUFR register after the CMD18 multi-block read command is issued.

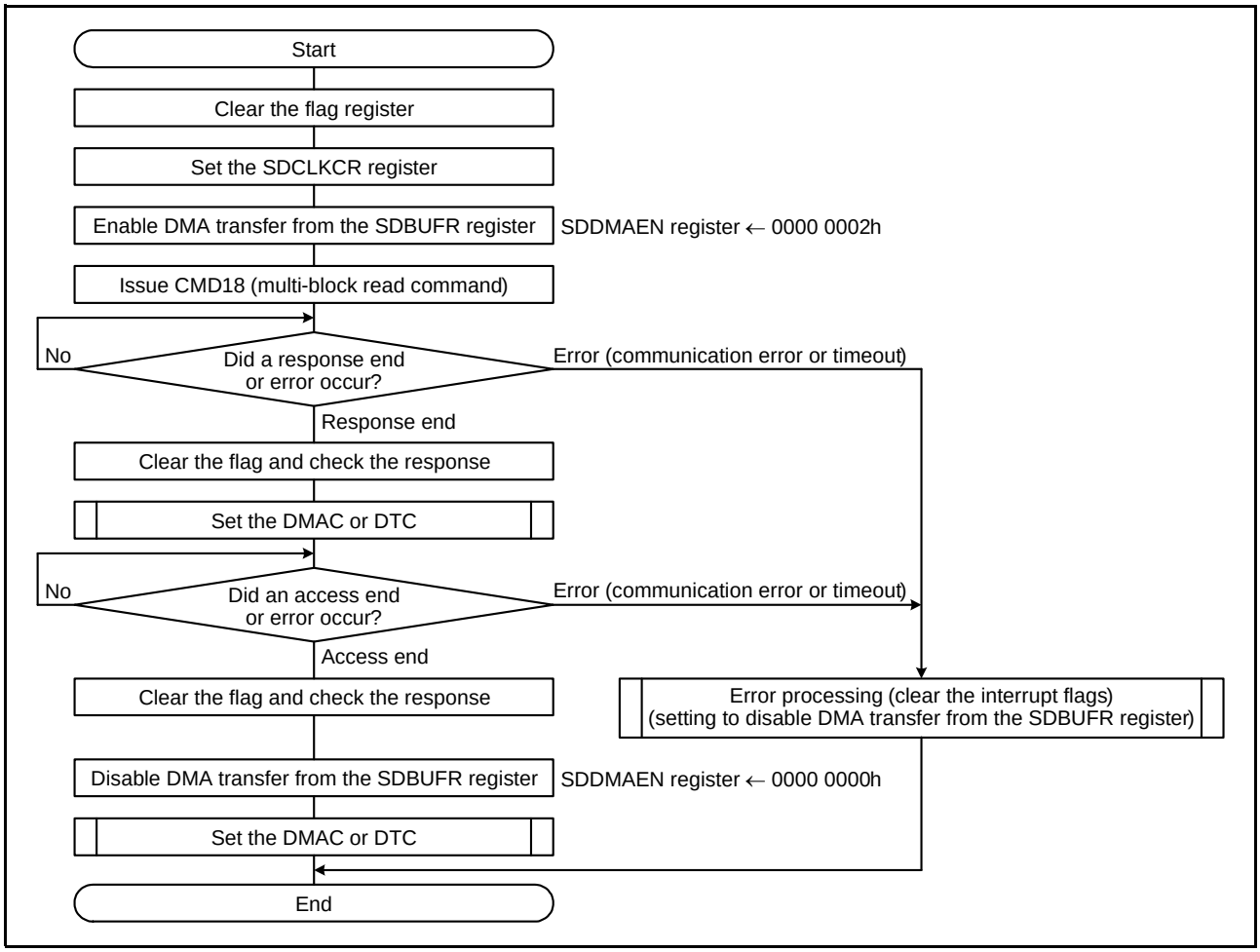


Figure 49.21 DMA Transfer After CMD18 is Issued

Figure 49.22 shows an example of data being DMA transferred to the SDBUFR register after the CMD25 multi-block write command is issued.

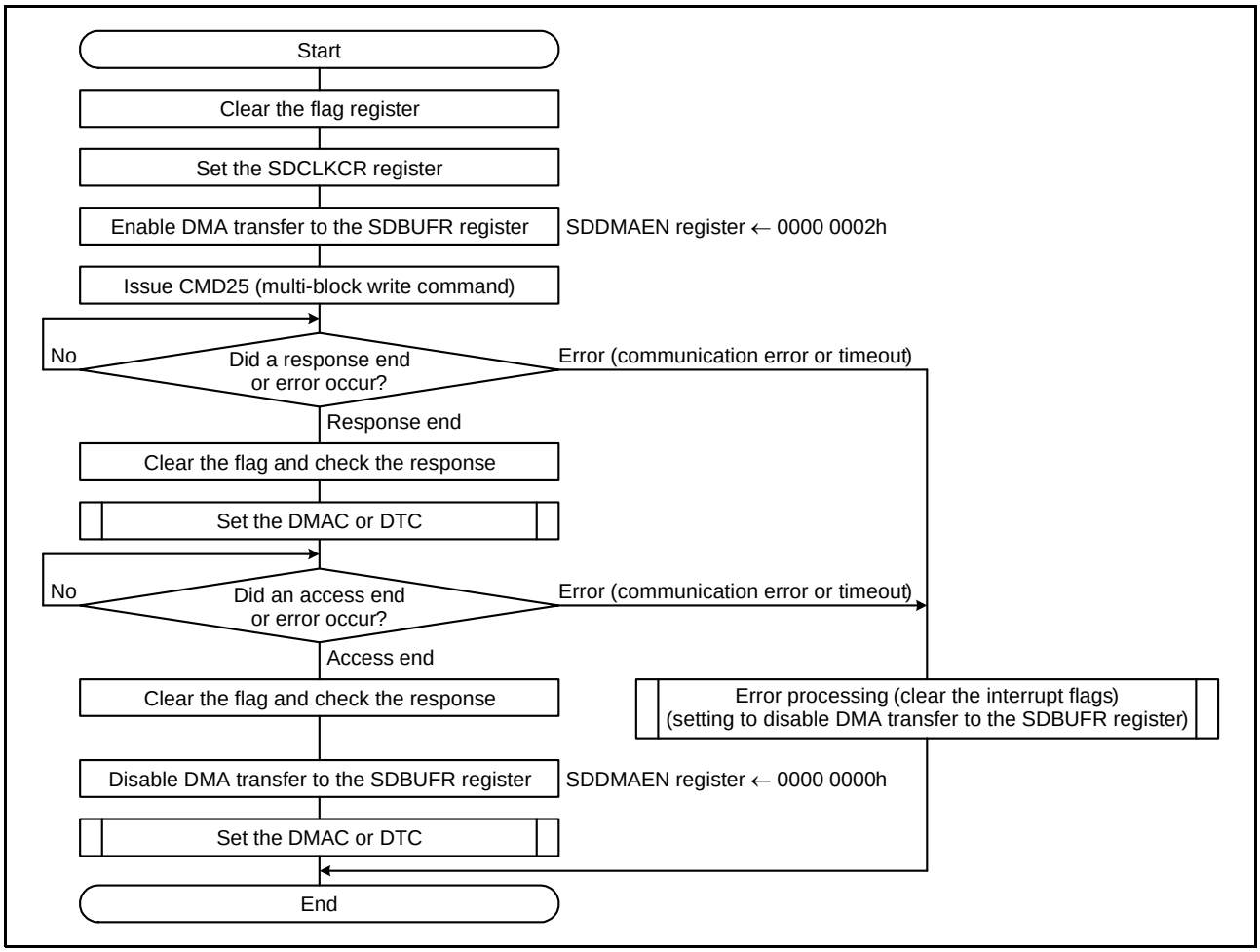


Figure 49.22 DMA Transfer After CMD25 is Issued

49.4 Interrupts

Table 49.8 lists the SDHI interrupt sources. When the status flags in registers SDSTS1, SDSTS2, and SDIOSTS become 1, if the corresponding bits in registers SDIMSK1, SDIMSK2, and SDIOIMSK are 0, the SDHI requests an interrupt. When clearing the status flags in registers SDSTS1, SDSTS2, and SDIOSTS, write 0 to the status flags to be cleared and write 1 to the states flags not being cleared.

Table 49.8 Interrupt Sources

Interrupt Source	Status Flag Register		Interrupt Mask/Enable Register		DMAC/DTC Triggerable
	Register symbol	Bit symbol	Register symbol	Bit symbol	
CACI	SDSTS1	ACEND	SDIMSK1	ACENDM	No
		RSPEND		RSPENDM	
	SDSTS2	ILA	SDIMSK2	ILAM	
		BWE		BWEM	
		BRE		BREM	
		RSPTO		RSPTOM	
		ILR		ILRM	
		ILW		ILWM	
		DTO		DTTOM	
		ENDE		ENDEM	
		CRCE		CRCEM	
		CMDE		CMDEM	
SDACI	SDIOSTS	EXWT	SDIOIMSK	EXWTM	No
		EXPUB52		EXPUB52M	
		IOIRQ		IOIRQM	
CDETI	SDSTS1	SDD3IN	SDIMSK1	SDD3INM	No
		SDD3RM		SDD3RMM	
		SDCDIN		SDCDINM	
		SDCDRM		SDCDRMM	
SBFAI	SDSTS2	BWE	SDDMAEN	DMAEN	Yes
		BRE			

49.4.1 DMA Transfer Triggered by Interrupt Requests

When the SBFAI interrupt is requested, DMA/DTC transfer can be used to write to or read the SDBUFR register. When using the SBFAI interrupt, set the SDDMAEN.DMAEN bit to 1, the SDIMSK2.BWEM bit to 1, and SDIMSK2.BREM bit to 1.

When the SDDMAEN.DMAEN bit is 1, if a write command is issued, the SDSTS2.BWE flag becomes 1; if a read command is issued, the SDSTS2.BRE flag becomes 1. At this point, the SBFAI interrupt request is output. When the last data of a block is transferred (one block is the transfer data size set in the SDSIZE.LEN[9:0] bits), the SBFAI interrupt request is canceled, and the SDSTS2.BWE flag or the SDSTS2.BRE flag becomes 0.

The SBFAI interrupt request is also canceled by following:

- The SDRST.SDRST bit is set to 0 (SDHI software reset).
- The SDSTOP.STP bit is set to 1.
- The SDIOMD.IOABT bit is set to 1.
- The SDDMAEN.DMAEN bit is set to 0.
  However, if the DMAEN bit is set to 1 again before the next command is written to the SDCMD register, the SBFAI interrupt request is output again.

The SDSTS2.BWE flag and BRE flag will not become 0 when a communication error or timeout occurs, nor will they become 0 when the SDSTOP.STP bit and SDIOMD.IOABT bit are set to 1. If the SDSTS2.BWE flag or BRE flag remain set to 1, even if a write command or read command is issued, the SBFAI interrupt request will not be output, and the SDSTS2.BWE flag and BRE flag must be set to 0 before issuing the next command.

Table 49.9 lists the DMAC and DTC settings when performing DMA transfer.

Table 49.9 DMAC and DTC Settings When Performing DMA Transfer

Item		Setting Description
Transfer mode		Block transfer mode
Transfer data	1 data	32 bits
	Block size	Size set in the SDSIZE.LEN[9:0] bits divided by 4
Number of block transfers		Number of transfers set in the SDBLKCNT register

49.5 Notes on Using the SDHI

49.5.1 Illegal Read Access During a Multi-Block Read and How To Avoid It

When the multi-block read command (CMD18) is issued to read one or two blocks, if the response to CMD18 stored in the SDRSP10 register is read, the timing of the read access may cause the response to be read incorrectly.
Figure 49.23 shows examples of a normal read and an illegal read when using CMD18 to read two blocks.

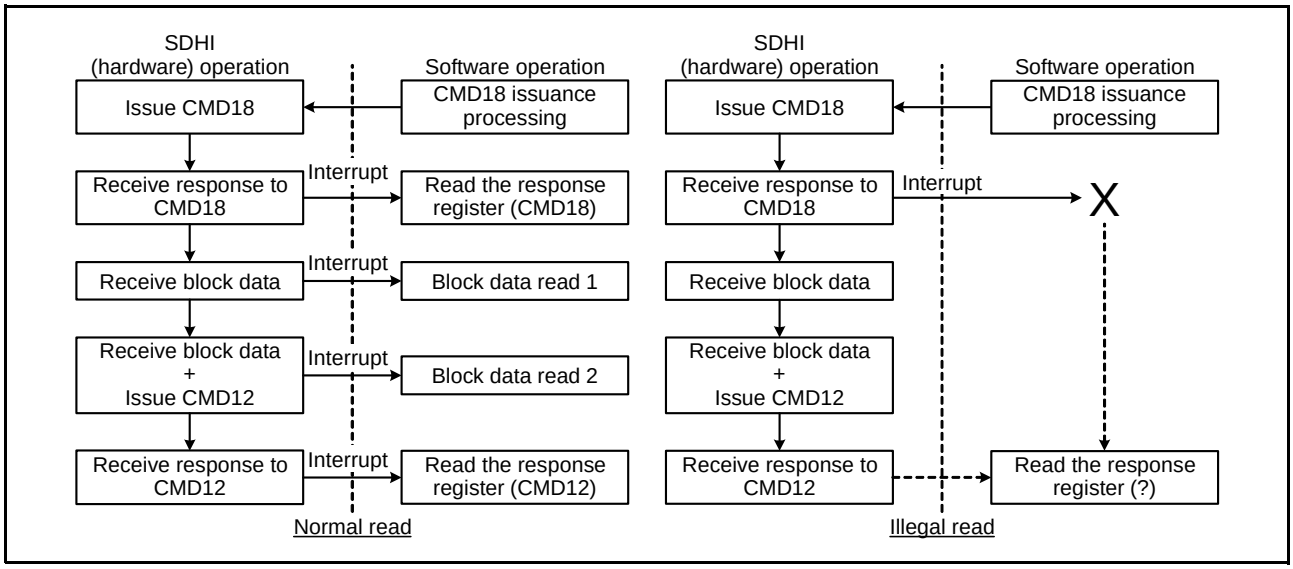


Figure 49.23 Multi-Block Read Processing When Reading Two Blocks

In the example of the illegal read, when the interrupt is generated by receiving the response to CMD18, the timing for reading the SDRSP10 register during the interrupt handling is delayed, and instead of the response for CMD18 being read, the response for CMD12 or the data during the response for CMD12 may be read.
This problem can be avoided by performing either of the following:

- When reading one block or two blocks, use a single block read command instead of a multi-block read command.
- When reading the response for CMD18, read the SDRSP54 register instead of the SDRSP10 register.

When using a multi-block read command to read three or more blocks, CMD12 will not be issued if no block data is read, which will prevent this problem from happening. Also, when using a multi-block write command, the above problem can be avoided by transmitting block data after reading the response to CMD25.

49.5.2 SDBUFR Register Illegal Write Error

When writing to the SDBUFR register after issuing a single block write command or a multi-block write command, write data for the size set by the SDSIZE.LEN[9:0] bits. If the amount of data written to the SDBUFR register is greater than the size set by the SDSIZE.LEN[9:0] bits, an illegal write error occurs in the SDBUFR register, and the SDSTS2.ILW flag becomes 1. However, the padding data included in the data being written to the SDBUFR register is ignored, so this error does not occur. For example, if the data size set by the SDSIZE.LEN[9:0] bits is an odd number, of the data written to the SDBUFR register, there is a remainder of 1 byte or 3 bytes. Although the extra data is written to the register, no error occurs. If the data size set by the SDSIZE.LEN[9:0] bits is an even number, and there is a remainder of 2 bytes, no error occurs even if these 2 bytes are written to the SDBUFR register.
If data written to the SDBUFR register is not transmitted, the SDSTS2.SDCLKCREN flag may remain set to 0. In this case, in order to set the SDSTS2.SDCLKCREN flag to 1, the SDRST.SDRST bit must be set to 0 and then set back to 1.

49.5.3 Automatic Control of the SDHI Clock Output

As per the SD card specifications, after MCU power-on, 74 cycles of the SDHI clock must be output from the host to the SD card before the card initialization command (CMD0) can be issued. Therefore, 74 cycles of the SDHI clock should be output from the SDHI to the SD card before enabling automatic control of the SDHI clock output.
When automatic control of the SDHI clock output is enabled, SDHI clock output stops if the command sequence is ended by a communication error or timeout. Therefore, if it is necessary to change the internal status of the SD card even after the command sequence ends, disable automatic control of the SDHI clock output and output the SDHI clock to the SD card.

49.5.4 Restrictions on Setting the C52PUB Bit During a Multi-Block Write Sequence

During a CMD53 multi-block write sequence, if the SDIOMD.C52PUB bit is set to 1, the SDHI issues CMD52 after the SD buffer becomes empty. To immediately issue CMD52, perform one of the procedures below to suspend writing to the SD buffer, and set the C52PUB bit to 1.

Procedure to suspend writing to the SD buffer when not performing DMA transfer (interrupt used)

1. Set the SDIMSK2.BWEM bit to 1 to disable the interrupt, and suspend writing to the SDBUFR register.
2. Set the SDIOMD.C52PUB bit to 1. Then, when the SD buffer becomes empty, the SDHI issues CMD52.
3. After receiving the response for CMD52, set the SDIMSK2.BWEM bit to 0 to enable the interrupt, and resume writing to the SDBUFR register.

Procedure to suspend writing to the SD buffer when performing DMA transfer

1. Configure settings to perform DMA transfer every [SDSIZE register setting value × *n* blocks], and suspend writing to the SDBUFR register before setting the SDIOMD.C52PUB bit (*n* = 1, 2, ...).
2. Set the SDIOMD.C52PUB bit to 1. Then, when the SD buffer becomes empty, the SDHI issues CMD52.
3. After receiving the response for CMD52, resume DMA transfer to the SDBUFR register.

49.5.5 Note on Setting the SDCLKCR Register

The SDCLKCR register cannot be written when the SDSTS2.SDCLKCREN flag is 0. Set the SDSTS2.SDCLKCREN flag to 1 before writing to the SDCLKCR register.

49.5.6 Writing to the SDSTOP Register During a Multi-Block Read Sequence

When the SDSTOP.SDBLKCNTEN bit is 1 during a multi-block read sequence, if the SDSTOP.STP bit is set to 1 and the command sequence is stopped, the command sequence may not be completed depending on when the SDSTOP.STP bit is set to 1. To avoid this problem, set the SDSTOP.STP bit and the SDSTOP.SDBLKCNTEN bit to 0 simultaneously. Note that at this time, the SDSTOP.SDBLKCNTEN bit should be set to 0 even if the SDSTS2.SDCLKCREN bit is 0.
If the SDSTOP.SDBLKCNTEN bit is not set to 0, the command sequence can be completed by setting the SDRST.SDRST bit to 0.
During a CMD53 multi-block transfer, when stopping data transfer by setting the SDIOMD.IOABT bit to 1, the SDSTOP.SDBLKCNTEN bit should remain set to 1.

49.5.7 Controlling Module Operation

SDHI operation is controlled by setting the MSTPCRD.MSTPD19 bit. Setting the MSTPD19 bit to 0 enables the SDHI; setting the MSTPD19 bit to 1 disables the SDHI. The SDHI is disabled after a reset. Registers in the SDHI can be accessed by setting the MSTPD19 bit to 0. Refer to section 11, Low Power Consumption for details.

50. MultiMediaCard Interface (MMCIF)

50.1 Overview

This MCU incorporates a MultiMediaCard interface (MMCIF) that conforms to the JEDEC Standard JESD84-A441. Table 50.1 lists the specifications of the MMCIF.

Table 50.1 MMCIF Specifications

Item	Function
MMC bus interface	Transfer bus width selectable from 1, 4, or 8 bits
Transfer modes	• Backward compatible mode or high-speed mode selectable • Supports the single data rate • Supports block transfer (stream read and stream write operations are not supported)
Clocks	The MMCIF clock is generated by dividing peripheral module clock B (PCLKB) by *n*, where *n* = 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1,024.
Boot operation mode	The MMCIF clock frequency is switchable during boot operation (alternative boot operation is not supported).
MMCIF buffers	512 bytes × 2
Interrupt sources	• Normal operation interrupt (ACCIO) • Error/timeout interrupt (ERRIO) • MMC detection interrupt (CDETIO) • MMCIF buffer access interrupt (MBFAI)
DMA transfer requests	• MBFAI interrupt can be used to trigger the DMAC and DTC • DMAC or DTC can be used to read and write to the MMCIF buffer
Other functions	• MMC detection • Supports background operations • Supports the high priority interrupt (HPI) during CMD6, CMD24, CMD25 (pre-defined), and CMD38 command sequences

Figure 50.1 shows a Block Diagram of the MMCIF.

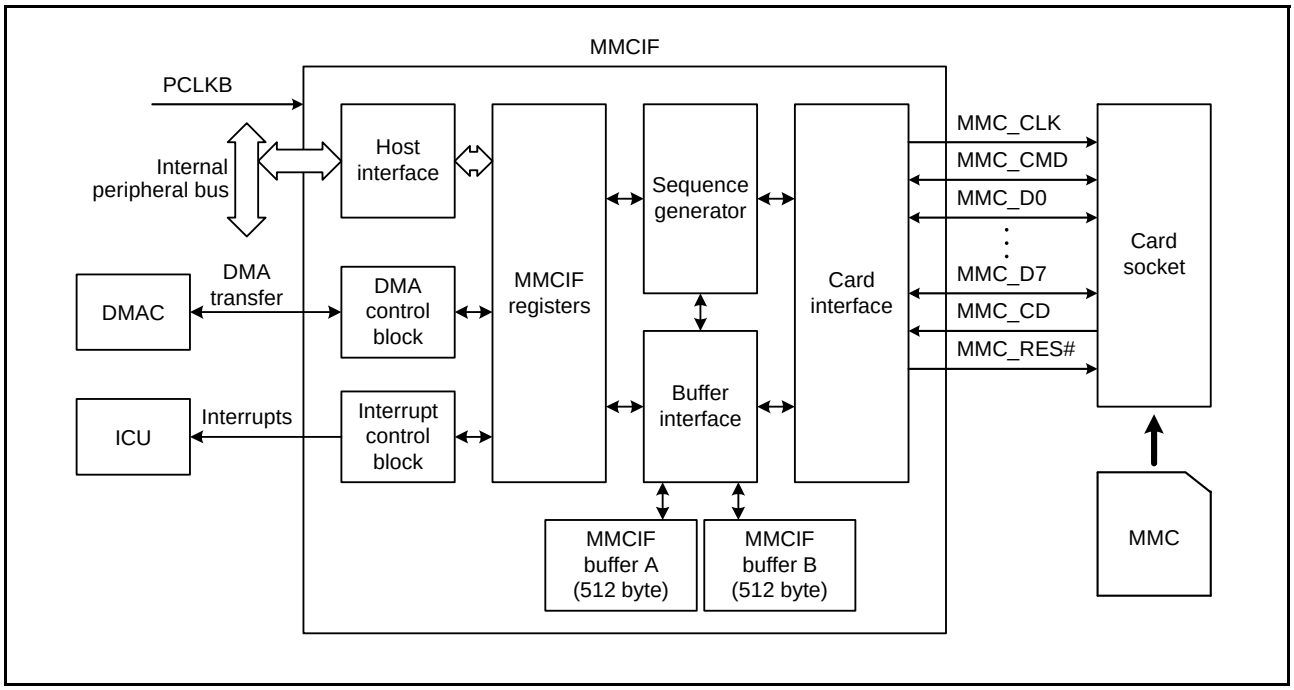


Figure 50.1 Block Diagram of the MMCIF

Table 50.2 lists the MMCIF I/O Pins.

Table 50.2 MMCIF I/O Pins

Pin Name	I/O	Description
MMC_CLK	Output	MMCIF clock
MMC_CMD	I/O	Command output and response input
MMC_D0 to MMC_D7	I/O	Transfer data (DAT0 to DAT7)
MMC_CD	Input	MMC detection *1
MMC_RES#	Output	MMC reset *2

Note 1. Check the specifications of the card socket to be used before connecting the pins to the MMC.
Note 2. Similar reset control is possible with a general purpose port.

50.2 Register Descriptions

50.2.1 Command Setting Register (CECMDSET)

Address 0008 8500h

Bit	b31	b30	b29	b28	b27	b26	b25	b24
Symbol	—	BOOT	CMD[5:0]					
Value after reset	0	0	0	0	0	0	0	0

Bit	b23	b22	b21	b20	b19	b18	b17	b16
Symbol	RTYP[1:0]		RBSY	—	WDAT	DWEN	CMLTE	CMD12EN
Value after reset	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8
Symbol	RIDXC[1:0]		RCRC7C[1:0]		—	CRC16C	BOOTACK	CRCSTE
Value after reset	0	0	0	0	0	0	0	0

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	TBIT	OPDM	—	—	SBIT	—	DATW[1:0]	
Value after reset	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0 *1	DATW[1:0]	Data Bus Width Select	b1 b0 0 0 : 1-bit data bus width 0 1 : 4-bit data bus width 1 0 : 8-bit data bus width 1 1 : Do not set this value.	R/W
b2	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b3 *2	SBIT	Read Data Start Bit Detection Setting	0: When the MMC_D*n* pins corresponding to the data bus width set in the DATW[1:0] bits are all low, this bit is detected as a start bit (*n* = 0, 0 to 3, or 0 to 7) 1: When the MMC_D0 pin is low, this bit is detected as a start bit	R/W
b5, b4	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b6	OPDM	MMC_CMD Output Select	0: Normal output from the MMC_CMD pin 1: Open-drain output from the MMC_CMD pin	R/W
b7	TBIT	Transmission Bit Setting	0: Transmission bit is set to 1 1: Transmission bit is set to 0	R/W
b8 *3	CRCSTE	CRC Status Token Reception	0: CRC status token received 1: CRC status token not received (set to 1 when CMD19 is issued)	R/W
b9	BOOTACK	Boot Acknowledge Reception	0: Boot acknowledge not received 1: Boot acknowledge received	R/W
b10 *2	CRC16C	CRC16 Check During Reception	0: Check CRC16 in the transfer data 1: Do not check CRC16 in the transfer data (set to 1 when CMD14 is issued)	R/W
b11	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b13, b12	RCRC7C[1:0]	Response CRC7 Check	b13 b12 0 0 : Performs an error check on CRC7 field value (set the RTYP[1:0] bits to 01b) 0 1 : Confirms that the field values for the response check bits are all 1 (set the RTYP[1:0] bits to 01b) 1 0 : Performs an error check on internal CRC7 field value (set the RTYP[1:0] bits to 10b) 1 1 : Error check not performed on response's CRC7 field value	R/W
b15, b14	RIDXC[1:0]	Response Index Check	b15 b14 0 0 : The command index field value checks to see if the command matches the response. 0 1 : Checks to see if the check bits in the response are all 1 1 0 : Response's command index field value is not checked 1 1 : Do not set this value.	R/W
b16	CMD12EN	CMD12 Automatic Issue Setting	0: CMD12 is not automatically issued during a multiple block transfer 1: CMD12 is automatically issued during a multiple block transfer	R/W
b17 *1	CMLTE	Transfer Type Select	0: Single block transfer 1: Multiple block transfer	R/W

Bit	Symbol	Bit Name	Description	R/W
b18 *1	DWEN	MMC Access Select	0: Read data from the MMC 1: Write data to the MMC	R/W
b19	WDAT	Data Presence Determination	0: No data present in the MMC 1: Data present in the MMC	R/W
b20	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b21	RBSY	MMC Response	0: No response busy 1: Response busy is included (R1b)	R/W
b23, b22	RTYP[1:0]	Response Type Select	b23 b22 0 0 : No response 0 1 : 6-byte response (R1, R1b, R3, R4, R5) 1 0 : 17-byte response (R2) 1 1 : Do not set this value.	R/W
b29 to b24	CMD[5:0]	Command Index	Set the command index field value.	R/W
b30	BOOT	Boot Operations	0: Command sequence when not performing boot operations 1. Command sequence when performing boot operations	R/W
b31	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W

Note 1. The DATW[1:0], CMLTE, and DWEN bit values are valid when the WDAT bit is 1.
Note 2. The SBIT and CRC16C bit values are valid when the WDAT bit is 1 and the DWEN bit is 0.
Note 3. The CRCSTE bit value is valid when bits WDAT and DWEN are both 1.

The CECMDSET register defines the command sequence. The command sequence starts when a value is written to the CECMDSET register. The CECMDSET register cannot be write accessed when the CEHOSTSTS1.CMDSEQ bit is 1. Table 50.3 lists the setting values for the CECMDSET register.

CMD12EN Bit (CMD12 Automatic Issue Setting)

Set this bit to select whether CMD12 is automatically issued or not during a multiple block transfer. When the CMD12EN bit is 1 and CMD12 automatic issuance is enabled, set the RBSY bit to 0 and set the CEBLOCKSET.BLKSIZ[15:0] bits to 200h (512 bytes). For details on CMD12 automatic issuance, refer to section 50.3.4, Automatically Issuing CMD12.

RBSY Bit (MMC Response)

Set this bit to enable or disable the busy signal when receiving a response.

Table 50.3 Setting Values for the CECMDSET Register

Command	Response	CECMDSET Register Setting Value																						Remark
		—	BOOT	CMD[5:0]	RTYP[1:0]	RBSY	—	WDAT	DWEN	CMLTE	CMD12EN	RIDXC[1:0]	RCRC7C[1:0]	—	CRC16C	BOOTACK	CRCSTE	TBIT	OPDM	—	—	SBIT	—	DATW[1:0]	
CMD0	—	0	0	000000	00	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD1	R3	0	0	000001	01	0	0	0	0	0	0	01	01	0	0	0	0	0	0	0	0	0	0	00	
CMD2	R2	0	0	000010	10	0	0	0	0	0	0	01	10	0	0	0	0	0	0	0	0	0	0	00	
CMD3	R1	0	0	000011	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD4	—	0	0	000100	00	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD5	R1b	0	0	000101	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD6	R1	0	0	000110	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	BGO
	R1b	0	0	000110	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD7	R1	0	0	000111	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
	R1b	0	0	000111	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD8	R1	0	0	001000	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD9	R2	0	0	001001	10	0	0	0	0	0	0	01	10	0	0	0	0	0	0	0	0	0	0	00	
CMD10	R2	0	0	001010	10	0	0	0	0	0	0	01	10	0	0	0	0	0	0	0	0	0	0	00	
CMD12	R1	0	0	001100	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
	R1b	0	0	001100	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD13	R1	0	0	001101	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
	R1b	0	0	001101	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	HPI
CMD14	R1	0	0	001110	01	0	0	1	0	0	0	00	00	0	1	0	0	0	0	0	0	1	0	**	
CMD15	—	0	0	001111	00	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD16	R1	0	0	010000	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD17	R1	0	0	010001	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD18	R1	0	0	010010	01	0	0	1	0	1	0	00	00	0	0	0	0	0	0	0	0	0	0	**	Pre-defined
	R1	0	0	010010	01	0	0	1	0	1	1	00	00	0	0	0	0	0	0	0	0	0	0	**	Open-ended
CMD19	R1	0	0	010011	01	0	0	1	1	0	0	00	00	0	0	0	1	0	0	0	0	0	0	**	
CMD23	R1	0	0	010111	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD24	R1	0	0	011000	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD25	R1	0	0	011001	01	0	0	1	1	1	0	00	00	0	0	0	0	0	0	0	0	0	0	**	Pre-defined
	R1	0	0	011001	01	0	0	1	1	1	1	00	00	0	0	0	0	0	0	0	0	0	0	**	Open-ended
CMD26	R1	0	0	011010	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD27	R1	0	0	011011	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD28	R1b	0	0	011100	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD29	R1b	0	0	011101	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD30	R1	0	0	011110	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD31	R1	0	0	011111	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD35	R1	0	0	100011	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD36	R1	0	0	100100	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD38	R1b	0	0	100110	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD39	R4	0	0	100111	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD40	R5	0	0	101000	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	Send CMD
	R5	0	0	101000	01	0	0	0	0	0	0	00	00	0	0	0	0	1	1	0	0	0	0	00	Send RSP
CMD42	R1	0	0	101010	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD55	R1	0	0	110111	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD56	R1	0	0	111000	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	Read
	R1	0	0	111000	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	Write
Boot operation		0	1	000000	00	0	0	1	0	1	0	00	00	0	0	*	0	0	0	0	0	0	0	**	

50.2.2 Argument Register (CEARG)

Address 0008 8508h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol																																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CEARG register defines the argument field value for the command issued. Set a value to the CEARG register before starting the command sequence. The argument field value for an automatically issued CMD12 should be set in the CEARGCMD12 register.

50.2.3 Automatically Issued CMD12 Argument Register (CEARGCMD12)

Address 0008 850Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol																																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CEARGCMD12 register defines the argument field value of an automatically issued CMD12. This register is only accessible when CMD12 is automatically issued during a multiple block transfer. Set a value to the CEARGCMD12 register before starting the command sequence. For details on CMD12 automatic issuance, refer to section 50.3.4, Automatically Issuing CMD12.

50.2.4 Command Control Register (CECMDCTRL)

Address 0008 8510h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Symbol	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BREAK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

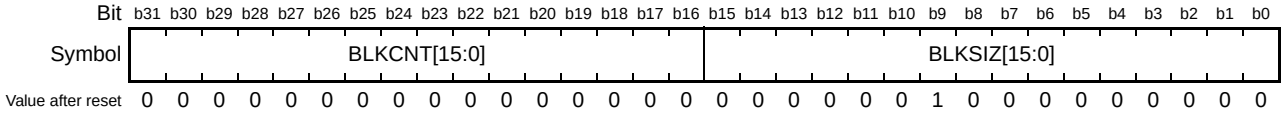
Bit	Symbol	Bit Name	Description	R/W
b0	BREAK	Command Sequence Force Stop	Writing 1 to this bit stops the command sequence. This bit will not automatically revert to 0, so after setting it to 1, set it back to 0.	R/W
b31 to b1	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

BREAK Bit (Command Sequence Force Stop)

To stop the command sequence, when the BREAK bit is 0, set it to 1 and then to 0 in succession. Then confirm that the CEHOSTSTS1.CMDSEQ flag is 0, set the CEVERSION.SWRST bit to 1, and perform an MMCIF software reset. When an MMCIF software reset is performed, registers associated with the MMCIF module become their value after reset values, so initial settings are required.

50.2.5 Transfer Block Setting Register (CEBLOCKSET)

Address 0008 8514h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	BLKSIZ[15:0]	Transfer Block Size	These bits define the transfer block size. When performing a single block transfer, values equivalent to 1 to 512 (bytes) can be set. When performing a multiple block transfer, set a value equivalent to 512 (bytes).	R/W
b31 to b16	BLKCNT[15:0]	Number of Transfer Blocks *1	These bits define the number of transfer blocks.	R/W

Note 1. These bits are accessible during a multiple block transfer.

The CEBLOCKSET register defines the block size of the data to be transferred, and the number of blocks. Set a value to the CEBLOCKSET register before starting the command sequence.

50.2.6 Clock Control Register (CECLKCTRL)

Address 0008 8518h

Bit	b31	b30	b29	b28	b27	b26	b25	b24
Symbol	MMCBUSBSY	—	—	—	—	—	—	CLKEN
Value after reset	0	0	0	0	0	0	0	0

Bit	b23	b22	b21	b20	b19	b18	b17	b16
Symbol	—	—	—	—	CLKDIV[3:0]			
Value after reset	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8
Symbol	—	—	SRSPTO[1:0]		SRBSYTO[3:0]			
Value after reset	0	0	0	0	0	0	0	0

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	SRWDTO[3:0]				—	—	—	—
Value after reset	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description		R/W
b3 to b0	—	Reserved	These bits are 0 when read. Set them to 0 when writing.		R/W
b7 to b4	SRWDTO[3:0]	Write Data Timeout/Read Data Timeout Setting	b7 b4 0 0 0 0: MMCIF × 2^{14} 0 0 0 1: MMCIF × 2^{15} 0 0 1 0: MMCIF × 2^{16} 0 0 1 1: MMCIF × 2^{17} 0 1 0 0: MMCIF × 2^{18} 0 1 0 1: MMCIF × 2^{19} 0 1 1 0: MMCIF × 2^{20} 0 1 1 1: MMCIF × 2^{21}	b7 b4 1 0 0 0: MMCIF × 2^{22} 1 0 0 1: MMCIF × 2^{23} 1 0 1 0: MMCIF × 2^{24} 1 0 1 1: MMCIF × 2^{25} 1 1 0 0: MMCIF × 2^{26} 1 1 0 1: MMCIF × 2^{27} 1 1 1 0: MMCIF × 2^{28} 1 1 1 1: MMCIF × 2^{29}	R/W
b11 to b8	SRBSYTO[3:0]	Response Busy Timeout Setting	b11 b8 0 0 0 0: MMCIF × 2^{14} 0 0 0 1: MMCIF × 2^{15} 0 0 1 0: MMCIF × 2^{16} 0 0 1 1: MMCIF × 2^{17} 0 1 0 0: MMCIF × 2^{18} 0 1 0 1: MMCIF × 2^{19} 0 1 1 0: MMCIF × 2^{20} 0 1 1 1: MMCIF × 2^{21}	b11 b8 1 0 0 0: MMCIF × 2^{22} 1 0 0 1: MMCIF × 2^{23} 1 0 1 0: MMCIF × 2^{24} 1 0 1 1: MMCIF × 2^{25} 1 1 0 0: MMCIF × 2^{26} 1 1 0 1: MMCIF × 2^{27} 1 1 1 0: MMCIF × 2^{28} 1 1 1 1: MMCIF × 2^{29}	R/W
b13, b12	SRSPTO[1:0]	Response Timeout Setting	b13 b12 0 0 : MMCIF clock × 64 0 1 : MMCIF clock × 128	b13 b12 1 0 : MMCIF clock × 256 1 1 : Do not set this value.	R/W
b15, b14	—	Reserved	These bits are 0 when read. Set them to 0 when writing.		R/W
b19 to b16	CLKDIV[3:0]	MMCIF Clock Frequency Select	b19 b16 0 0 0 0: PCLKB/2 0 0 0 1: PCLKB/2^2 0 0 1 0: PCLKB/2^3 0 0 1 1: PCLKB/2^4 0 1 0 0: PCLKB/2^5	b19 b16 0 1 0 1: PCLKB/2^6 0 1 1 0: PCLKB/2^7 0 1 1 1: PCLKB/2^8 1 0 0 0: PCLKB/2^9 1 0 0 1: PCLKB/2^{10}	R/W
			Only set the values listed above.		
b23 to b20	—	Reserved	These bits are 0 when read. Set them to 0 when writing.		R/W
b24	CLKEN	MMCIF Clock Output Control	0: MMCIF clock is not output from the MMC_CLK pin (low level output) 1: MMCIF clock is output from the MMC_CLK pin		R/W
b30 to b25	—	Reserved	These bits are 0 when read. Set them to 0 when writing.		R/W
b31	MMCBUSBSY	MMC Bus Status Flag	0: MMC bus is not busy 1: MMC bus is busy		R

Note 1. Do not rewrite the CLKEN or CLKDIV[3:0] bits when the MMCBUSBSY flag is 1.

The CECLKCTRL register controls the MMCIF clock and defines the timeout value. Do not rewrite the CECLKCTRL register during the command sequence.

MMCBUSBSY Flag (MMC Bus Status Flag)

When writing a value to the CECMDSET register and starting the command sequence, the CEHOSTSTS1.CMDSEQ flag and MMCBUSBSY flag become 1 simultaneously. When the command sequence is complete and the CEHOSTSTS1.CMDSEQ flag becomes 0, 10 cycles of the MMCIF clock are output before the MMCBUSBSY flag becomes 0.

50.2.7 Buffer Access Setting Register (CEBUFACC)

Address 0008 851Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Symbol	—	—	—	—	—	DMATYP	DMAWEN	DMAREN	—	—	—	—	—	—	—	ATYP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b16	ATYP	Buffer Access Select	0: When accessing the CEDATA register, the byte endian is not switched 1: When accessing the CEDATA register, the byte endian is switched	R/W
b23 to b17	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b24	DMAREN	Enable DMA Transfer Request to Read the CEDATA Register	0: CEDATA register cannot be read using DMA transfer 1: CEDATA register can be read using DMA transfer	R/W
b25	DMAWEN	Enable DMA Transfer Request to Write to the CEDATA Register	0: CEDATA register cannot be written to using DMA transfer 1: CEDATA register can be written to using DMA transfer	R/W
b26	DMATYP	DMA Transfer Method Select	Set this bit to 1 when the DMAREN or DMAWEN bit is 1 (enabled).	R/W
b31 to b27	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

The CEBUFACC register is used to select a method of accessing the CEDATA register, and to set the DMAC and DTC transfer method. Do not rewrite the CEBUFACC register during the command sequence. For details on the MMCIF buffer, refer to section 50.3.3, MMCIF Buffer Structure and Access Method.

50.2.8 Response Register 0 (CERESP0), Response Register 1 (CERESP1), Response Register 2 (CERESP2), Response Register 3 (CERESP3)

Addresses CERESP0 0008 852Ch, CERESP1 0008 8528h, CERESP2 0008 8524h, CERESP3 0008 8520h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol																																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Registers CERESP0 to CERESP3 are read-only registers used for storing response values from the MMC. Depending on the type of response received, the MMCIF divides and stores the responses in their appropriate registers. Table 50.4 lists the Response Value Storage Destinations. Refer to section 50.3.1, Command and Response Formats for details on the format of the response value.

Table 50.4 Response Value Storage Destinations

Response Type	CERESP3 Register	CERESP2 Register	CERESP1 Register	CERESP0 Register
R1	—	—	—	[39:8]
R1b *1	—	—	—	[39:8]
R2	[127:96]	[95:64]	[63:32]	[31:0]
R3	—	—	—	[39:8]
R4	—	—	—	[39:8]
R5	—	—	—	[39:8]

Note 1. The response for an automatically issued CMD12 is stored in the CERESPCMD12 register.

50.2.9 Automatically Issued CMD12 Response Register (CERESPCMD12)

Address 0008 8530h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol																																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CERESPCMD12 is a read-only register that stores the response from the MMC when CMD12 is automatically issued.

50.2.10 Data Register (CEDATA)

Address 0008 8534h

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol																																
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CEDATA data register is used for accessing the MMCIF buffer. Refer to section 50.3.2, Data Block Format for details on the format of the read and write data.

50.2.11 Boot Operation Setting Register (CEBOOT)

Address 0008 853Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Symbol	SBTCLKDIV[3:0]				SBTACKTO[3:0]				SFSTBTDATTO[3:0]				SBTDATTO[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description		R/W
b15 to b0	—	Reserved	These bits are 0 when read. Set them to 0 when writing.		R/W
b19 to b16	SBTDATTO[3:0]	Setting for Timeout Between Boot Data	b19 b16 0 0 0 0: 2^{14} × MMCIF clock cycle 0 0 0 1: 2^{15} × MMCIF clock cycle 0 0 1 0: 2^{16} × MMCIF clock cycle 0 0 1 1: 2^{17} × MMCIF clock cycle 0 1 0 0: 2^{18} × MMCIF clock cycle 0 1 0 1: 2^{19} × MMCIF clock cycle 0 1 1 0: 2^{20} × MMCIF clock cycle 0 1 1 1: 2^{21} × MMCIF clock cycle	b19 b16 1 0 0 0: 2^{22} × MMCIF clock cycle 1 0 0 1: 2^{23} × MMCIF clock cycle 1 0 1 0: 2^{24} × MMCIF clock cycle 1 0 1 1: 2^{25} × MMCIF clock cycle 1 1 0 0: 2^{26} × MMCIF clock cycle 1 1 0 1: 2^{27} × MMCIF clock cycle 1 1 1 0: 2^{28} × MMCIF clock cycle 1 1 1 1: 2^{29} × MMCIF clock cycle	R/W
b23 to b20	SFSTBTDATTO[3:0]	First Boot Data Timeout Setting	b23 b20 0 0 0 0: 2^{14} × MMCIF clock cycle 0 0 0 1: 2^{15} × MMCIF clock cycle 0 0 1 0: 2^{16} × MMCIF clock cycle 0 0 1 1: 2^{17} × MMCIF clock cycle 0 1 0 0: 2^{18} × MMCIF clock cycle 0 1 0 1: 2^{19} × MMCIF clock cycle 0 1 1 0: 2^{20} × MMCIF clock cycle 0 1 1 1: 2^{21} × MMCIF clock cycle	b23 b20 1 0 0 0: 2^{22} × MMCIF clock cycle 1 0 0 1: 2^{23} × MMCIF clock cycle 1 0 1 0: 2^{24} × MMCIF clock cycle 1 0 1 1: 2^{25} × MMCIF clock cycle 1 1 0 0: 2^{26} × MMCIF clock cycle 1 1 0 1: 2^{27} × MMCIF clock cycle 1 1 1 0: 2^{28} × MMCIF clock cycle 1 1 1 1: 2^{29} × MMCIF clock cycle	R/W
b27 to b24	SBTACKTO[3:0]	Boot Acknowledge Timeout Setting	b27 b24 0 0 0 0: 2^{14} × MMCIF clock cycle 0 0 0 1: 2^{15} × MMCIF clock cycle 0 0 1 0: 2^{16} × MMCIF clock cycle 0 0 1 1: 2^{17} × MMCIF clock cycle 0 1 0 0: 2^{18} × MMCIF clock cycle 0 1 0 1: 2^{19} × MMCIF clock cycle 0 1 1 0: 2^{20} × MMCIF clock cycle 0 1 1 1: 2^{21} × MMCIF clock cycle	b27 b24 1 0 0 0: 2^{22} × MMCIF clock cycle 1 0 0 1: 2^{23} × MMCIF clock cycle 1 0 1 0: 2^{24} × MMCIF clock cycle 1 0 1 1: 2^{25} × MMCIF clock cycle 1 1 0 0: 2^{26} × MMCIF clock cycle 1 1 0 1: 2^{27} × MMCIF clock cycle 1 1 1 0: 2^{28} × MMCIF clock cycle 1 1 1 1: 2^{29} × MMCIF clock cycle	R/W
b31 to b28	SBTCLKDIV[3:0]	MMCIF Clock Frequency During Boot Operations	b31 b28 0 0 0 0: PCLKB/2 0 0 0 1: PCLKB/4 0 0 1 0: PCLKB/8 0 0 1 1: PCLKB/16 Only set the values listed above.		R/W

The CEBOOT register is used for selecting the MMCIF clock frequency during boot operations, and for setting timeout values. Do not rewrite this register during a command sequence or during boot operations.

SBTCLKDIV[3:0] Bits (MMCIF Clock Frequency During Boot Operations)

The SBTCLKDIV[3:0] bit setting value must be the equal to or lower than the MMCIF clock frequency selected in the CECLKCTRL.CLKDIV[3:0] bits (the MMCIF clock frequency during boot operations must be equal to or greater than the MMCIF clock frequency in normal mode).

Refer to section 50.3.5, MMCIF Clock Frequency During Boot Operations for details on the MMCIF clock frequency during boot operation.

50.2.12 Interrupt Status Flag Register (CEINT)

Address 0008 8540h

Bit	b31	b30	b29	b28	b27	b26	b25	b24
Symbol	—	—	—	—	—	CMD12DRE	CMD12RBE	CMD12CRE
Value after reset	0	0	0	0	0	0	0	0

Bit	b23	b22	b21	b20	b19	b18	b17	b16
Symbol	DTRANE	BUFRE	BUFWEN	BUFREN	—	—	RBSYE	CRSPE
Value after reset	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8
Symbol	CMDVIO	BUFVIO	—	—	WDATERR	RDATERR	RIDXERR	RSPERR
Value after reset	0	0	0	0	0	0	0	0

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	—	—	—	CRCSTO	WDATTO	RDATTO	RBSYTO	RSPTO
Value after reset	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RSPTO	Response Timeout Flag	0: Response timeout did not occur 1: Response timeout occurred	R(/W) *1
b1	RBSYTO	Response Busy Timeout Flag	0: Response busy timeout did not occur 1: Response busy timeout occurred	R(/W) *1
b2	RDATTO	Read Data Timeout Flag	0: Read data timeout did not occur 1: Read data timeout occurred	R(/W) *1
b3	WDATTO	Write Data Timeout Flag	0: Write data timeout did not occur 1: Write data timeout occurred	R(/W) *1
b4	CRCSTO	CRC Status Timeout Flag	0: CRC status timeout did not occur 1: CRC status timeout occurred	R(/W) *1
b7 to b5	—	Reserved	These bits are 0 when read and cannot be modified.	R
b8	RSPERR	Response Error Flag	0: Response error did not occur 1: Response error occurred	R(/W) *1
b9	RIDXERR	Response Index Error Flag	0: Response index error did not occur 1: Response index error occurred	R(/W) *1
b10	RDATERR	Read Data Error Flag	0: Read data error did not occur 1: Read data error occurred	R(/W) *1
b11	WDATERR	Write Data Error Flag	0: Write data error did not occur 1: Write data error occurred	R(/W) *1
b13, b12	—	Reserved	These bits are 0 when read and cannot be modified.	R
b14	BUFVIO	MMCIF Buffer Access Error Flag	0: MMCIF buffer access error did not occur 1: MMCIF buffer access error occurred	R(/W) *1
b15	CMDVIO	Command Issue Error Flag	0: Command issuance error did not occur 1: Command issuance error occurred	R(/W) *1
b16	CRSPE	Command Response Complete Flag	0: Command transmission or response reception incomplete 1: Command transmission or response reception complete	R(/W) *1
b17	RBSYE	Response Busy Complete Flag	0: Response busy reception incomplete 1: Response busy reception complete	R(/W) *1
b19, b18	—	Reserved	These bits are 0 when read and cannot be modified.	R
b20	BUFREN	MMCIF Buffer Read Enable Flag	0: Reading data from MMCIF buffer is disabled 1: Reading data from MMCIF buffer is enabled	R(/W) *1
b21	BUFWEN	MMCIF Buffer Write Enable Flag	0: Writing data to MMCIF buffer is disabled 1: Writing data to MMCIF buffer is enabled	R(/W) *1
b22	BUFRE	MMCIF Buffer Read Complete Flag	0: Reading data from MMCIF buffer is incomplete 1: Reading data from MMCIF buffer is complete	R(/W) *1
b23	DTRANE	Data Transmit Complete Flag	0: Data transmission incomplete 1: Data transmission complete	R(/W) *1
b24	CMD12CRE	Automatically Issued CMD12 Response Complete Flag	0: CMD12 automatic issuance incomplete 1: CMD12 automatic issuance complete	R(/W) *1
b25	CMD12RBE	Automatically Issued CMD12 Response Busy Complete Flag	0: Response reception and response busy triggered by automatically issued CMD12 are incomplete 1: Response reception and response busy triggered by automatically issued CMD12 are complete	R(/W) *1

Bit	Symbol	Bit Name	Description	R/W
b26	CMD12DRE	Automatically Issued CMD12 Response Busy and MMCIF Buffer Read Complete Flag	0: Response busy and MMCIF buffer read access triggered by automatically issued CMD12 are incomplete 1: Response busy and MMCIF buffer read access triggered by automatically issued CMD12 are complete	R(/W) *1
b31 to b27	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. This flag cannot be set to 1 by the user. This flag becomes 0 when set to 0 by the user.

The CEINT register indicates the various states in the command sequence. When clearing a flag, set the flag to be cleared to 0 and set the other flags to 1. Refer to section 50.3.8, MMCIF Processing When an Error or Timeout Occurs for details on MMCIF operation when errors and timeouts occur.

RSPTO Flag (Response Timeout Flag)

This flag indicates that a response or boot acknowledge was not received within the specified period. Note that the command sequence does not stop even if the RSPTO flag becomes 1.

— When not performing a boot operation, this flag becomes 1 under the following condition:
- After a command is issued (including an automatically issued CMD12), a response is not received even after the number of clocks selected in the CECLKCTRL.SRSPTO[1:0] bits have been output.

— When performing a boot operation, this flag becomes 1 under the following condition:
- When the CECMDSET.BOOTACK bit is set to 1 (boot acknowledge received), a boot acknowledge is not received even after the number of clocks selected in the CEBOOT.SBTACKTO[3:0] bits have been output.

— This flag becomes 0 under the following condition:
- The flag is set to 0.

RBSYTO Flag (Response Busy Timeout Flag)

This flag indicates that the response is in a busy state longer than the specified period. Note that the command sequence does not stop even if the RBSYTO flag becomes 1.

— This flag becomes 1 under the following condition:
- After a command is issued (including an automatically issued CMD12), the response is in the busy state even after the number of clocks selected in the CECLKCTRL.SRBSYTO[3:0] bits have been output.

— This flag becomes 0 under the following condition:
- The flag is set to 0.

RDATTO Flag (Read Data Timeout Flag)

This flag indicates that data was not received within the specified period. Note that the command sequence does not stop even if the RDATTO flag becomes 1.

— When not performing a boot operation, this flag becomes 1 under any of the following conditions:
- After a read command is issued, data is not received even after the number of clocks selected in the CECLKCTRL.SRWDTO[3:0] bits have been output.
- After read data is received, data is not received even after the number of clocks selected in the CECLKCTRL.SRWDTO[3:0] bits have been output.

— When performing a boot operation, this flag becomes 1 under any of the following conditions:
- The first data is not received even after the number of clocks selected in the CEBOOT.SFSTBTDATTO[3:0] bits have been output.
- After read data is received, the next data is not received even after the number of clocks selected in the CEBOOT.SBTDATTO[3:0] bits have been output.

— This flag becomes 0 under the following condition:
- The flag is set to 0.

WDATTO Flag (Write Data Timeout Flag)

During a data write access, this flag indicates that the MMC was in the busy state longer than the specified period. Note that the command sequence does not stop even when the WDATTO flag becomes 1.

— This flag becomes 1 under the following condition:
- After a CRC status token is received, the MMC is in a busy state even after the number of clocks selected in the CECLKCTRL.SRWDTO[3:0] bits have been output.

— This flag becomes 0 under the following condition:
- The flag is set to 0.

CRCSTO Flag (CRC Status Timeout Flag)

This flag indicates that a CRC status token was not received. Note that the command sequence does not stop even if the CRCSTO flag becomes 1.

— This flag becomes 1 under the following condition:
- A CRC status token is not received.

— This flag becomes 0 under the following condition:
- The flag is set to 0.

RSPERR Flag (Response Error Flag)

This flag indicates that the response or boot acknowledge is in error. Note that the command sequence stops when the RSPERR flag becomes 1.

— When not performing a boot operation, this flag becomes 1 under any of the following conditions:
- The response's transmission bit is 1.
- The response's end bit is in error.
- Bits [7:1] of the response are in error (content to be checked is specified in the CECMDSET.RCRC7C[1:0] bits).

— When performing a boot operation, this flag becomes 1 under any of the following conditions:
- The bit pattern of the boot acknowledge is in error.
- The end bit of the boot acknowledge is in error.

— This flag becomes 0 under the following condition:
- The flag is set to 0.

RIDXERR Flag (Response Index Error Flag)

This flag indicates that the command index or check bits for the response is in error. Note that the command sequence stops when the RIDXERR flag becomes 1.

— This flag becomes 1 under the following condition:
- The response's command index field value or check bits field value is in error (content to be checked is specified in the CECMDSET.RIDXC[1:0] bits).

— This flag becomes 0 under the following condition:
- The flag is set to 0.

RDATERR Flag (Read Data Error Flag)

This flag indicates that the read data is in error. Note that the command sequence stops when the RDATERR flag becomes 1.

— This flag becomes 1 under any of the following conditions:
- CMD16 of the read data is in error.
- The end bit of the read data is in error.

— This flag becomes 0 under the following condition:
- The flag is set to 0.

WDATERR Flag (Write Data Error Flag)

This flag indicates that the write data is in error. Note that the command sequence stops when the WDATERR flag becomes 1.

— This flag becomes 1 under any of the following conditions:
- The status of the CRC status token is in error.
- The end bit of the CRC status token is in error.

— This flag becomes 0 under the following condition:
- The flag is set to 0.

BUFVIO Flag (MMCIF Buffer Access Error Flag)

This flag indicates illegal accesses to the MMCIF buffer. Note that the command sequence does not stop even if the BUFVIO flag becomes 1.

— This flag becomes 1 under any of the following conditions:
- The CEDATA register was accessed more times than the block size set in the CEBLOCKSET.BLKSIZ[15:0] bits.
- Despite the BUFREN bit not being set to 1 (during DMA transfer, the MMCIF buffer read DMA transfer is not requested), the CEDATA register was accessed while data was being read from the MMC.
- Despite the BUFWEN bit not being set to 1 (during DMA transfer, the MMCIF buffer write DMA transfer is not requested), the CEDATA register was accessed while data was being written to the MMC.

— This flag becomes 0 under the following condition:
- The flag is set to 0.

Note: If the MMCIF buffer is accessed after a communication error occurs, the BUFVIO flag may become 1.

CMDVIO Flag (Command Issue Error Flag)

This flag indicates that registers CECMDSET and CEBLOCKSET were set illegally.

— This flag becomes 1 under any of the following conditions:
- A value is set to the CECMDSET.CMD[5:0] bits during a command sequence (the command sequence does not automatically stop).
- A value is set to the CECMDSET.CMD[5:0] bits when one of the following is in agreement with the CECMDSET and CEBLOCKSET register setting:
  - — The RTYP[1:0] bits are 00b (no response), and the RBSY bit is 1 (response busy).
  - — The RTYP[1:0] bits are 00b, the WDAT bit is 1 (data present), and the BOOT bit is 0 (command sequence when not performing boot operations).
  - — The WDAT bit is 0 (no data present), and the CMD12EN bit is 1 (CMD12 is automatically issued).
  - — The WDAT bit is 1, the CMLTE bit is 0 (single block transfer), and the CMD12EN bit is 1.
  - — The WDAT bit is 1, the RBSY bit is 1 (response busy), and the CMD12EN bit is 1.
  - — The WDAT bit is 1, and the BLKSIZ[15:0] bits (transfer block size) are 00h.
  - — The WDAT bit is 1, and the BLKSIZ[15:0] bits are greater than 512.
  - — The WDAT bit is 1, the CMLTE bit is 1 (multiple block transfer), and the BLKCNT[15:0] bits (number of transfer blocks) are 00h.
  - — The BOOT bit is 1 (command sequence when performing boot operations), and the WDAT bit is 0 (no data present).
  - — The BOOT bit is 1, and the DWEN bit is 1 (write data to the MMC).
  - — The BOOT bit is 1, and the RBSY bit is 1.
  - — The BOOT bit is 1, and the CMD12EN bit is 1.
  - — The BOOTACK bit is 1 (boot acknowledge received), and the BOOT bit is 0 (command sequence when not performing boot operations).

— This flag becomes 0 under the following condition:
- The flag is set to 0.

CRSPE Flag (Command Response Complete Flag)

This flag indicates when command issuance or response reception is complete. The response complete flag for the automatically issued CMD12 is the CMD12CRE flag.

— When not performing a boot operation, this flag becomes 1 under any of the following conditions:
- A command is issued when no response is configured.
- A response is received when a 6-byte response and 17-byte response are set.

— When performing a boot operation, this flag becomes 1 under the following condition:
- A boot acknowledge is received while the CECMDSET.BOOTACK bit is set to 1 (boot acknowledge received).

— This flag becomes 0 under the following condition:
- The flag is set to 0.

RBSYE Flag (Response Busy Complete Flag)

This flag indicates that the response busy is complete. Note that when the RBSYE flag becomes 1, the CRSPE flag also becomes 1. The response busy complete flag for the automatically issued CMD12 is the CMD12RBE flag.

— This flag becomes 1 under the following condition:
- Response reception and response busy reception are complete.

— This flag becomes 0 under the following condition:
- The flag is set to 0.

BUFREN Flag (MMCIF Buffer Read Enable Flag)

This flag indicates that the MMCIF buffer has become read accessible. When using the CPU to read data from the CEDATA register, set the BUFREN flag to 0 before reading the amount of data set in the CEBLOCKSET.BLKSIZ[15:0] bits. If the CEBUFACC.DMAREN bit is 1, the BUFREN flag will not become 1.

— This flag becomes 1 under the following condition:
- The transfer block size amount of data is stored in the MMCIF buffer and the MMCIF buffer becomes read accessible.

— This flag becomes 0 under the following condition:
- The flag is set to 0.

BUFWEN Flag (MMCIF Buffer Write Enable Flag)

This flag indicates that the MMCIF buffer has become write accessible. When using the CPU to write data to the CEDATA register, set the BUFWEN flag to 0 before writing the amount of data set in the CEBLOCKSET.BLKSIZ[15:0] bits. If the CEBUFACC.DMAWEN bit is 1, the BUFWEN flag will not become 1.

— This flag becomes 1 under the following condition:
- The MMCIF buffer becomes empty and becomes write accessible.

— This flag becomes 0 under the following condition:
- The flag is set to 0.

BUFRE Flag (MMCIF Buffer Read Complete Flag)

This flag indicates that the read access of the MMCIF is complete.

— When not performing a boot operation, this flag becomes 1 under the following condition:
- All data in all blocks is received, the read access of the MMCIF buffer is finished, and the MMCIF buffer is empty.

— When performing a boot operation, this flag becomes 1 under the following condition:
- All data in all blocks is received, the read access of the MMCIF buffer is finished, the MMC_CMD pin changes from low to high, and the amount of time it takes 48 MMCIF clock cycles to elapse has passed.

— This flag becomes 0 under the following condition:
- The flag is set to 0.

DTRANE Flag (Data Transmit Complete Flag)

This flag indicates the all data in all blocks has been transmitted.

— This flag becomes 1 under any of the following conditions:

- When receiving a CRC status token, the busy status is complete after the CRC status token is received.
- When not receiving a CRC status token, data transmission is complete.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

CMD12CRE Flag (Automatically Issued CMD12 Response Complete Flag)

This flag indicates that the response for the automatically issued CMD12 was received.

— This flag becomes 1 under the following condition:

- A response is received for the automatically issued CMD12.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

CMD12RBE Flag (Automatically Issued CMD12 Response Busy Complete Flag)

This flag indicates that the response busy for the automatically issued CMD12 is complete. Note that when the CMD12RBE flag becomes 1, the CMD12CRE flag also becomes 1. When the CMD12RBE flag becomes 1 during a multiple block write, the DTRANE flag also becomes 1.

— This flag becomes 1 under the following condition:

- A response is received for the automatically issued CMD12, and the response busy is complete.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

CMD12DRE Flag (Automatically Issued CMD12 Response Busy and MMCIF Buffer Read Complete Flag)

This flag indicates that the response busy for the automatically issued CMD12 is complete, and the MMCIF buffer read access is complete. Note that when the CMD12DRE flag becomes 1, the CMD12RBE, CMD12CRE, and BUFRE flags also become 1.

— This flag becomes 1 under the following condition:

- The response busy for the automatically issued CMD12 is complete, and the read access from the MMCIF buffer is complete.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

50.2.13 Interrupt Request Enable Register (CEINTEN)

Address 0008 8544h

Bit	b31	b30	b29	b28	b27	b26	b25	b24
Symbol	—	—	—	—	—	MCMD12DRE	MCMD12RBE	MCMD12CRE
Value after reset	0	0	0	0	0	0	0	0

Bit	b23	b22	b21	b20	b19	b18	b17	b16
Symbol	MDTRANE	MBUFRE	MBUFWEN	MBUFREN	—	—	MRBSYE	MCRSPE
Value after reset	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8
Symbol	MCMDVIO	MBUFVIO	—	—	MWDATERR	MRDATERR	MRIDXERR	MRSPERR
Value after reset	0	0	0	0	0	0	0	0

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	—	—	—	MCRCSTO	MWDATTO	MRDATTO	MRBSYTO	MRSPTO
Value after reset	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MRSPTO	RSPTO Interrupt Request Enable	0: RSPTO flag interrupt requests disabled 1: RSPTO flag interrupt requests enabled	R/W
b1	MRBSYTO	RBSYTO Interrupt Request Enable	0: RBSYTO flag interrupt requests disabled 1: RBSYTO flag interrupt requests enabled	R/W
b2	MRDATTO	RDATTO Interrupt Request Enable	0: RDATTO flag interrupt requests disabled 1: RDATTO flag interrupt requests enabled	R/W
b3	MWDATTO	WDATTO Interrupt Request Enable	0: WDATTO flag interrupt requests disabled 1: WDATTO flag interrupt requests enabled	R/W
b4	MCRCSTO	CRCSTO Interrupt Request Enable	0: CRCSTO flag interrupt requests disabled 1: CRCSTO flag interrupt requests enabled	R/W
b7 to b5	—	Reserved	These bits are 0 when read and cannot be modified.	R
b8	MRSPERR	RSPERR Interrupt Request Enable	0: RSPERR flag interrupt requests disabled 1: RSPERR flag interrupt requests enabled	R/W
b9	MRIDXERR	RIDXERR Interrupt Request Enable	0: RIDXERR flag interrupt requests disabled 1: RIDXERR flag interrupt requests enabled	R/W
b10	MRDATERR	RDATERR Interrupt Request Enable	0: RDATERR flag interrupt requests disabled 1: RDATERR flag interrupt requests enabled	R/W
b11	MWDATERR	WDATERR Interrupt Request Enable	0: WDATERR flag interrupt requests disabled 1: WDATERR flag interrupt requests enabled	R/W
b13, b12	—	Reserved	These bits are 0 when read and cannot be modified.	R
b14	MBUFVIO	BUFVIO Interrupt Request Enable	0: BUFVIO flag interrupt requests disabled 1: BUFVIO flag interrupt requests enabled	R/W
b15	MCMDVIO	CMDVIO Interrupt Request Enable	0: CMDVIO flag interrupt requests disabled 1: CMDVIO flag interrupt requests enabled	R/W
b16	MCRSPE	CRSPE Interrupt Request Enable	0: CRSPE flag interrupt requests disabled 1: CRSPE flag interrupt requests enabled	R/W
b17	MRBSYE	RBSYE Interrupt Request Enable	0: RBSYE flag interrupt requests disabled 1: RBSYE flag interrupt requests enabled	R/W
b19, b18	—	Reserved	These bits are 0 when read and cannot be modified.	R
b20	MBUFREN	BUFREN Interrupt Request Enable	0: BUFREN flag interrupt requests disabled 1: BUFREN flag interrupt requests enabled	R/W
b21	MBUFWEN	BUFWEN Interrupt Request Enable	0: BUFWEN flag interrupt requests disabled 1: BUFWEN flag interrupt requests enabled	R/W
b22	MBUFRE	BUFRE Interrupt Request Enable	0: BUFRE flag interrupt requests disabled 1: BUFRE flag interrupt requests enabled	R/W
b23	MDTRANE	DTRANE Interrupt Request Enable	0: DTRANE flag interrupt requests disabled 1: DTRANE flag interrupt requests enabled	R/W
b24	MCMD12CRE	CMD12CRE Interrupt Request Enable	0: CMD12CRE flag interrupt requests disabled 1: CMD12CRE flag interrupt requests enabled	R/W
b25	MCMD12RBE	CMD12RBE Interrupt Request Enable	0: CMD12RBE flag interrupt requests disabled 1: CMD12RBE flag interrupt requests enabled	R/W
b26	MCMD12DRE	CMD12DRE Interrupt Request Enable	0: CMD12DRE flag interrupt requests disabled 1: CMD12DRE flag interrupt requests enabled	R/W

Bit	Symbol	Bit Name	Description	R/W
b31 to b27	—	Reserved	These bits are 0 when read and cannot be modified.	R

When the status flags in the CEINT register become 1, the CEINTEN register controls whether or not to generate an interrupt request. When bits in this register are 1 and the corresponding status flag in the CEINT register is 1, then an interrupt request is generated. Refer to section 50.5, Interrupts for details on interrupt requests.

50.2.14 Status Register 1 (CEHOSTSTS1)

Address 0008 8548h

Bit	b31	b30	b29	b28	b27	b26	b25	b24
Symbol	CMDSEQ	CMDSIG	RSPIDX[5:0]					
Value after reset	0	x	0	0	0	0	0	0

Bit	b23	b22	b21	b20	b19	b18	b17	b16
Symbol	DATSIG[7:0]							
Value after reset	x	x	x	x	x	x	x	x

Bit	b15	b14	b13	b12	b11	b10	b9	b8
Symbol	RCVBLK[15:0]							
Value after reset	0	0	0	0	0	0	0	0

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	RCVBLK[15:0]							
Value after reset	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RCVBLK[15:0]	Number of Blocks Transferred Flag	Indicates the number of blocks transferred	R
b23 to b16	DATSIG[7:0]	MMC_D[7:0] Pin Status Flag	Indicates the status of pins MMC_D7 to MMC_D0 0: Pin level is low 1: Pin level is high	R
b29 to b24	RSPIDX[5:0]	Response Index	Indicates the command index field value or the check bits field value of the response	R
b30	CMDSIG	MMC_CMD Pin Status Flag	Indicates the status of the MMC_CMD pin 0: Pin level is low 1: Pin level is high	R
b31	CMDSEQ	Command Sequence Status Flag	0: Command sequence is in its initial state 1: Command sequence in progress	R

The CEHOSTSTS1 register indicates the number of blocks that have been transferred, indicates the status of the MMC_CMD pin and pins MMC_D0 to MMC_D7, indicates the index of the response received, and indicates the status of the command sequence.

RCVBLK[15:0] Bits (Number of Blocks Transferred Flag)

These bits indicate the number of blocks that have been transferred.
When the CECMDSET.DWEN bit is 0, these bits indicate the number of blocks read from the MMC.
When the CECMDSET.DWEN bit is 1, these bits indicate the number of blocks written to the MMC.

DATSIG[7:0] Bits (MMC_D[7:0] Pin Status Flag)

These bits indicate the status of pins MMC_D0 to MMC_D7.

Note: If a communication error or timeout occurs, the MMC_D0 pin may remain low.

50.2.15 Status Register 2 (CEHOSTSTS2)

Address 0008 854Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24
Symbol	CRCSTE	CRC16E	AC12CRCE	RSPCRC7E	CRCSTEBE	RDATEBE	AC12REBE	RSPEBE
Value after reset	0	0	0	0	0	0	0	0

Bit	b23	b22	b21	b20	b19	b18	b17	b16
Symbol	AC12IDXE	RSPIDXE	BTACKPATE	BTACKEBE	—		CRCST[2:0]	
Value after reset	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8
Symbol	—	STRDATTO	DATBSYTO	CRCSTTO	AC12BSYTO	RSPBSYTO	AC12RSPTO	STRSPTO
Value after reset	0	0	0	0	0	0	0	0

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	BTACKTO	FSTBTDATTO	BTDATTO	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b5	BTDATTO	Boot Operation Timeout Flag	0: Boot operation timeout did not occur 1: Boot operation timeout occurred	R
b6	FSTBTDATTO	First Boot Data Timeout Flag	0: First boot data timeout did not occur 1: First boot data timeout occurred	R
b7	BTACKTO	Boot Acknowledge Timeout Flag	0: Boot acknowledge timeout did not occur 1: Boot acknowledge timeout occurred	R
b8	STRSPTO	Response Timeout Flag	0: Response timeout did not occur 1: Response timeout occurred	R
b9	AC12RSPTO	Automatically Issued CMD12 Response Timeout Flag	0: Automatically issued CMD12 response timeout did not occur 1: Automatically issued CMD12 response timeout occurred	R
b10	RSPBSYTO	Response Busy Timeout Flag	0: Response busy timeout did not occur 1: Response busy timeout occurred	R
b11	AC12BSYTO	Automatically Issued CMD12 Response Busy Timeout Flag	0: Automatically issued CMD12 response busy timeout did not occur 1: Automatically issued CMD12 response busy timeout occurred	R
b12	CRCSTTO	CRC Status Timeout Flag	0: CRC status timeout did not occur 1: CRC status timeout occurred	R
b13	DATBSYTO	Data Busy Timeout Flag	0: Data busy timeout did not occur 1: Data busy timeout occurred	R
b14	STRDATTO	Read Data Timeout Flag	0: Read data timeout did not occur 1: Read data timeout occurred	R
b15	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b18 to b16	CRCST[2:0]	CRC Status Indicator Flag	Indicates the CRC status	R
b19	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b20	BTACKEBE	Boot Acknowledge End Bit Error Flag	0: Boot acknowledge end bit error did not occur 1: Boot acknowledge end bit error occurred	R
b21	BTACKPATE	Boot Acknowledge Pattern Error Flag	0: Boot acknowledge patter error did not occur 1: Boot acknowledge patter error occurred	R
b22	RSPIDXE	Command Response Index Error Flag (Other Than Automatically Issued CMD12)	0: Command response index error did not occur 1: Command response index error occurred	R
b23	AC12IDXE	Automatically Issued CMD12 Response Index Error Flag	0: Automatically issued CMD12 response index error did not occur 1: Automatically issued CMD12 response index error occurred	R
b24	RSPEBE	Command Response End Bit Error Flag (Other Than Automatically Issued CMD12)	0: Command response end bit error did not occur 1: Command response end bit error occurred	R

Bit	Symbol	Bit Name	Description	R/W
b25	AC12REBE	Automatically Issued CMD12 Response End Bit Error Flag	0: Automatically issued CMD12 response end bit error did not occur 1: Automatically issued CMD12 response end bit error occurred	R
b26	RDATEBE	Read Data End Bit Error Flag	0: Read data end bit error did not occur 1: Read data end bit error occurred	R
b27	CRCSTEBE	CRC Status End Bit Error Flag	0: CRC status end bit error did not occur 1: CRC status end bit error occurred	R
b28	RSPCRC7E	Command Response CMD7 Error Flag (Other Than Automatically Issued CMD12)	0: Command response CMD7 error did not occur 1: Command response CMD7 error occurred	R
b29	AC12CRCE	Automatically Issued CMD12 Response CMD7 Error Flag	0: Automatically issued CMD12 response CMD7 error did not occur 1: Automatically issued CMD12 response CMD7 error occurred	R
b30	CRC16E	Read Data CMD16 Error Flag	0: Read data CMD16 error did not occur 1: Read data CMD16 error occurred	R
b31	CRCSTE	CRC Status Error Flag	0: CRC status error did not occur 1: CRC status error occurred	R

The CEHOSTSTS2 register indicates the status of various timeouts and errors.

BTDATTO Flag (Boot Operation Timeout Flag)

When performing boot operations, after read data is received, if the next data is not received even after the period set in the CEBOOT.SBTDATTO[3:0] bits elapses, this flag becomes 1.

FSTBTDATTO Flag (First Boot Data Timeout Flag)

When performing boot operations, if the first data is not received even after the period set in the CEBOOT.SFSTBTDATTO[3:0] bits elapses, this flag becomes 1.

BTACKTO Flag (Boot Acknowledge Timeout Flag)

When performing boot operations, if the boot acknowledge is not received even after the period set in the CEBOOT.SBTDATTO[3:0] bits elapses, this flag becomes 1.

STRSPTO Flag (Response Timeout Flag)

After issuing a command other than the automatically issued CMD12, if the response is not received even after the period set in the CECLKCTRL.SRSPTO[1:0] bits elapses, this flag becomes 1.

AC12RSPTO Flag (Automatically Issued CMD12 Response Timeout Flag)

After CMD12 is automatically issued, if the response is not received even after the period set in the CECLKCTRL.SRSPTO[1:0] bits elapses, this flag becomes 1.

RSPBSYTO Flag (Response Busy Timeout Flag)

After issuing a command other than the automatically issued CMD12, if the busy status continues even after the period set in the CECLKCTRL.SRBSYTO[3:0] bits elapses, this flag becomes 1.

AC12BSYTO Flag (Automatically Issued CMD12 Response Busy Timeout Flag)

After CMD12 is automatically issued, if the busy status continues even after the period set in the CECLKCTRL.SRBSYTO[3:0] bits elapses, this flag becomes 1.

CRCSTTO Flag (CRC Status Timeout Flag)

This flag becomes 1 if the CRC status token is not received.

DATBSYTO Flag (Data Busy Timeout Flag)

After the CRC status token is received, if the busy status continues even after the period set in the CECLKCTRL.SRWDTO[3:0] bits elapses, this flag becomes 1.

STRDATTO Flag (Read Data Timeout Flag)
After a read command is issued, if data is not received even after the period set in the CECLKCTRL.SRWDTO[3:0] bits elapses, this flag becomes 1. After data is received, if the next data is not received even after the period set in the CECLKCTRL.SRWDTO[3:0] bits elapses, this flag becomes 1.

CRCST[2:0] Flags (CRC Status Indicator Flag)
When the CECMDSET.BOOT bit is 0, these bits indicate the status of the received CRC status token.
When the CECMDSET.BOOT bit is 1, these bits indicate the bit pattern of the boot acknowledge.

BTACKEBE Flag (Boot Acknowledge End Bit Error Flag)
This flag becomes 1 if the end bit of the boot acknowledge is in error.

BTACKPATE Flag (Boot Acknowledge Pattern Error Flag)
This flag becomes 1 if the bit pattern of the boot acknowledge is in error.

RSPIDXE Flag (Command Response Index Error Flag (Other Than Automatically Issued CMD12))
If the command index field value of the response, or the check bits field value is in error, this flag becomes 1. The content of the error is defined in the CECMDSET.RIDXC[1:0] bits.

AC12IDXE Flag (Automatically Issued CMD12 Response Index Error Flag)
If the command index field value of the response for the automatically issued CMD12 is in error, this flag becomes 1. The content of the error is defined in the CECMDSET.RIDXC[1:0] bits.

RSPEBE Flag (Command Response End Bit Error Flag (Other Than Automatically Issued CMD12))
If the end bit of the response for a command other than the automatically issued CMD12 is in error, this flag becomes 1.

AC12REBE Flag (Automatically Issued CMD12 Response End Bit Error Flag)
This flag becomes 1 if the end bit of the response for the automatically issued CMD12 is in error.

RDATEBE Flag (Read Data End Bit Error Flag)
This flag becomes 1 if the end bit for the read data is in error.

CRCSTEBE Flag (CRC Status End Bit Error Flag)
This flag becomes 1 if the end bit of the CRC status token is in error.

RSPCRC7E Flag (Command Response CMD7 Error Flag (Other Than Automatically Issued CMD12))
If bits [7:1] of a response for a command other than the automatically issued CMD12 are in error, this flag becomes 1. The content of the error is stored in the CECMDSET.RCRC7C[1:0] bits.

AC12CRCE Flag (Automatically Issued CMD12 Response CMD7 Error Flag)
If bits [7:1] of the response for the automatically issued CMD12 are in error, this bit becomes 1. The content of the error is stored in the CECMDSET.RCRC7C[1:0] bits.

CRC16E Flag (Read Data CMD16 Error Flag)
This flag becomes 1 when CRC16 of the read data is in error.

CRCSTE Flag (CRC Status Error Flag)
This flag becomes 1 when the CRC status value is in error.

50.2.16 MMC Detection and Port Control Register (CEDETECT)

Address 0008 8570h

Bit	b31	b30	b29	b28	b27	b26	b25	b24
Symbol	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0

Bit	b23	b22	b21	b20	b19	b18	b17	b16
Symbol	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8
Symbol	—	CDSIG	CDRISE	CDFALL	—	—	—	—
Value after reset	0	x	0	0	0	x	0	0

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	—	—	MCDRISE	MCDFALL	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b4	MCDFALL	CDFALL Interrupt Request Enable	0: CDFALL flag interrupt request disabled 1: CDFALL flag interrupt request enabled	R/W
b5	MCDRISE	CDRISE Interrupt Request Enable	0: CDRISE flag interrupt request disabled 1: CDRISE flag interrupt request enabled	R/W
b9 to b6	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b10	—	Reserved	This bit is undefined when read. Set it to 0 when writing.	R/W
b11	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b12	CDFALL	MMC_CD Pin Falling Edge Detection Flag	0: Falling edge on the MMC_CD pin not detected 1: Falling edge on the MMC_CD pin detected	R/W *1
b13	CDRISE	MMC_CD Pin Rising Edge Detection Flag	0: Rising edge on the MMC_CD pin not detected 1: Rising edge on the MMC_CD pin detected	R/W *1
b14	CDSIG	MMC_CD Pin Status Indication Flag	This flag indicates the status of the MMC_CD pin. 0: Pin level is low 1: Pin level is high	R
b31 to b15	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. This flag cannot be set to 1 by the user. This flag becomes 0 when set to 0 by the user.

The CEDETECT register controls detection of the MMC. Refer to section 50.5, Interrupts for details on the MMC detection interrupt request.

CDFALL Flag (MMC_CD Pin Falling Edge Detection Flag)

— This flag becomes 1 under the following condition:
- The MMC_CD pin changes from high to low.

— This flag becomes 0 under the following condition:
- The flag is set to 0.

CDRISE Flag (MMC_CD Pin Rising Edge Detection Flag)

— This flag becomes 1 under the following condition:
- The MMC_CD pin changes from low to high.

— This flag becomes 0 under the following condition:
- The flag is set to 0.

50.2.17 Special Mode Setting Register (CEADDMODE)

Address 0008 8574h

Bit	b31	b30	b29	b28	b27	b26	b25	b24
Symbol	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0

Bit	b23	b22	b21	b20	b19	b18	b17	b16
Symbol	—	—	RESNOUT	—	CLKMAIN	—	—	—
Value after reset	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8
Symbol	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b18 to b0	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b19	CLKMAIN	MMCIF Internal Clock Control	0: Normal mode 1: Low power consumption mode (possible only when MMC is detected)	R/W
b20	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b21	RESNOUT	Reset Output *1	0: MMC_RES# pin is high 0: MMC_RES# pin is low	R/W
b31 to b22	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W

Note 1. The same functions can be performed using a general-use I/O port.

The CEADDMODE register controls the MMCIF internal clock.

50.2.18 Version Register (CEVERSION)

Address 0008 857Ch

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Symbol	SWRST	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Symbol	VERSION[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	VERSION[15:0]	Version Indicator	These bits indicate the version of the MMCIF.	R
b30 to b16	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b31	SWRST	MMCIF Software Reset	0: MMCIF exits the MMCIF software reset and enters normal operation 1: MMCIF is in the MMCIF software reset state	R/W

The CEVERSION register indicates the MMCIF version and controls the MMCIF software reset.

SWRST Bit (MMCIF Software Reset)

When the SWRST bit is set to 1, the MMCIF is reset, and all register values in the MMCIF become their value after reset (except for the SWRST bit).

50.3 MMCIF Operation

50.3.1 Command and Response Formats

Figure 50.2 shows the Command Format. The command index field value set in the CECMDSET.CMD[5:0] bits is reflected in bits [45:40] of the command, and the argument field value set in the CEARG register is reflected in bits [39:8] of the command.

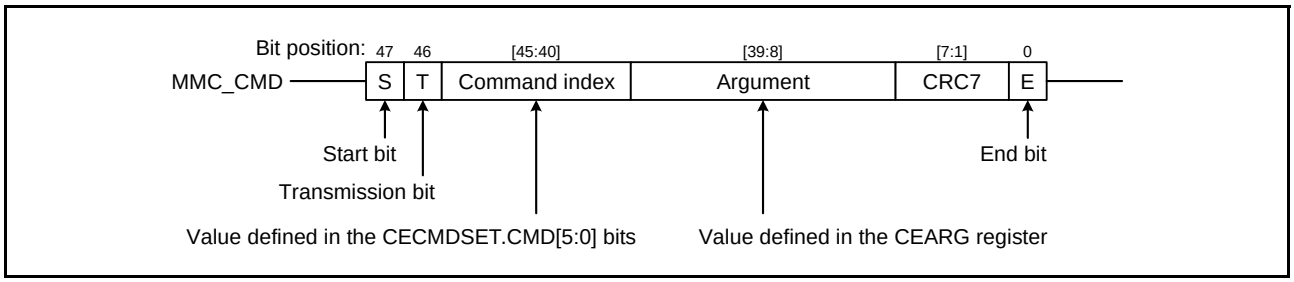


Figure 50.2 Command Format

Figure 50.3 shows the format of a 6-byte response (response other than the R2 response), and Figure 50.4 shows the format of a 17-byte response (R2 response). The command index field value or check bits field value of the response is stored in the CEHOSTSTS1.RSPIDX[5:0] flags. For a 6-byte response, bits [39:8] of the response are stored in the CERESP0 register, and for a 17-byte response, bits [127:0] of the response are stored in registers CERESP3 to CERESP0.

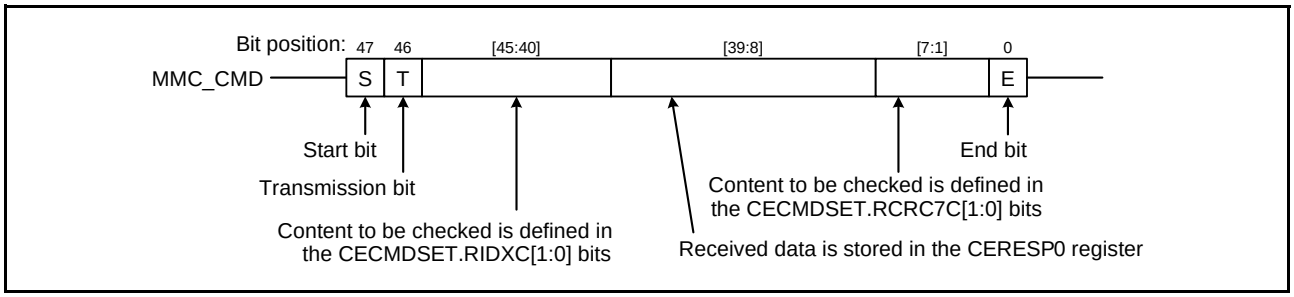


Figure 50.3 6-Byte Response Format (Response Other Than the R2 Response)

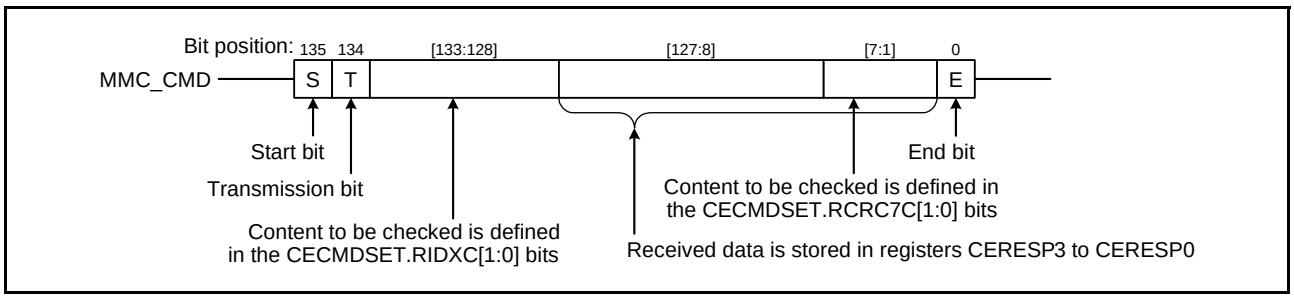


Figure 50.4 17-Byte Response Format (R2 Response)

50.3.2 Data Block Format

Figure 50.5 shows the data block format. Refer to section 50.3.3, MMCIF Buffer Structure and Access Method for details on data 0 to data 3 shown in the figure. When writing data to an MMC, data stored in the MMCIF buffer is transmitted. When reading data from an MMC, data received from the MMC is stored in the MMCIF buffer.

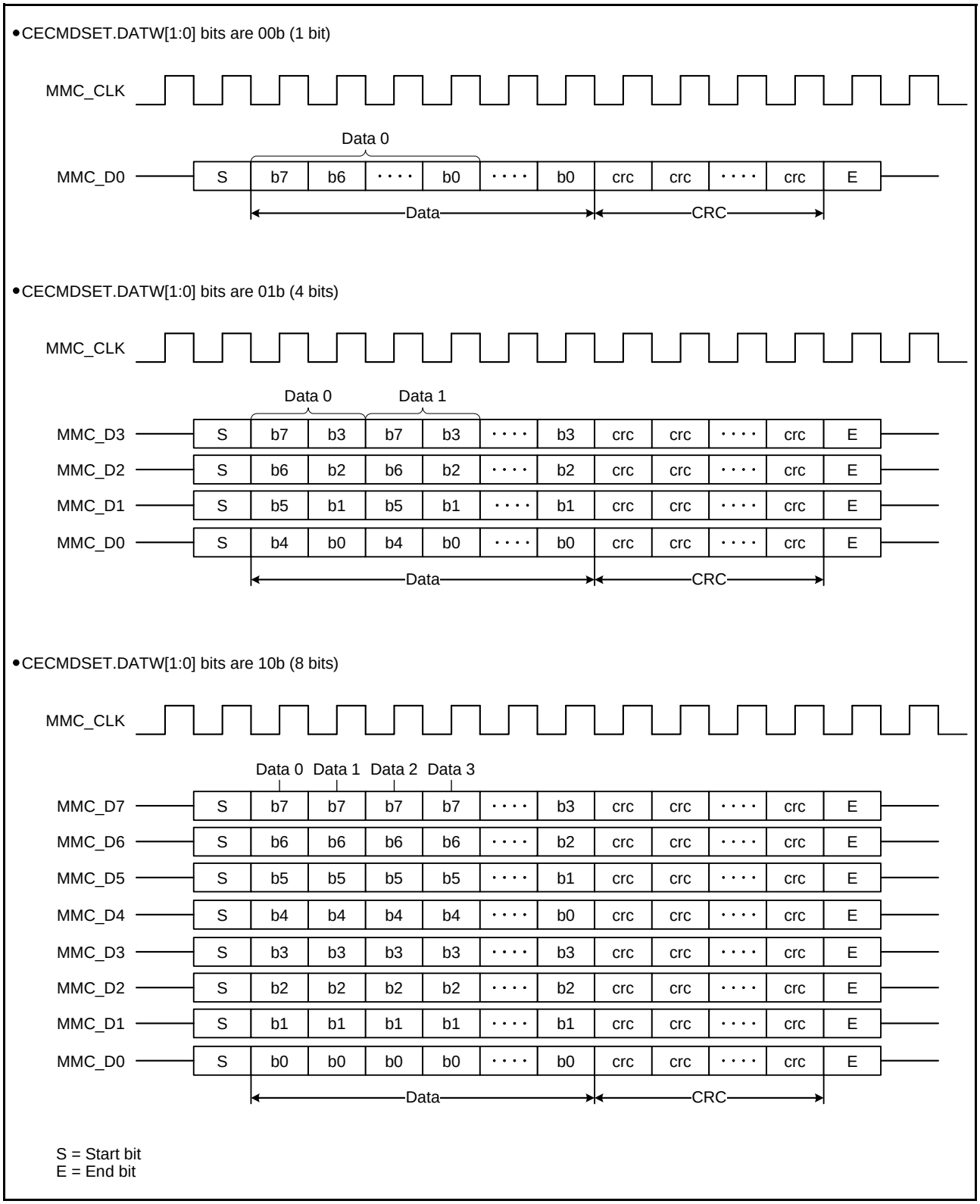


Figure 50.5 Data Block Format

50.3.3 MMCIF Buffer Structure and Access Method

Figure 50.6 shows the structure of the MMCIF buffer. The MMCIF buffer is a double buffer structure, and each buffer is 512 bytes. During a multi-block write operation, after transmitting one block of data (512 bytes) from the data stored in the MMCIF buffer, if the other buffer is full, the next block of data can be transmitted. During a multi-block read operation, even if one block of receive data (512 bytes) is stored in the MMCIF buffer, if the other MMCIF buffer is empty, the next block of receive data can be stored in the MMCIF buffer. If neither buffer is empty when performing a multi-block read operation, the MMCIF stops the MMCIF clock and pauses data reception. Then, when one of the buffers becomes empty, the MMCIF starts supplying the MMCIF clock and data reception resumes.
The CEDATA register is used to access the MMCIF buffer. If the transfer block size is set to 4*n* + 1 bytes, 4*n* + 2 bytes, or 4*n* + 3 bytes, access the CEDATA register for 4(*n* + 1) bytes (*n* = 0 to 127).

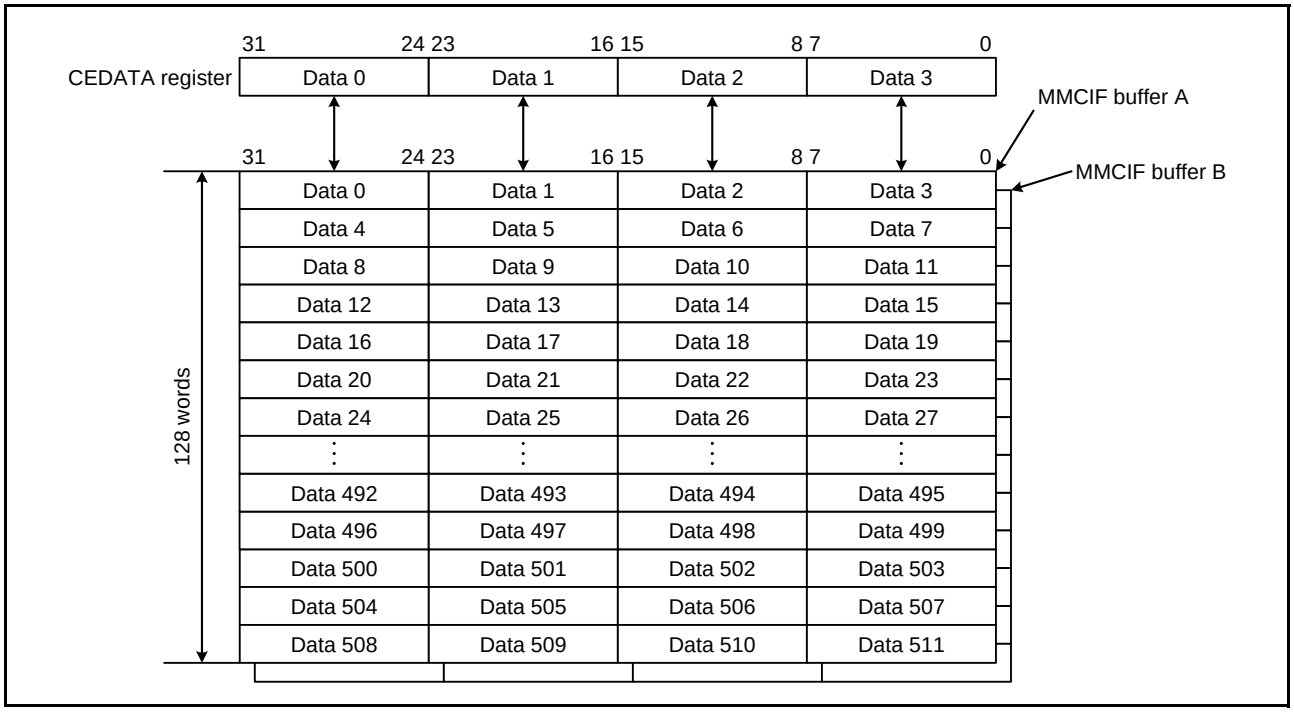


Figure 50.6 MMCIF Buffer Structure (32-Bit Access)

If the CEBUFACC.ATYP bit is set to 1, when data is written from the CEDATA register to the MMCIF buffer, or when data read from the MMCIF buffer and written to the CEDATA register, the data endian can be swapped in 1-byte units. An overview of data being swapped in 1-byte units is shown in Figure 50.7.

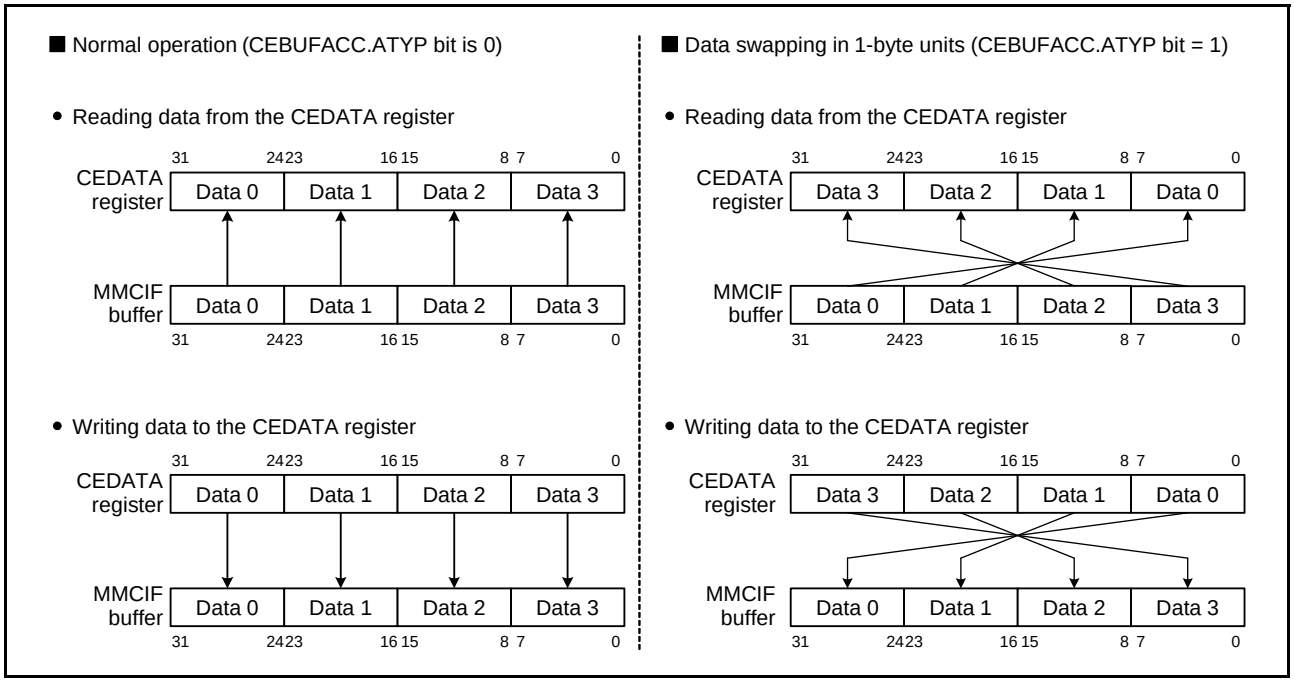


Figure 50.7 Overview of Data Swapping in 1-Byte Units (32-Bit Access)

50.3.4 Automatically Issuing CMD12

When the CECMDSET.CMD12EN bit is set to 1 and a multiple block transfer is performed, the MMCIF automatically issues CMD12. Figure 50.8 shows the Timing for Automatically Issuing CMD12 During a Multi-Block Read Operation (1-Bit Data Bus Width). CMD12 is issued such that the end bit of the command is sent 2 bits before the end bit of the data during reception of the last block.

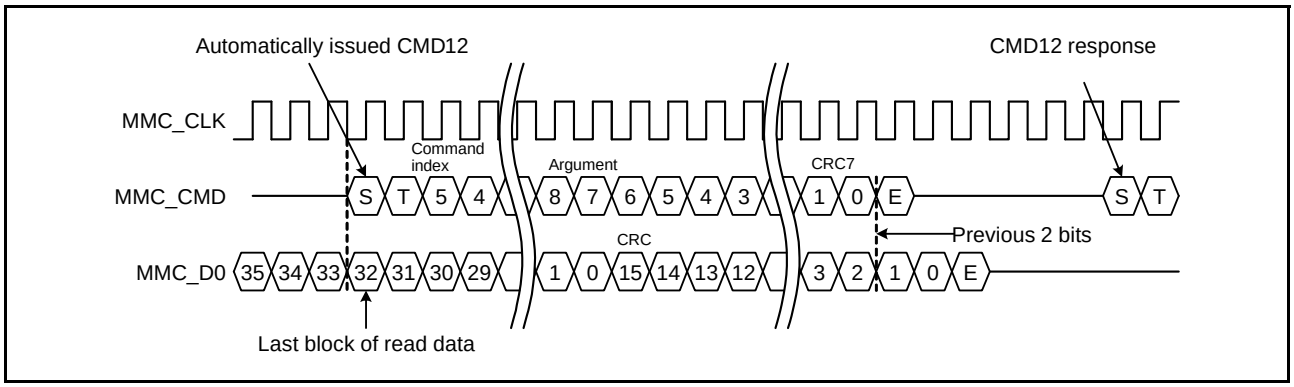


Figure 50.8 Timing for Automatically Issuing CMD12 During a Multi-Block Read Operation (1-Bit Data Bus Width)

Figure 50.9 shows the Timing for Automatically Issuing CMD12 During a Multi-Block Write Operation (1-Bit Data Bus Width). CMD12 is issued after the last block is transferred and the busy state is completed.

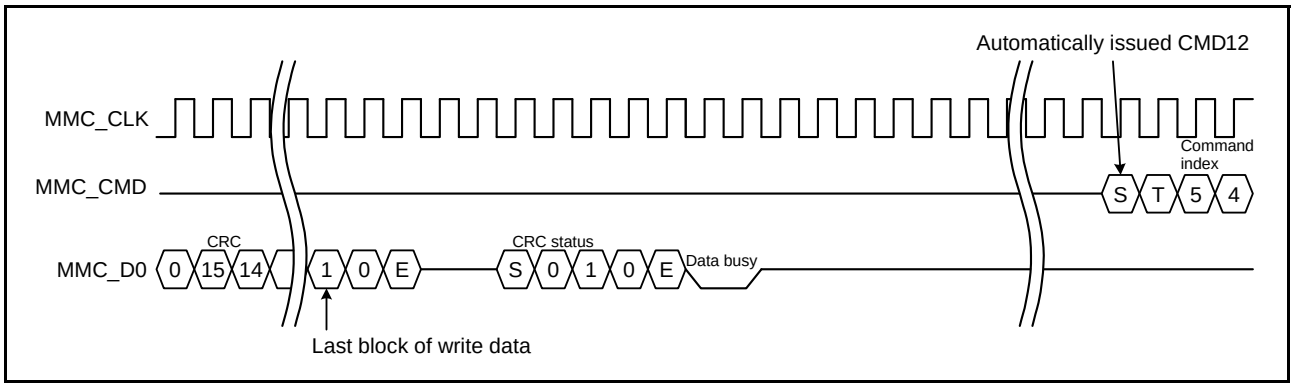


Figure 50.9 Timing for Automatically Issuing CMD12 During a Multi-Block Write Operation (1-Bit Data Bus Width)

The argument field value of the automatically issued CMD12 is set in the CEARGCMD12 register. The value for bits [39:8] of the CMD12 response are stored in the CERESPCMD12 register. Then, the response is received and the MMCIF enters the busy state.

50.3.5 MMCIF Clock Frequency During Boot Operations

Figure 50.9 shows Switching the MMCIF Clock Frequency During Boot Operations. During boot operations, when the MMC_CMD pin changes from high to low, 74 cycles of the MMCIF clock are output before the MMCIF clock frequency set in the CEBOOT.SBTCLKDIV[3:0] bits changes. After the MMC_CMD pin changes from low to high, 48 cycles of the MMCIF clock are output before the MMCIF returns to the MMCIF clock frequency set in the CECLKCTRL.CLKDIV[3:0] bits.

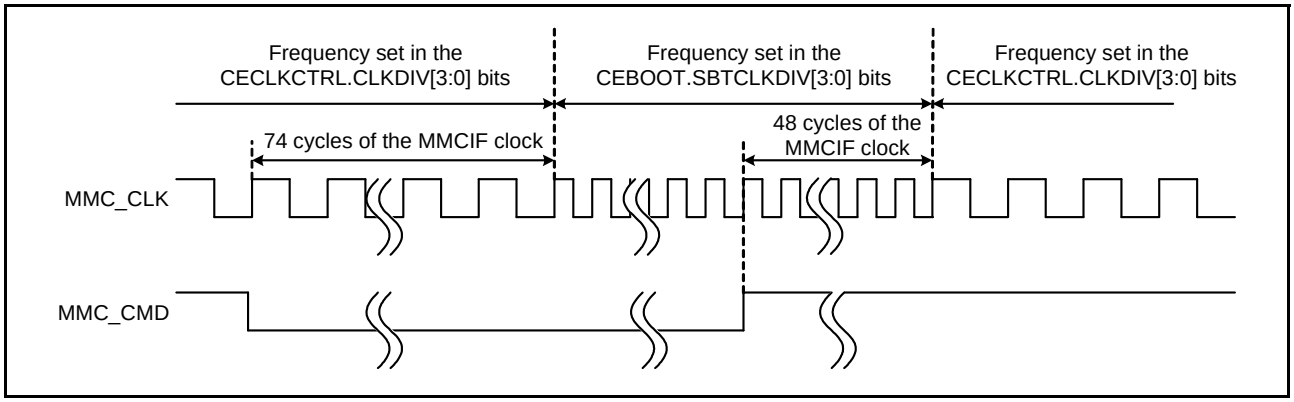


Figure 50.10 Switching the MMCIF Clock Frequency During Boot Operations

50.3.6 High Priority Interrupt (HPI)

The MMCIF supports the HPI in the sequence of CMD6, CMD24, CMD25 (pre-defined), and CMD38.
The following procedures are for HPI processing.

(1) Executing an HPI while writing to the MMC

1. Forcibly stop the command sequence.
2. Wait until the CEHOSTSTS1.CMDSEQ flag becomes 0.
3. Issue CMD12 (R1) and transition the state of the MMC from rcv to prg. If the state of the MMC is already in prg at this time, the MMC will not output a response.
4. Issue CMD13 (R1).
5. Issue the HPI command (i.e. CMD12 (R1b) or CMD13 (R1b), depending on which e.MMC is connected).

(2) Executing an HPI when not writing to the MMC an while the response is busy

1. Forcibly stop the command sequence.
2. Wait until the CEHOSTSTS1.CMDSEQ bit becomes 0.
3. Issue CMD13 (R1).
4. Issue the HPI command (i.e. CMD12 (R1b) or CMD13 (R1b), depending on which e.MMC is connected).

50.3.7 Background Operations (BGOs)

To execute BGOs, issue CMD6 (R1) and write to the BKOPS_START field in the EXT_CSD register of the MMC. To stop BGOs, issue CMD6 (R1) followed by CMD13 (R1), and either confirm that the device state is in tran, or confirm that the MMC_D0 pin is high. The HPI is used to interrupt BGOs. Refer to section 50.3.6, High Priority Interrupt (HPI) for details.

50.3.8 MMCIF Processing When an Error or Timeout Occurs

When an error occurs, the MMCIF may not stop. If an error occurs during a command sequence (the CEHOSTSTS1.CMDSEQ flag is 1), after forcibly stopping the command sequence, perform a MMCIF software reset. When an error occurs, note that transmit data and receive data stored in the MMCIF buffer is not guaranteed.
When a timeout occurs, the MMCIF will not stop. After a timeout occurs, before issuing the next command, forcibly stop the command sequence, perform an MMCIF software reset, and then issue the next command.

50.4 Examples of Issuing Commands

50.4.1 Issuing a Command Where a Response is Not Received and Data is Not Transferred

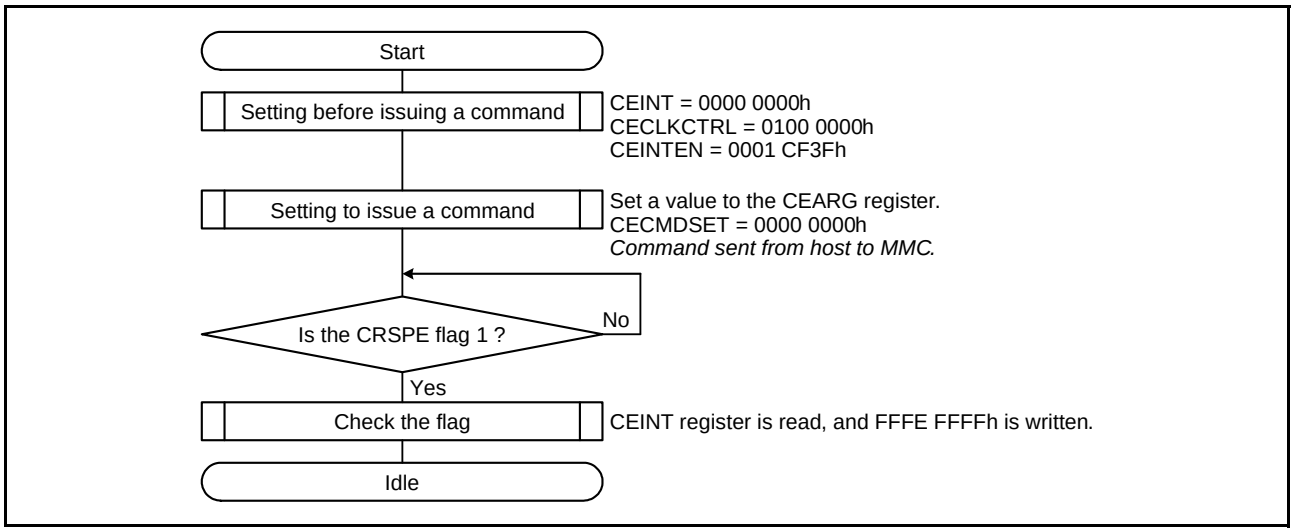


Figure 50.11 Issuing a Command Where a Response is Not Received and Data is Not Transferred (CMD0)

50.4.2 Issuing a Command Where Data is Not Transferred

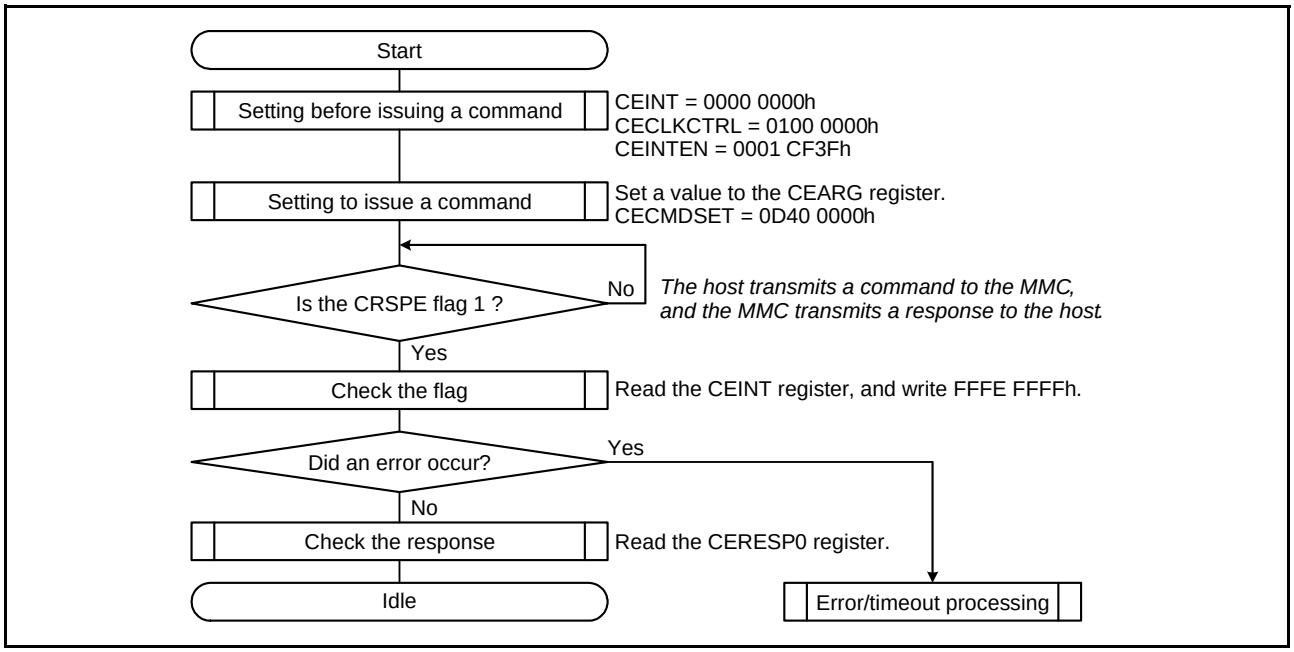


Figure 50.12 Issuing a Command Where Data is Not Transferred (CMD13)

50.4.3 Issuing a Command Where Data is Not Transferred and the Busy Signal is Used

(1) Example where the busy time is less than the period set in bits CECLKCTRL.SRBSYTO[3:0]

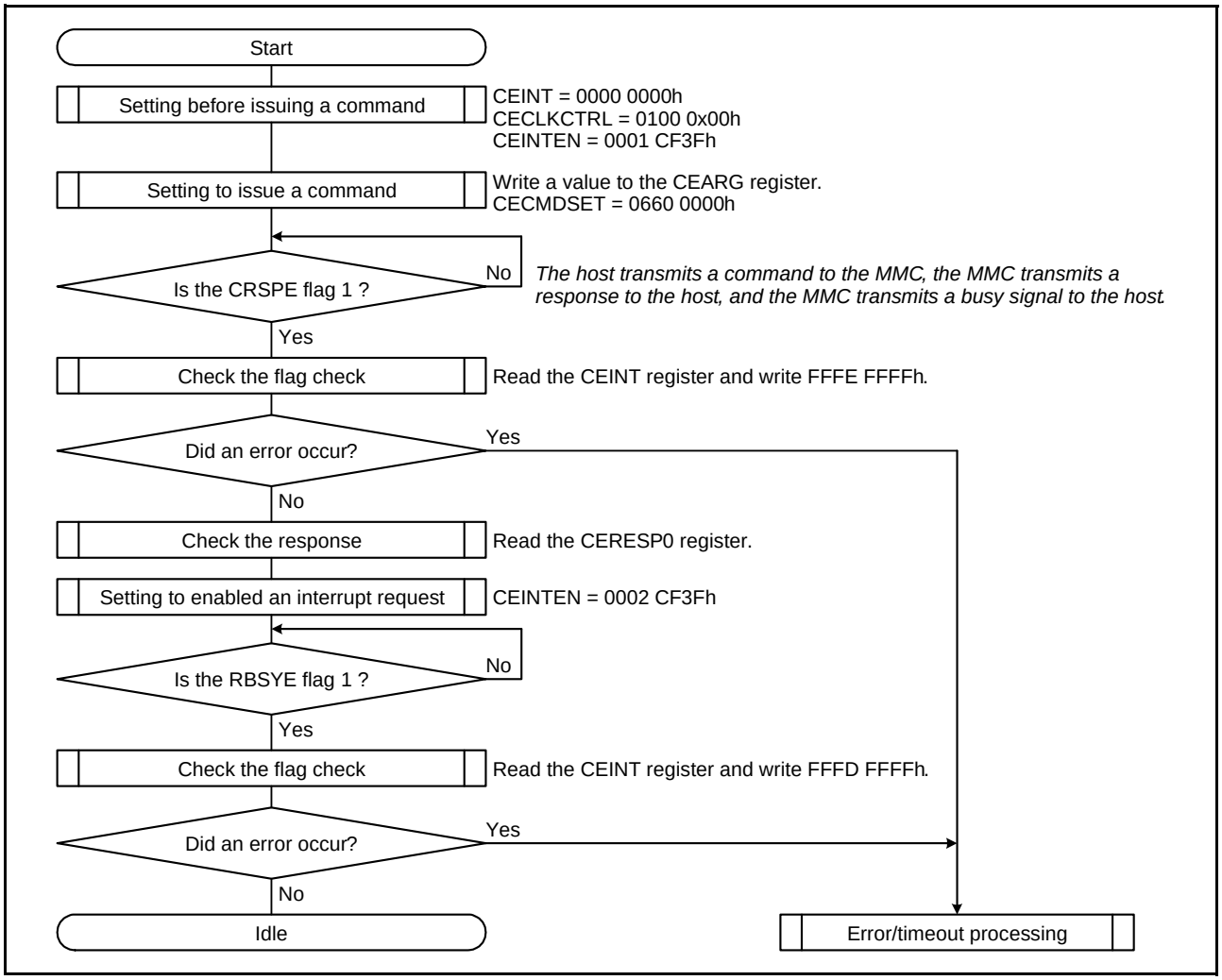
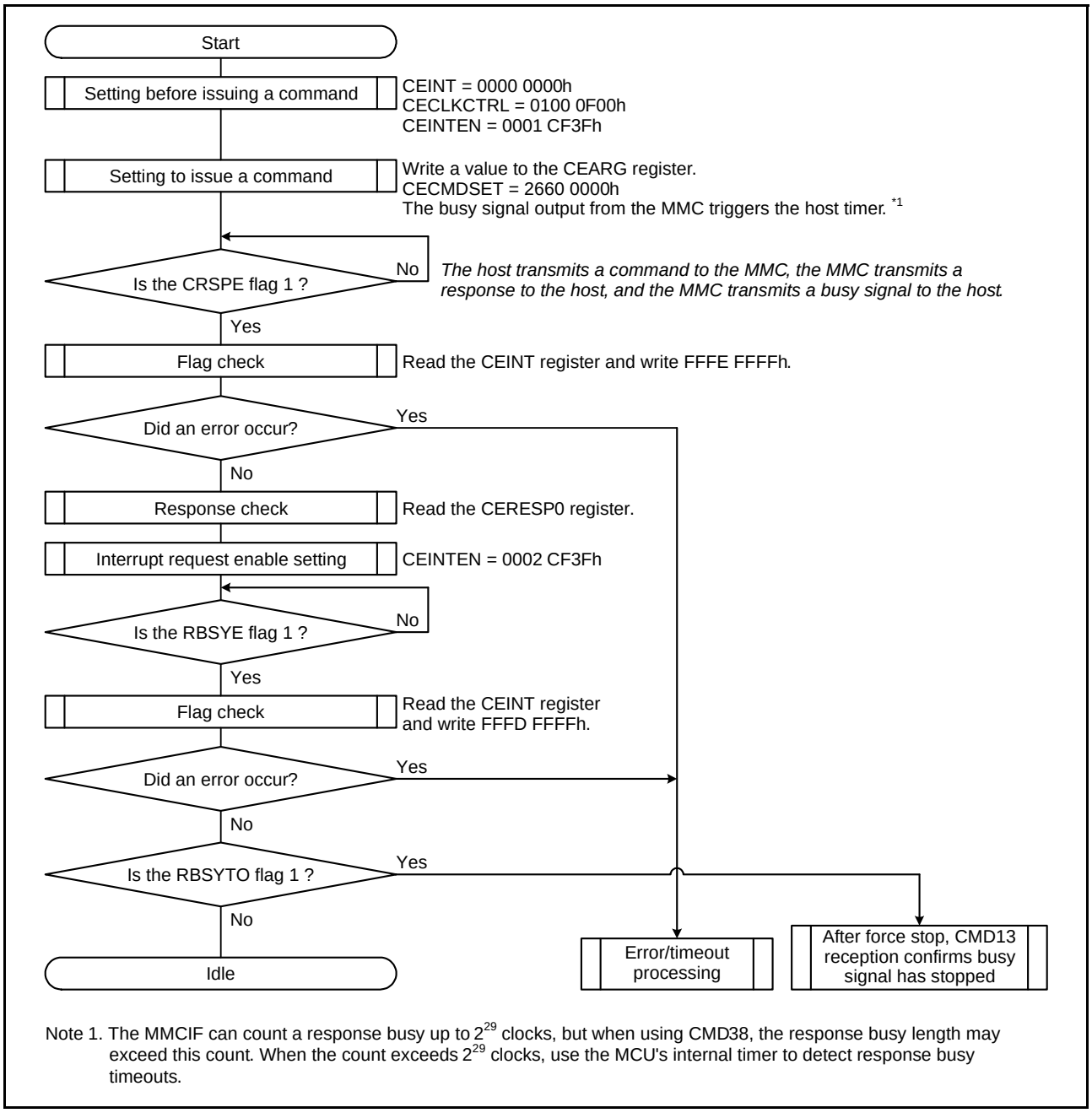


Figure 50.13 Issuing a Command Where Data is Not Transferred and the Busy Signal is Used (CMD6)

(2) Example where the busy time is equal to or more than the period set in bits CECLKCTRL.SRBSYTO[3:0]



Note 1. The MMCIF can count a response busy up to 2^{29} clocks, but when using CMD38, the response busy length may exceed this count. When the count exceeds 2^{29} clocks, use the MCU's internal timer to detect response busy timeouts.

Figure 50.14 Issuing a Command Where Data is Not Transferred and Busy Signal is Used (CMD38)

50.4.4 Issuing a Single Block Read Command (CMD17)

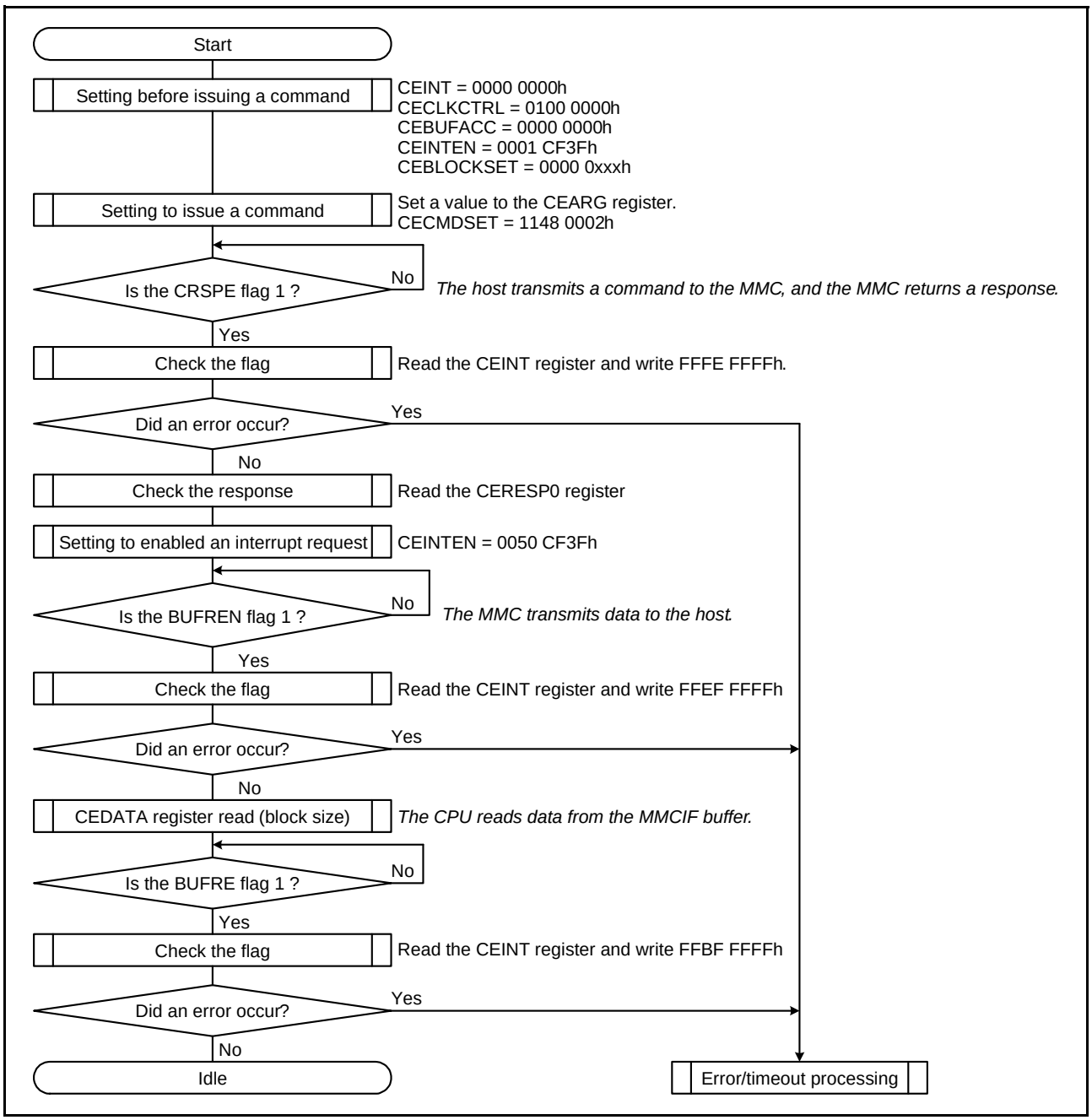


Figure 50.15 Issuing a Single Block Read Command (CMD17)

50.4.5 Issuing a Multi-Block Read Command (CMD18)

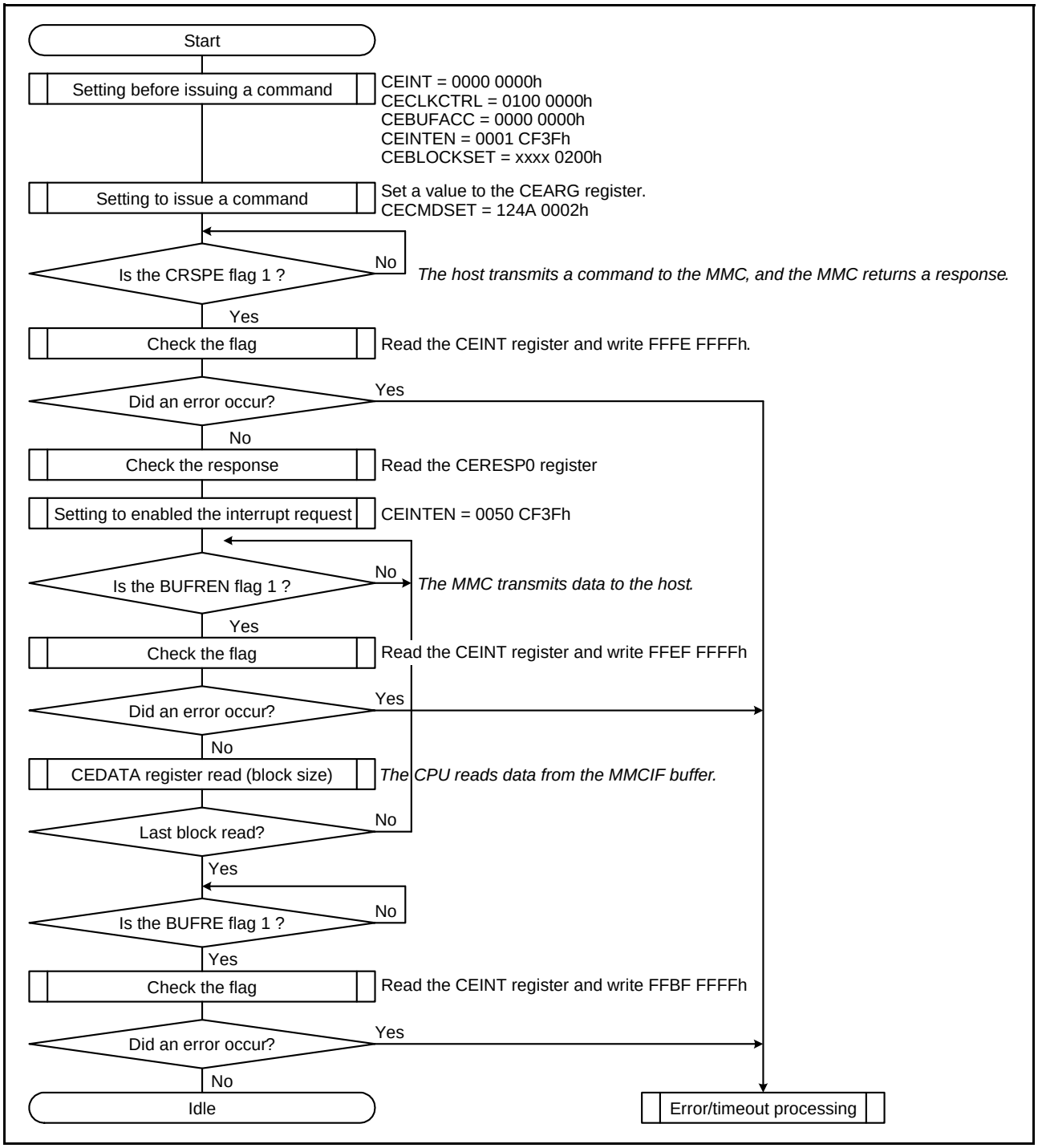


Figure 50.16 Issuing a Multi-Block Read Command (CMD18 Pre-Defined)

50.4.6 Issuing a Multi-Block Read Command (CMD18 With Automatically Issued CMD12)

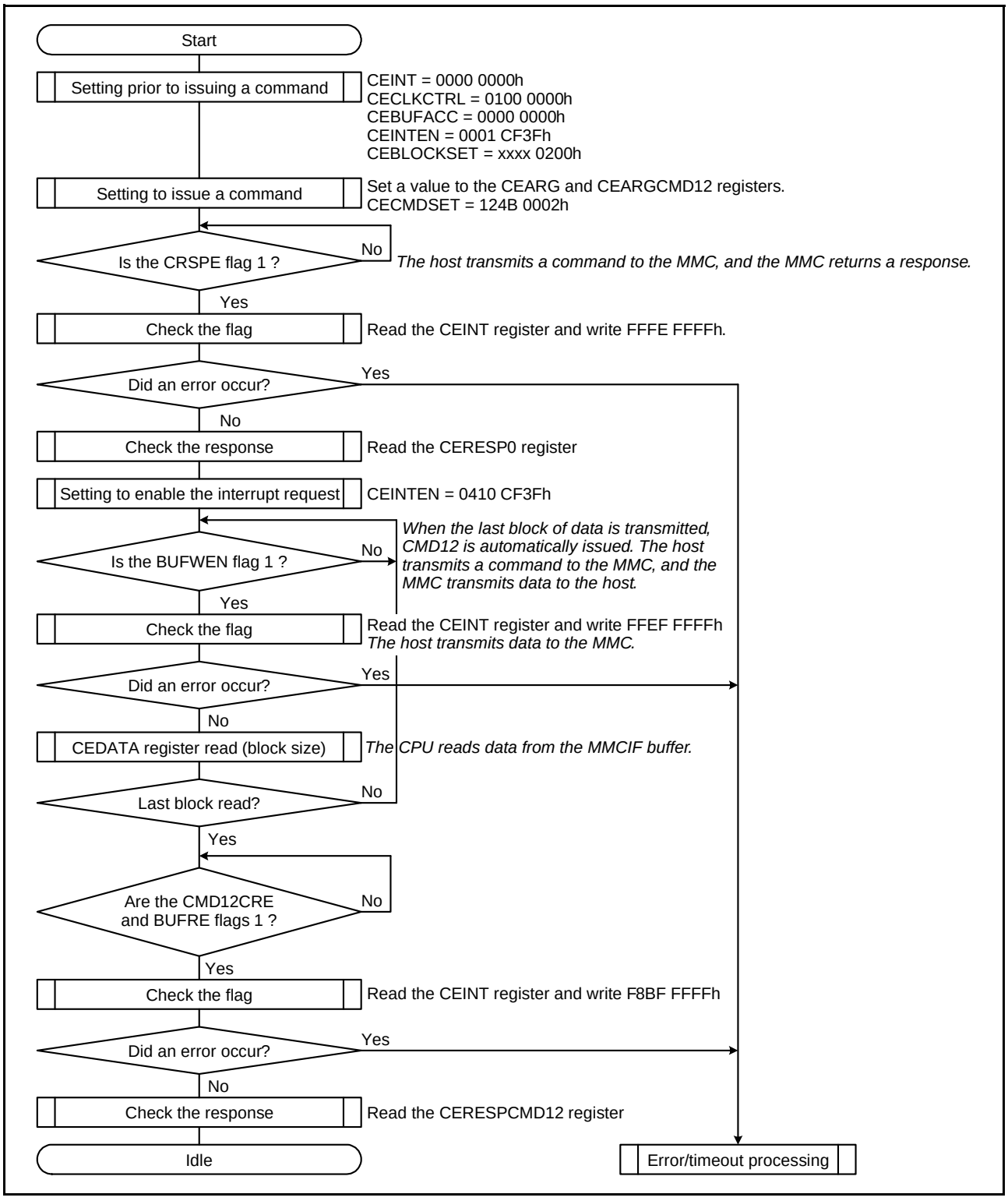


Figure 50.17 Issuing a Multi-Block Read Command (CMD18 Open-Ended)

50.4.7 Issuing a Single Block Write Command (CMD24)

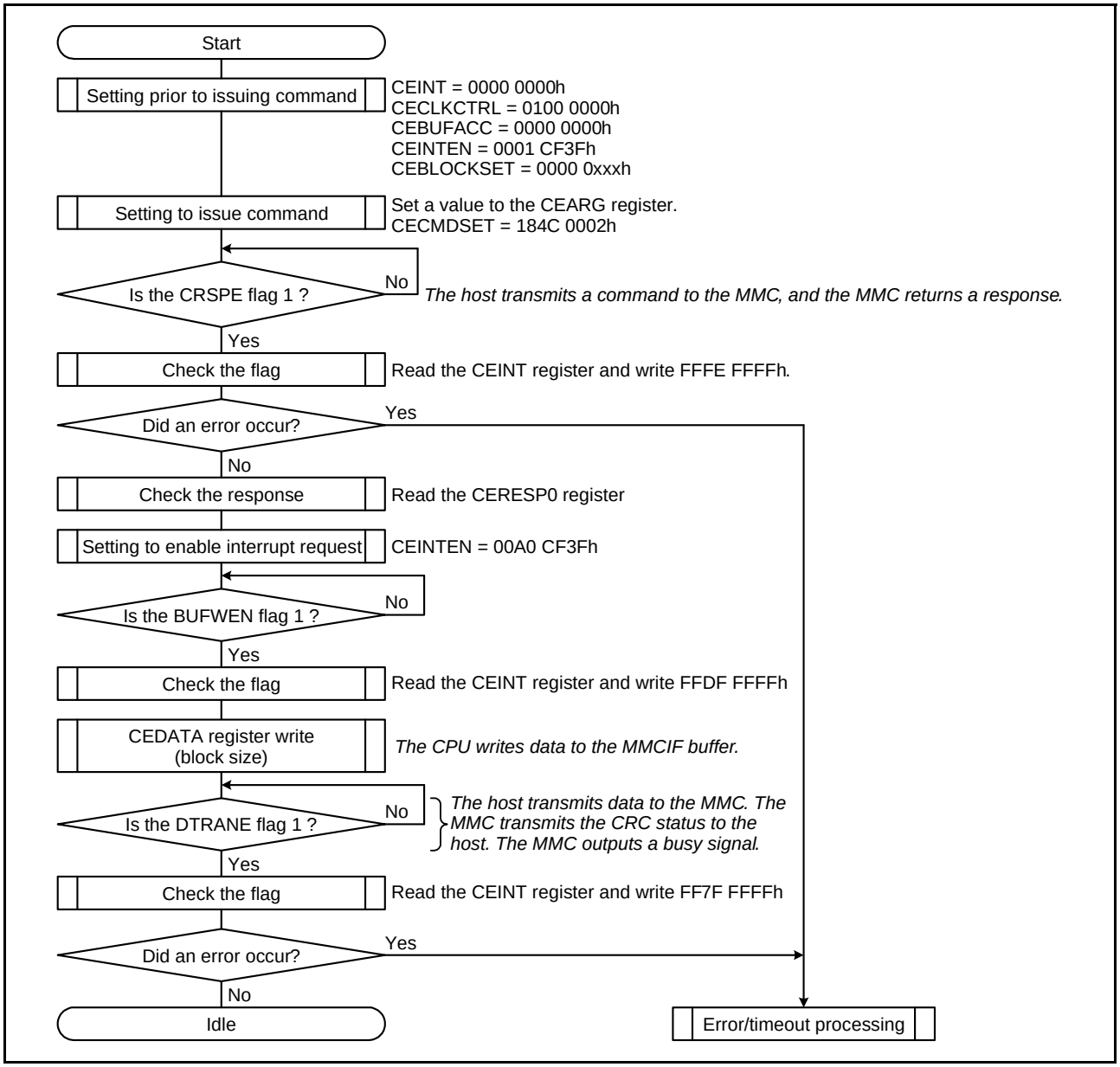


Figure 50.18 Issuing a Single Block Write Command (CMD24)

50.4.8 Issuing a Multi-Block Write Command (CMD25)

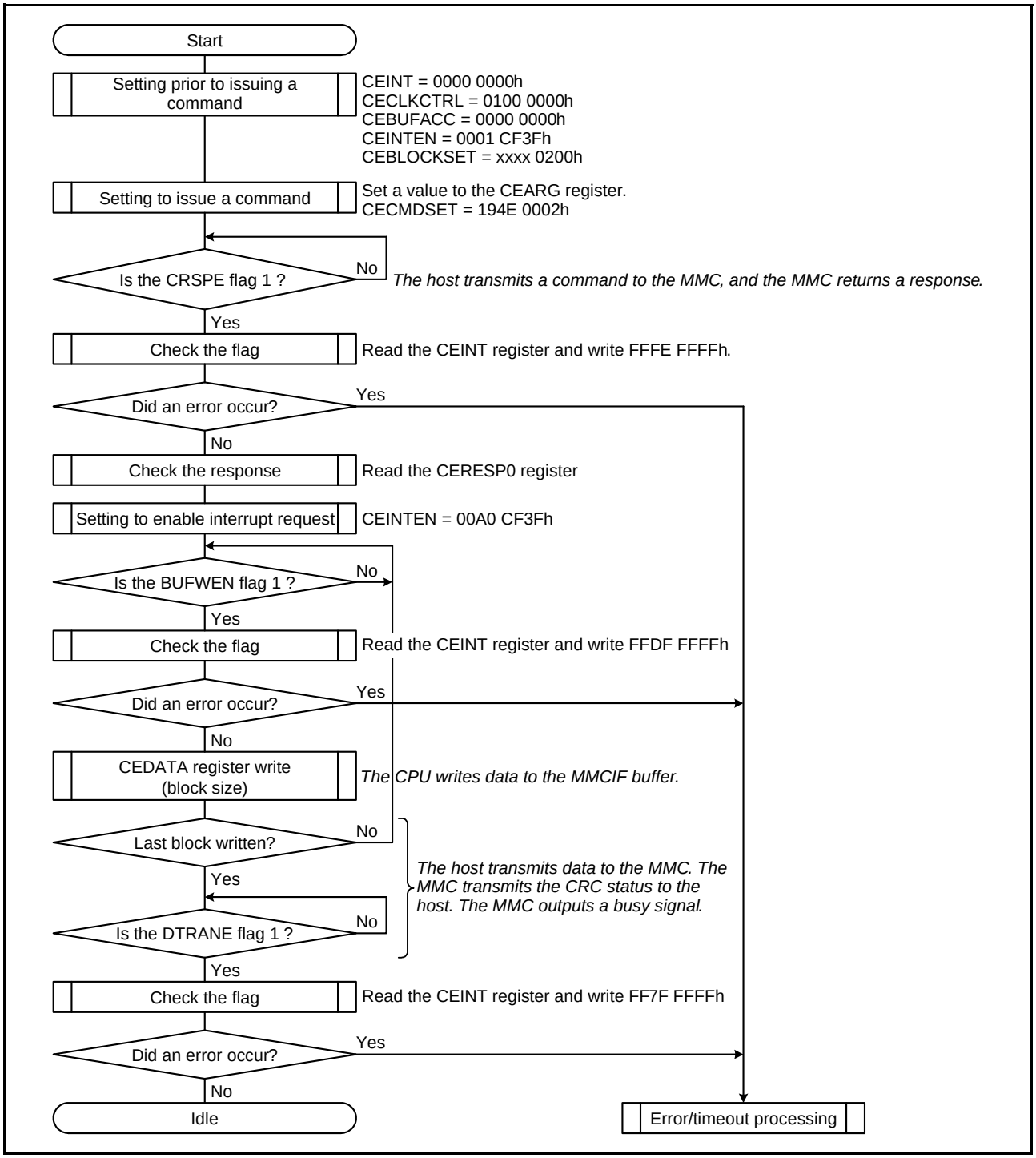


Figure 50.19 Issuing a Multi-Block Write Command (CMD25 Pre-Defined)

50.4.9 Issuing a Multi-Block Write Command (CMD25 With Automatically Issued CMD12)

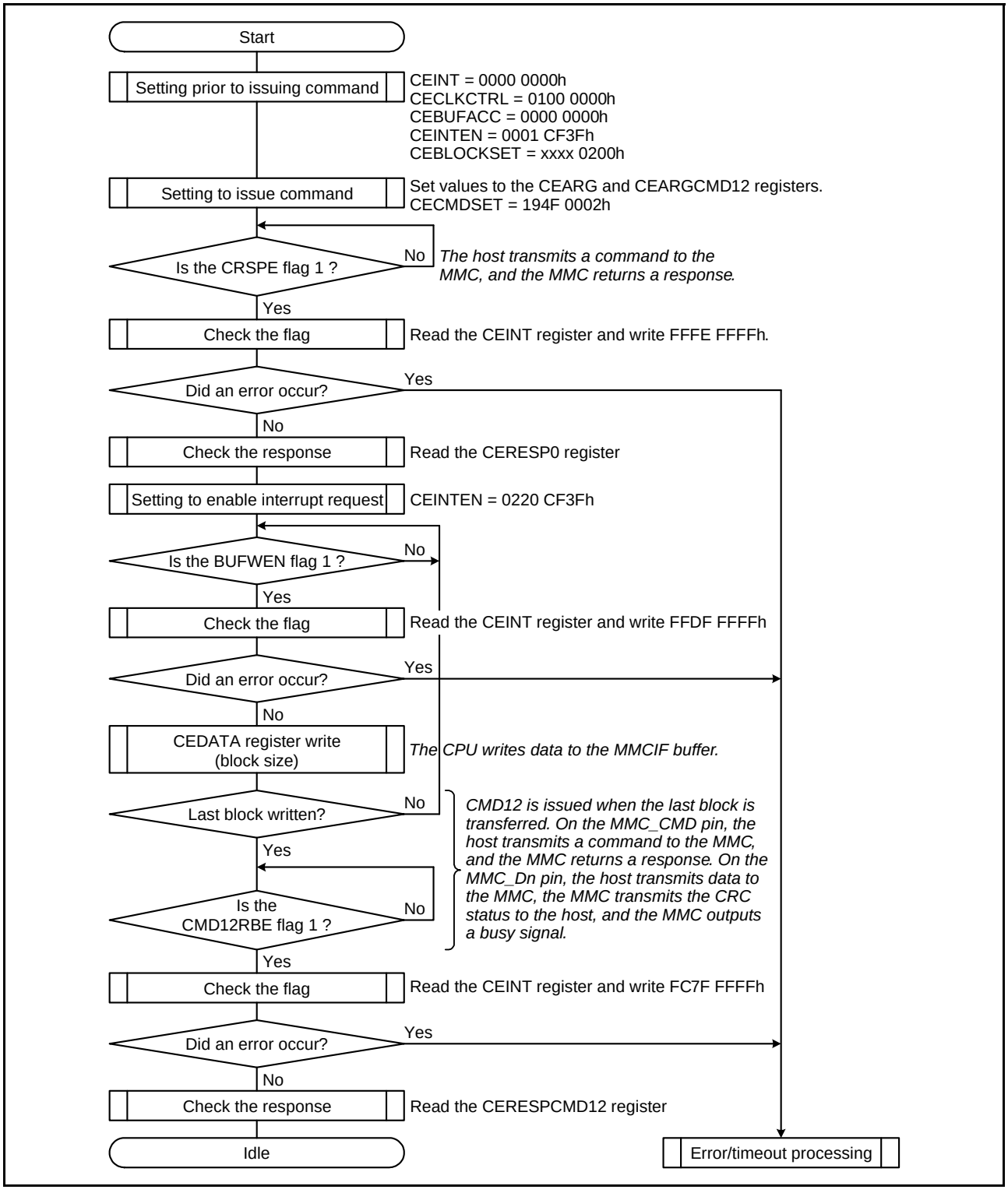


Figure 50.20 Issuing a Multi-Block Write Command (CMD25 Open-Ended)

50.4.10 Boot Operations

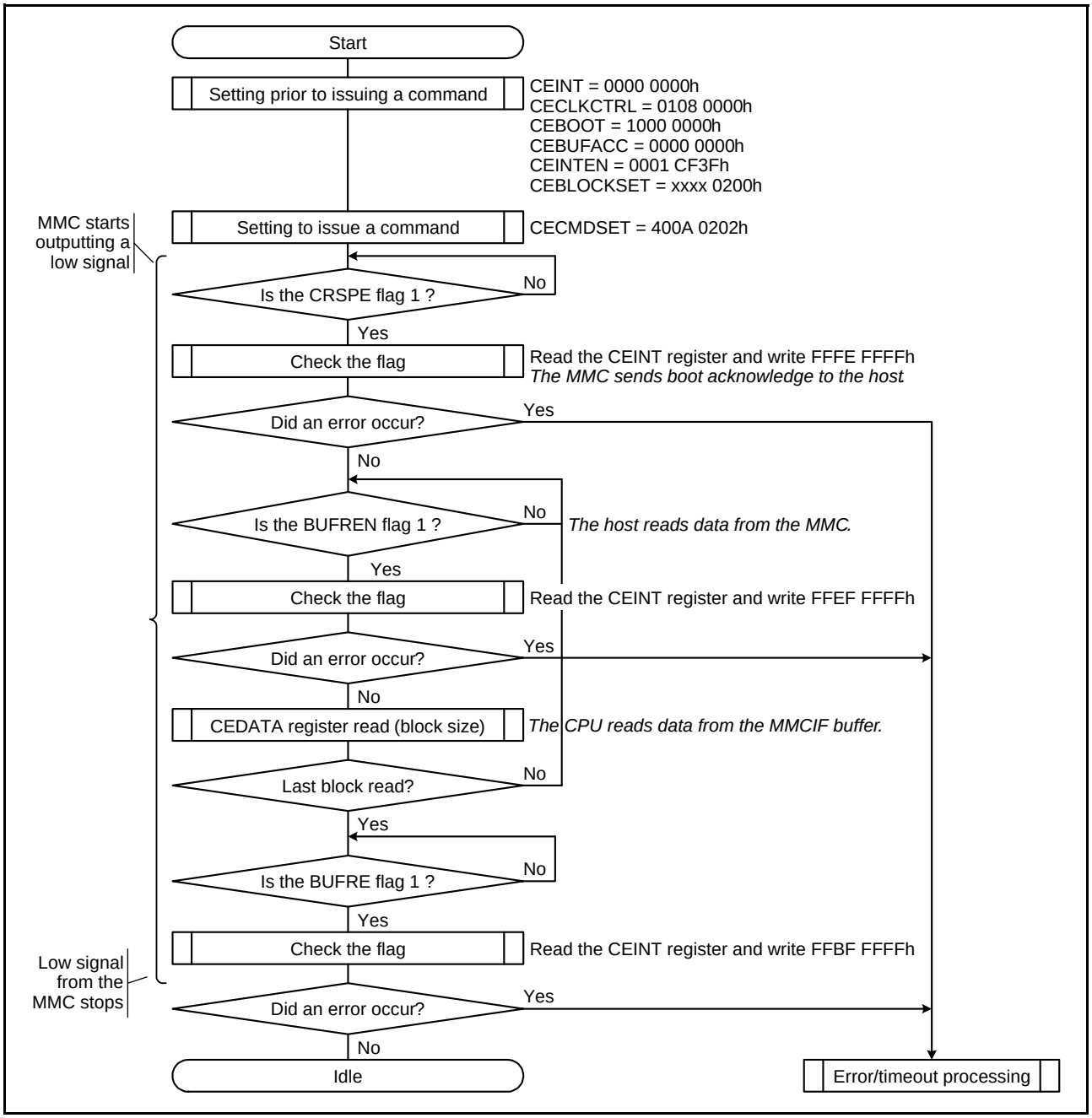


Figure 50.21 Boot Operation With Boot Acknowledge

50.4.11 Command Sequence Force Stop

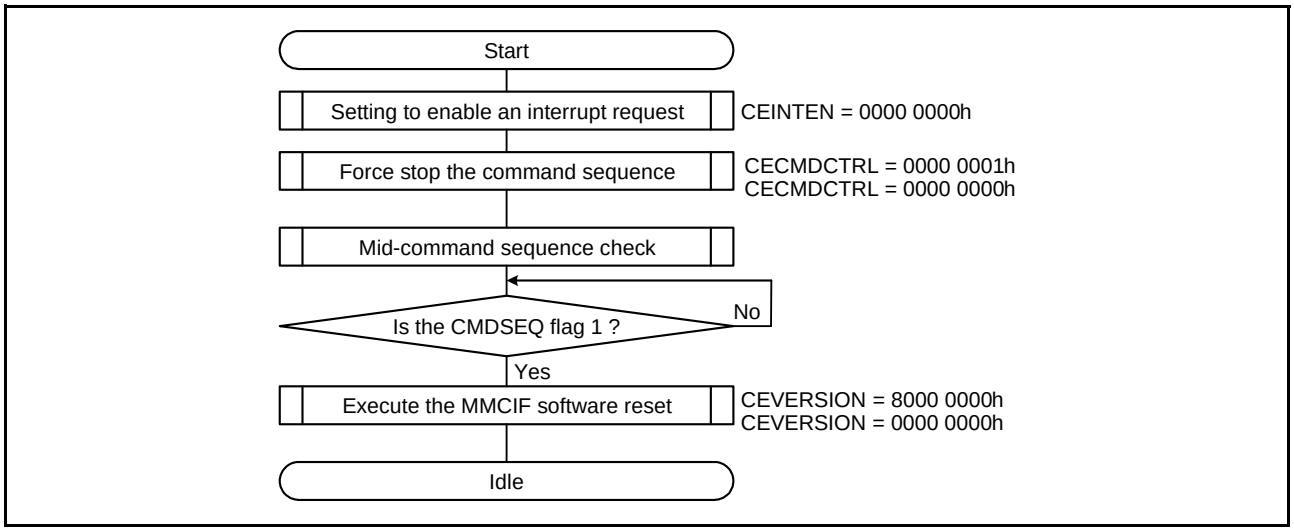


Figure 50.22 Command Sequence Force Stop

50.5 Interrupts

Table 50.5 lists the MMCIF Interrupt Sources. The MMCIF generates four types of interrupt requests: normal operation interrupt, error/timeout interrupt, MMC detection interrupt, and the MMCIF buffer access interrupt. When an interrupt status flag in the CEINT register becomes 1, if the corresponding bit in the CEINTEN register is 1, the MMCIF interrupt request is generated.

Table 50.5 MMCIF Interrupt Sources

Interrupt Source	Interrupt Status Flags		Interrupt Request Enable Register		DMAC/DTC Triggerable
	Register symbol	Bit symbol	Register symbol	Bit symbol	
Normal operation interrupt (ACCIO)	CEINT	CMD12DRE	CEINTEN	MCMD12DRE	No
		CMD12RBE		MCMD12RBE	
		CMD12CRE		MCMD12CRE	
		DTRANE		MDTRANE	
		BUFRE		MBUFRE	
		BUFWEN		MBUFWEN	
		BUFREN		MBUFREN	
		RBSYE		MRBSYE	
		CRSPE		MCRSPE	
Error/timeout interrupt (ERRIO)		CMDVIO		MCMDVIO	No
		BUFVIO		MBUFVIO	
		WDATERR		MWDATERR	
		RDATERR		MRDATERR	
		RIDXERR		MRIDXERR	
		RSPERR		MRSPERR	
		CRCSTO		MCRCSTO	
		WDATTO		MWDATTO	
		RDATTO		MRDATTO	
		RBSYTO		MRBSYTO	
		RSPTO		MRSPTO	
MMC detection interrupt (CDETIO)	CEDETECT	CDRISE	CEDETECT	MCDRISE	No
		CDFALL		MCDFALL	
MMCIF buffer access interrupt (MBFAI)	—	—	CEBUFACC	DMAREN (when reading)	Yes
				DMAWEN (when writing)	

50.5.1 DMA Transfer Interrupt Requests

The MBFAI interrupt can be used to trigger DMA transfer for writing data to and reading data from the CEDATA register. When the CEBUFACC.DMAWEN bit is set to 1, if the MMCIF buffer is empty, the MBFAI interrupt request is generated. This interrupt request triggers the DMAC or DTC, and data can be transferred to the MMCIF buffer. At this point, the CEINT.BUFWEN flag does not become 1. When the CEBUFACC.DMAREN bit is set to 1, if the amount of data in the MMCIF buffer is the size specified in the CEBLOCKSET.BLKSIZ[15:0] bits, the MBFAI interrupt request is generated. This interrupt request triggers the DMAC or DTC, and data can be transferred from the MMCIF buffer. At this point, the CEINT.BUFREN flag does not become 1.

The MBFAI interrupt request is asserted for (block size set in the CEBLOCKSET.BLKSIZ[15:0] bits × number of blocks set in the CEBLOCKSET.BLKCNT[15:0] bits), and when the last block is transferred, it is negated. During DMA transfer, if an error occurs or if the command sequence is forcibly ended, the command sequence is stopped, so the MBFAI interrupt request is canceled.

Table 50.6 lists the DMAC and DTC Settings When Performing DMA Transfer.

Table 50.6 DMAC and DTC Settings When Performing DMA Transfer

Item		Settings
Transfer mode		Block transfer mode
Transfer data	1 data	32 bits
	Block size	Size set in the CEBLOCKSET.BLKSIZ[15:0] bits divided by 4
Block transfer count		Number of times set in the CEBLOCKSET.BLKCNT[15:0] bits

50.6 Notes On Using the MMCIF

50.6.1 MMC Detection

Insertion and removal of an MMC can be detected by monitoring the CEDETECT.CDRISE flag and CDFALL flag. However, as the MMCIF does not include hardware to eliminate chattering that occurs when the MMC is inserted and removed, chattering must be eliminated with software.

50.6.2 Multiple Block Transfer

Using a pre-defined multiple block transfer is recommended to increase data transfer rates.

50.6.3 Module-Stop Function Settings

Set the MSTPCRD.MSTPD21 bit to enable and disable MMCIF operation. MMCIF operation is stopped after a reset. Registers in the MMCIF can be accessed by exiting the module-stop state. Refer to section 11, Low Power Consumption for details.

51. Parallel Data Capture Unit (PDC)

51.1 Overview

This MCU includes a single parallel data capture unit (PDC).
The PDC has the function of communicating with the external I/O devices including image sensors, and specifically transferring parallel data such as an image output from the external I/O devices via the DTC or DMAC to the on-chip RAM and external address spaces (the CS and SDRAM areas).

Table 51.1 PDC Specifications

Item	Description
Capture Range	• Desired amounts of parallel data within the following ranges in the vertical and horizontal directions. Vertical direction: 1 to 4095 lines Horizontal direction: 4 to 4095 bytes
Parallel data transfer clock (PIXCLK)	• Operating frequency: 1 to 27 MHz*1
Interrupt Sources	• Receive data ready • Frame end • Overrun • Underrun • Error in the setting for the number of lines • Error in setting for the number of bytes per line
Startup of DTC/DMAC	• The receive data ready interrupt is capable of starting the DTC or DMAC
Parallel data transfer clock output (PCKO)	• Operating frequency: 1 to 30 MHz*2 • Clock source: Peripheral module clock B (PCLKB) • Frequency division ratio: Selectable from 2, 4, 6, 8, 10, 12, 14, and 16.
Others	• PDC reset function • Selectable active sense for the VSYNC and HSYNC signals • Monitoring of the VSYNC and HSYNC signals • Endian selection
Reducing Power Consumption	• The PDC can be placed in the module stop state.
Internal Bus Interface	• The PDC is connected with internal peripheral bus 3

Note 1. The frequency of the parallel data transfer clock (PIXCLK) should be equal to or less than 0.6 × PCLKB (peripheral module clock).
Note 2. The operating frequency is 30 MHz when that of the peripheral module clock B (PCLKB) is 60 MHz and the frequency division ratio is 2.

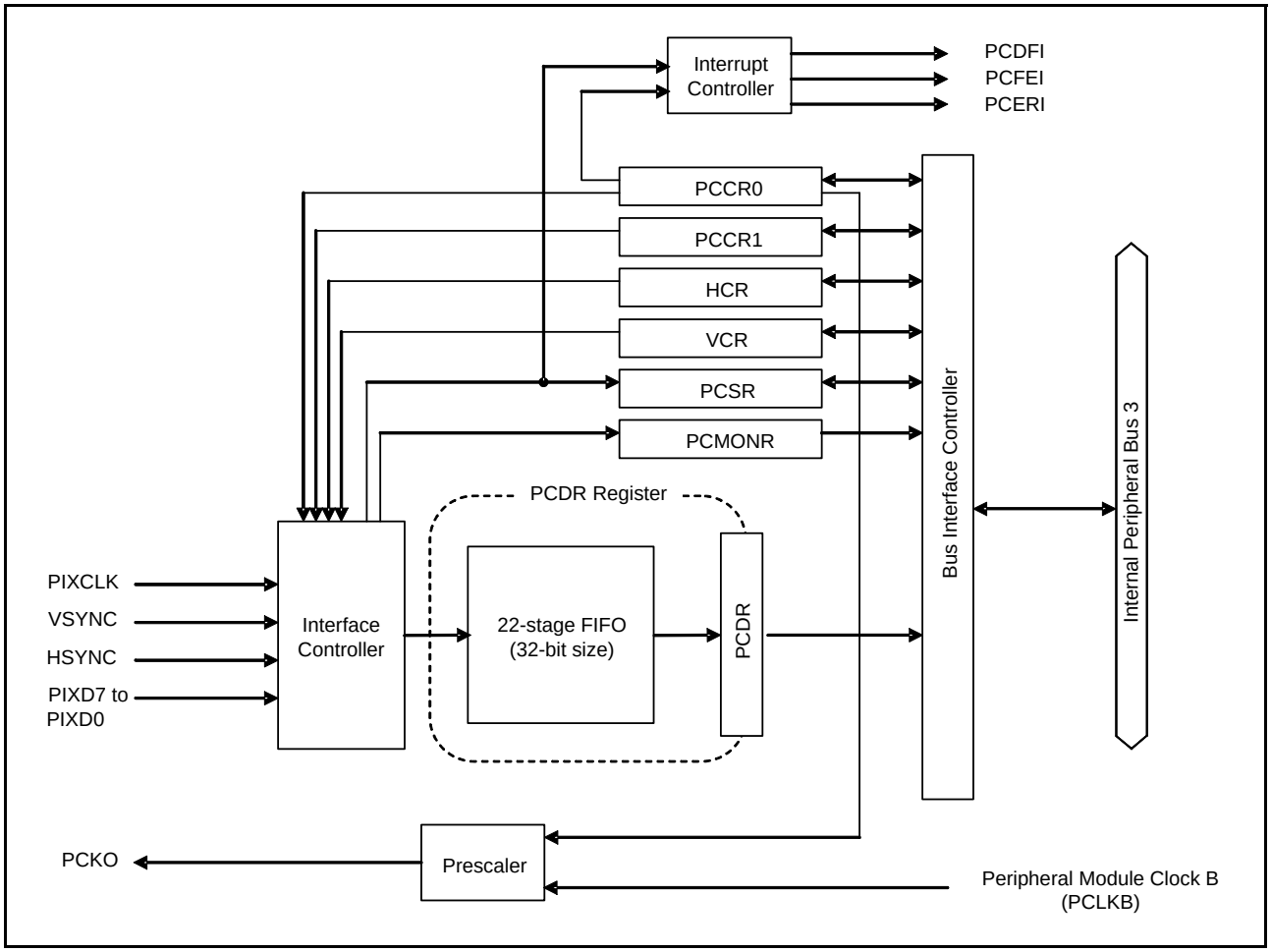


Figure 51.1 Block Diagram of the PDC

Table 51.2 Input and Output Pins of the PDC

Pin Name	Input/Output	Description
PIXCLK	Input	Parallel data transfer clock
VSYNC	Input	Vertical synchronization signal
HSYNC	Input	Horizontal synchronization signal
PIXD7 to PIXD0	Input	8-bit data
PCKO	Output	Output of the parallel data transfer clock

51.2 Register Descriptions

51.2.1 PDC Control Register 0 (PCCR0)

Address(es): 000A 0500h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	EDS	PCKDIV[2:0]			PCKOE	HERIE	VERIE	UDRIE	OVIE	FEIE	DFIE	PRST	HPS	VPS	PCKE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Name	Symbol	Bit Name	Description	R/W
b0	PCKE	PIXCLK Input Enable	0: PIXCLK input is disabled. 1: PIXCLK input is enabled.	R/W
b1	VPS	VSYNC Signal Polarity Select	0: VSYNC signal is active high. 1: VSYNC signal is active low.	R/W
b2	HPS	HSYNC Signal Polarity Select	0: HSYNC signal is active high. 1: HSYNC signal is active low.	R/W
b3	PRST	PDC Reset	0: PDC reset is not applied. 1: PDC is reset.	R/(W) *1
b4	DFIE	Receive Data Ready Interrupt Enable	0: Generation of receive data ready interrupt requests is disabled. 1: Generation of receive data ready interrupt requests is enabled.	R/W
b5	FEIE	Frame End Interrupt Enable	0: Generation of frame end interrupt requests is disabled. 1: Generation of frame end interrupt requests is enabled.	R/W
b6	OVIE	Overrun Interrupt Enable	0: Generation of overrun interrupt requests is disabled. 1: Generation of overrun interrupt requests is enabled.	R/W
b7	UDRIE	Underrun Interrupt Enable	0: Generation of underrun interrupt requests is disabled. 1: Generation of underrun interrupt requests is enabled.	R/W
b8	VERIE	Vertical Line Number Setting Error Interrupt Enable	0: Generation of vertical line number setting error interrupt requests is disabled. 1: Generation of vertical line number setting error interrupt requests is enabled.	R/W
b9	HERIE	Horizontal Byte Number Setting Error Interrupt Enable	0: Generation of horizontal byte number setting error interrupt requests is disabled. 1: Generation of horizontal byte number setting error interrupt requests is enabled.	R/W
b10	PCKOE	PCKO Output Enable	0: PCKO output is disabled (fixed to the high level) 1: PCKO output is enabled.	R/W
b13 to b11	PCKDIV[2:0]	PCKO Frequency Division Ratio Select	b13 b11 0 0 0: PCKO/2 0 0 1: PCKO/4 0 1 0: PCKO/6 0 1 1: PCKO/8 1 0 0: PCKO/10 1 0 1: PCKO/12 1 1 0: PCKO/14 1 1 1: PCKO/16	R/W
b14	EDS	Endian Select	0: Little endian 1: Big endian	R/W
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 1 can be written to this bit.

The PCCR0 register should only be set while the PCE bit in the PCCR1 register is 0.

PCKE Bit (PIXCLK Input Enable)

This bit enables and disables the input through the PIXCLK pin.
Set this bit to 1 before enabling reception. After enabling input through the PIXCLK pin, use the PRST bit to initialize the PDC.
When setting this bit to 0, do so after disabling reception operations.

VPS Bit (VSYNC Signal Polarity Select)

This bit selects the active sense of the VSYNC signal.

HPS Bit (HSYNC Signal Polarity Select)

This bit selects the active sense of the HSYNC signal.

PRST Bit (PDC Reset)

This bit initializes the internal status of the PDC and the target registers of the PDC reset. See section 51.3.11, Reset State for the target registers.
The PDC should be reset after setting the PCKE bit to 1.
When 1 is written to the PRST bit, initialization is started in synchronization with the PIXCLK. After completion of the initialization, the PRST bit is automatically cleared to 0. When the PDC is reset, ensure that the PIXCLK pin has an input signal. Also, after 1 has been written to the PRST bit, do not proceed to the next step until verifying that the bit has returned to 0.
In the case of consecutive PDC resets, wait for at least one cycle of PIXCLK after verifying that the PRST bit has returned to 0.

DFIE Bit (Receive Data Ready Interrupt Enable)

This bit enables and disables the generation of receive data ready interrupt requests.

FEIE Bit (Frame End Interrupt Enable)

This bit enables and disables the generation of frame end interrupt requests.

OVIE Bit (Overrun Interrupt Enable)

This bit enables and disables the generation of overrun interrupt requests.

UDRIE Bit (Underrun Interrupt Enable)

This bit enables and disables the generation of underrun interrupt requests.

VERIE Bit (Vertical Line Number Setting Error Interrupt Enable)

This bit enables and disables the generation of vertical line number setting error interrupt requests.

HERIE Bit (Horizontal Byte Number Setting Error Interrupt Enable)

This bit enables and disables the generation of horizontal byte number setting error interrupt requests.

PCKOE Bit (PCKO Output Enable)

This bit enables and disables an output from PCKO.
When the PCKOE bit is cleared to 0 during low output of PCKO, it may cause high output at the time of clearing, resulting in spoiling the duty factor.

PCKDIV[2:0] Bits (PCKO Frequency Division Ratio Select)

This bit selects the frequency division ratio of the PCKO.

The PCKO output is a clock signal derived by dividing the PCLKB clock signal by the value from 2 to 16 corresponding to the setting of the PCKDIV[2:0] bits.

The operating frequency should be set in accord with that of PCLKB to a value within the range from 1 to 30 MHz.

EDS Bit (Endian Select)

This bit selects an endian for the captured data.

51.2.2 PDC Control Register 1 (PCCR1)

Address(es): 000A 0504h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Name	Symbol	Bit Name	Description	R/W
b0	PCE	PDC Operation Enable	0: Operations for reception are disabled. 1: Operations for reception are enabled.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PCE Bit (PDC Operation Enable)

This bit enables and disables operations for reception.

When the PCE bit is set to 1 during assertion of the VSYNC signal, the PDC starts operations for reception from the next effective edge of the VSYNC signal.

The PCE bit should only be cleared to 0 while operations for reception or continued reception are stopped, including the frame end interrupt and so on. Regarding operations for continued reception, see section 51.3.6, Operations for Continued Reception at the Time of Frame End.

51.2.3 PDC Status Register (PCSR)

Address(es): 000A 0508h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	HERF	VERF	UDRF	OVRF	FEF	FEMPF	FBSY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit Name	Symbol	Bit Name	Description	R/W
b0	FBSY	Frame Busy Flag	0: Operations for reception are stopped. 1: Operations for reception are ongoing.	R
b1	FEMPF	FIFO Empty Flag	0: FIFO is not empty. 1: FIFO is empty.	R
b2	FEF	Frame End Flag	0: Frame end has not been generated. 1: Frame end has been generated.	R/(W) *1
b3	OVRF	Overrun Flag	0: FIFO overrun has not been generated. 1: FIFO overrun has been generated.	R/(W) *1
b4	UDRF	Underrun Flag	0: Underrun has not been generated. 1: Underrun has been generated.	R/(W) *1
b5	VERF	Vertical Line Number Setting Error Flag	0: Vertical line number setting error has not been generated. 1: Vertical line number setting error has been generated.	R/(W) *1
b6	HERF	Horizontal Byte Number Setting Error Flag	0: Horizontal byte number setting error has not been generated. 1: Horizontal byte number setting error has been generated.	R/(W) *1
b31 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. 0 can only be written to these bits to clear these flags after they have been read out as 1.

FBSY Flag (Frame Busy Flag)

This flag indicates the state of PDC operations.

[Setting Condition]

- Detection of the effective edge of the VSYNC signal after the enabling of operations for reception.

[Clearing Conditions]

- Reception of one-frame of data according to the settings of the VCR and HCR registers being completed.*1
- When an error (overrun, underrun, vertical line number setting error, or horizontal byte number setting error) occurs
- When the PCCR1.PCE bit is 0

Note 1. This flag is 0 during operations for continued reception.

FEMPF Flag (FIFO Empty Flag)

This flag indicates the state of the FIFO.

This flag indicates the state of the FIFO at the time when a vertical line number setting error or a horizontal byte number setting error occurs and is 0 following an overrun and undefined following an underrun.

[Setting Conditions]*1

- Reading of the PCDR register while the FIFO is empty.
- Detection of an effective edge of the VSYNC signal.

- The PDC being reset.

[Clearing Condition]*1

- Storage of the data captured in the FIFO.

Note 1. This flag is undefined following an underrun, and is set to either 0 or 1.

FEF Flag (Frame End Flag)

This flag indicates the end of a frame.

[Setting Condition]

- Reception of one frame of data in accord with the settings of the VCR and HCR registers.*1

[Clearing Conditions]

- The PDC being reset.
- Writing of 0 to the bit after it has been read as 1.

Note 1. In the case of operations for continued reception, this flag will be 1 after their completion.

OVRF Flag (Overrun Flag)

This flag indicates an overrun.

[Setting Condition]

- Data for reception arriving while the FIFO is full.

[Clearing Conditions]

- The PDC being reset.
- Writing of 0 to the bit after it has been read as 1.

UDRF Flag (Underrun Flag)

This flag indicates an underrun.

[Setting Condition]

- Reading of the PCDR register while the FIFO is empty.

[Clearing Conditions]

- The PDC being reset.
- Writing of 0 to the bit after it has been read as 1.

VERF Flag (Vertical Line Number Setting Error Flag)

This flag indicates an error in the setting for the number of lines.

[Setting Condition]

- The VSYNC signal is negated because fewer lines have been captured than the value in the VCR register.

[Clearing Conditions]

- The PDC being reset.
- Writing of 0 to the bit after it has been read as 1.

HERF Flag (Horizontal Byte Number Setting Error Flag)

This flag indicates an error in the setting for the number of bytes in a line.

[Setting Condition]

- The HSYNC signal is negated because fewer bytes in a line have been captured than the value in the HCR register.

[Clearing Conditions]

- The PDC being reset.
- Writing of 0 to the bit after it has been read as 1.

51.2.4 PDC Pin Monitor Register (PCMONR)

Address(es): 000A 050Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HSYNC	VSYNC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Name	Symbol	Bit Name	Description	R/W
b0	VSYNC	VSYNC Signal Status Flag	0: VSYNC signal is at the low level. 1: VSYNC signal is at the high level.	R
b1	HSYNC	HSYNC Signal Status Flag	0: HSYNC signal is at the low level. 1: HSYNC signal is at the high level.	R
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

VSYNC Flag (VSYNC Signal Status Flag)

This flag indicates the state of the VSYNC signal.

HSYNC Flag (HSYNC Signal Status Flag)

This flag indicates the state of the HSYNC signal.

51.2.5 PDC Receive Data Register (PCDR)

Address(es): 000A 0510h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The PDC includes a 32-bit, 22-stage FIFO buffer for the storage of captured data. The FIFO buffer is mapped to the 4-byte PCDR register, and captured data are read from this register in 4-byte units. The receive data ready flag is set for every 32 bytes of received data, and this also leads to a receive data ready interrupt if the DFIE bit in the PCCR0 register is set to 1. When a receive data ready interrupt is generated, read the PCDR register eight times. Figure 51.2 shows a schematic view of the PCDR register.

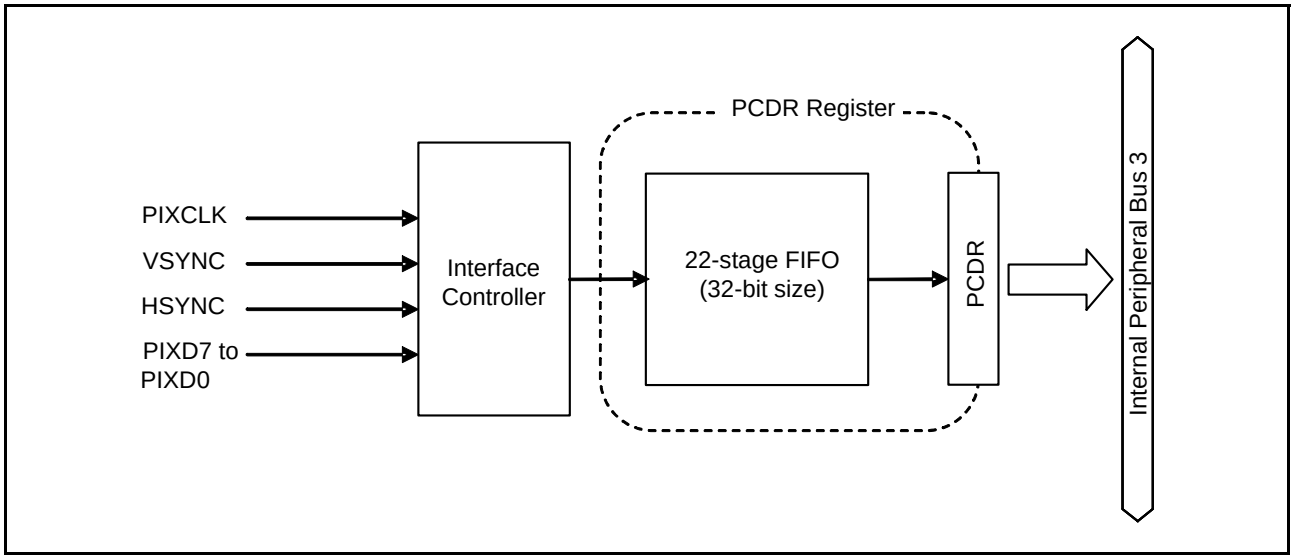


Figure 51.2 Schematic View of the PCDR Register

For the format of the captured data, either big or little endian can be selected by the EDS bit of the PCCR0 register. Figure 51.3 shows the arrangements of data according to the endian formats.

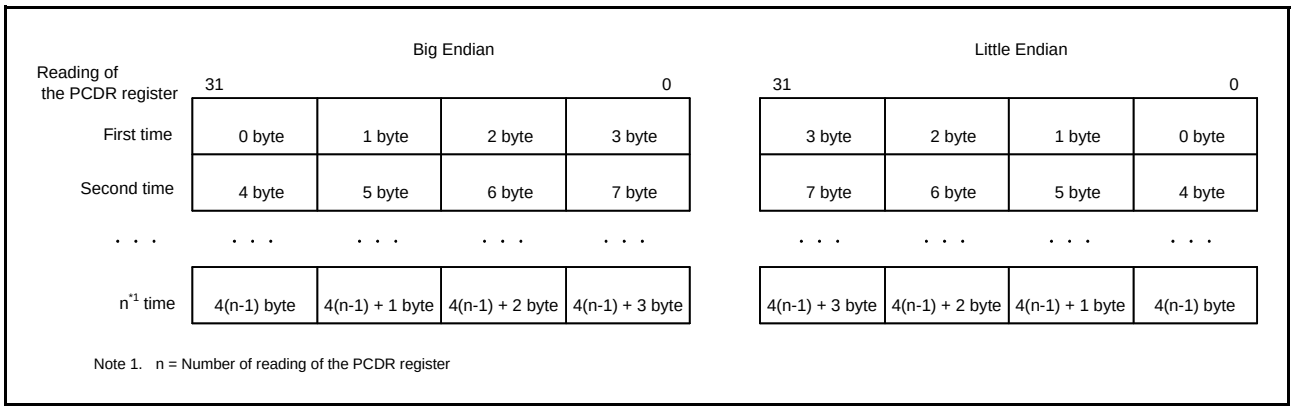


Figure 51.3 Endian Formats

51.2.6 Vertical Capture Register (VCR)

Address(es): 000A 0514h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	VSZ[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	VST[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Name	Symbol	Bit Name	Description	R/W
b11 to b0	VST[11:0]	Vertical Capture Start Line Position	Number of the line where capture is to start.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b16	VSZ[11:0]	Vertical Capture Size	Number of lines to be captured.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

For the relationship between the VCR register setting and the captured range, see section 51.3.3, Settings of the VCR and HCR Registers and the Captured Range.
The VCR register should be set while the PCE bit in the PCCR1 register is 0.

VST[11:0] Bit (Vertical Capture Start Line Position)

These bits set the number of the line where capture is to start.
To set the first line, these bits should be 000h; to set the 4095th line, they should be FFEh.
The setting of the VST[11:0] bits should be within the range from 000h to FFEh and, in combination with that of the VSZ[11:0] bits, satisfy the following relation:
Setting range of the VST[11:0] bits: $1 \leq VST[11:0] + VSZ[11:0] \leq FFFh$.

VSZ[11:0] Bit (Vertical Capture Size)

These bits set the number of lines to be captured.
To set one line, these bits should be 001h; to set 4095 lines, they should be FFFh.
The setting of the VSZ[11:0] bits should be within the range from 001h to FFFh and, in combination with that of the VST[11:0] bits, satisfy the following relation:
Setting range of the VSZ[11:0] bits: $1 \leq VST[11:0] + VSZ[11:0] \leq FFFh$.

51.2.7 Horizontal Capture Register (HCR)

Address(es): 000A 0518h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	HSZ[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	HST[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Name	Symbol	Bit Name	Description	R/W
b11 to b0	HST[11:0]	Horizontal Capture Start Byte Position	Horizontal position in bytes where capture is to start.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b16	HSZ[11:0]	Horizontal Capture Size	Number of bytes to be captured horizontally.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

For the relationship between the HCR register setting and the captured range, see section 51.3.3, Settings of the VCR and HCR Registers and the Captured Range.
The HCR register should be set while the PCE bit in the PCCR1 register is 0.

HST[11:0] Bit (Horizontal Capture Start Byte Position)

These bits set the horizontal position in bytes where capture is to start.
To set the first byte, these bits should be 000h; to set the 4092th byte, they should be FFBh.
The setting of the HST[11:0] bits should be within the range from 000h to FFBh and, in combination with that of the HSZ[11:0] bits, satisfy the following relation:
Setting range of the HST[11:0] bits: $4 \leq HST[11:0] + HSZ[11:0] \leq FFFh$.

HSZ[11:0] Bit (Horizontal Capture Size)

These bits set the number of bytes to be captured per line.
To set four bytes, these bits should be 004h; to set 4095 bytes, they should be FFFh.
The setting of the HSZ[11:0] bits should be within the range from 004h to FFFh and, in combination with that of the HST[11:0] bits, satisfy the following relation:
Setting range of the HSZ[11:0] bits: $4 \leq HST[11:0] + HSZ[11:0] \leq FFFh$.

51.3 Operation

51.3.1 Transfer Formats

The PDC supports the four transfer formats shown in Figure 51.4 to Figure 51.7. The format is determined by the setting of the VPS and HPS bits in the PCCR0 register.

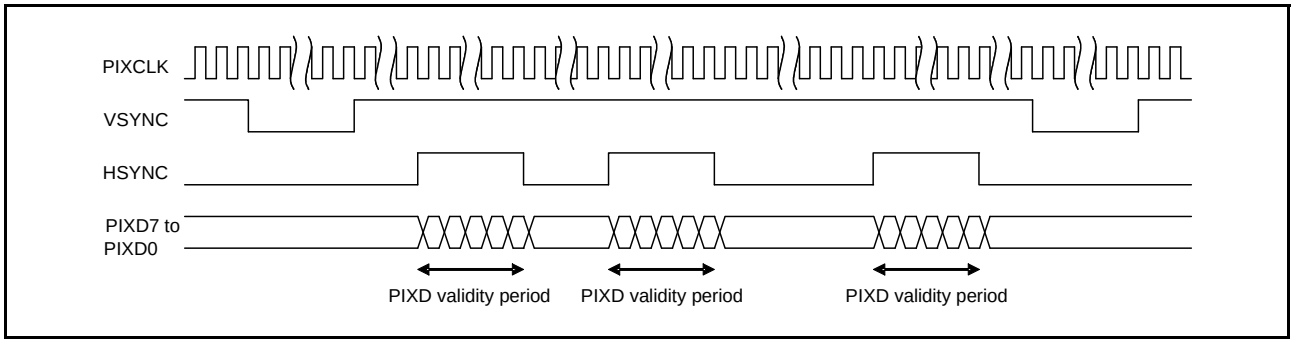


Figure 51.4 PDC Transfer Format (for VPS=0, HPS=0)

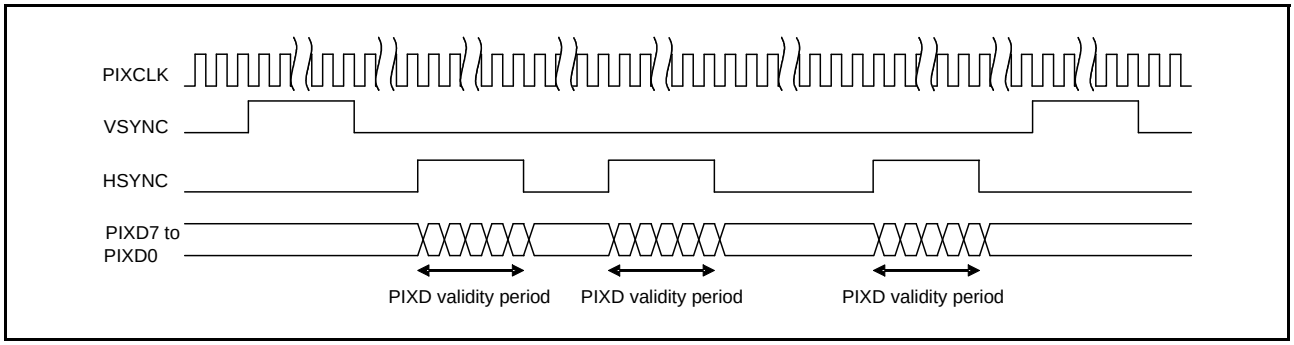


Figure 51.5 PDC Transfer Format (for VPS=1, HPS=0)

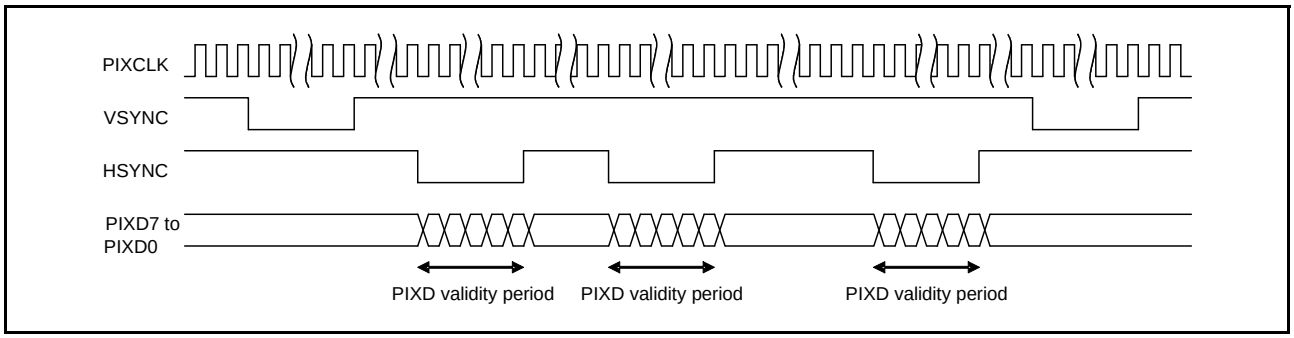


Figure 51.6 PDC Transfer Format (for VPS=0, HPS=1)

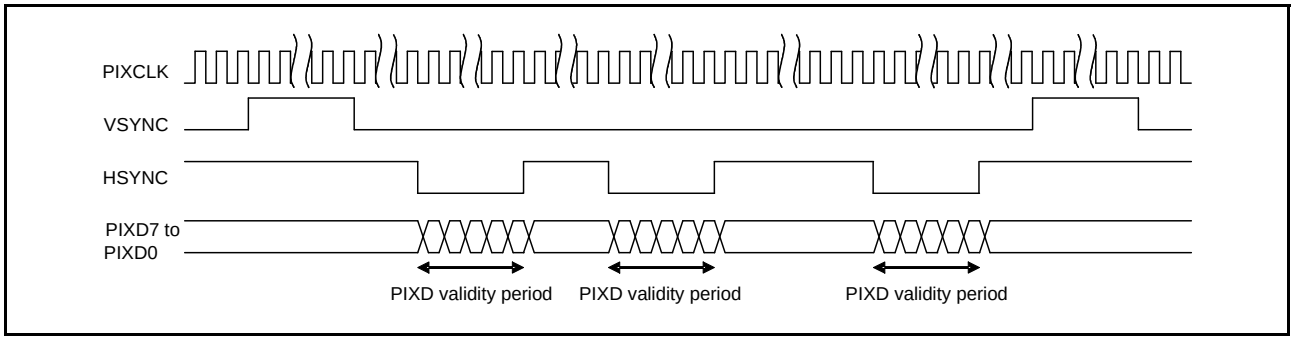


Figure 51.7 PDC Transfer Format (for VPS=1, HPS=1)

51.3.2 Transfer Timing

Figure 51.8 and Table 51.3 show the timing of transfer by the PDC.

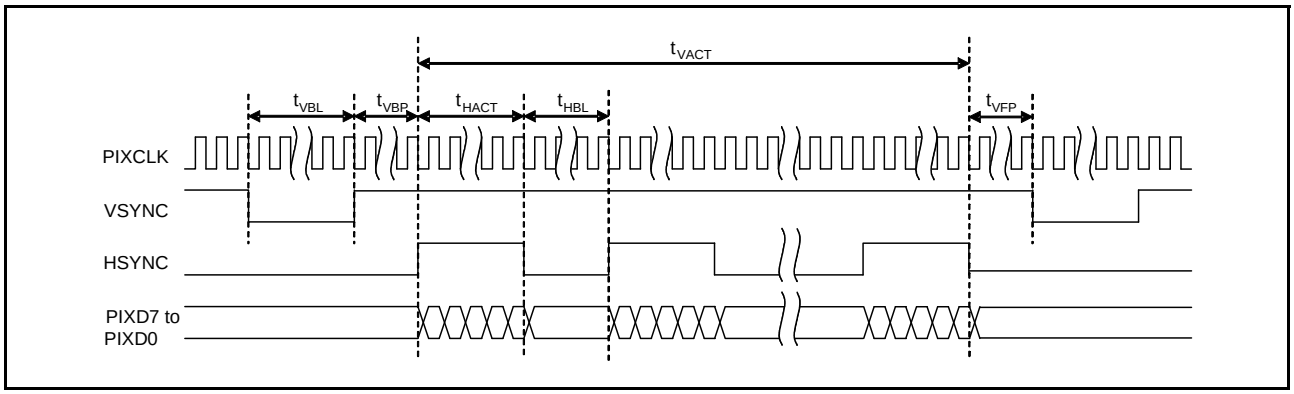


Figure 51.8 PDC Transfer Timing

Table 51.3 PDC Transfer Timing

Item	Symbol	Min*1	Max	Unit
Vertical Blanking Period	t_{VBL}	128	—	PIXCLK
Vertical Backporch	t_{VBP}	10	—	PIXCLK
Horizontal Valid Period	t_{HACT}	4	4095	PIXCLK
Horizontal Blanking Period	t_{HBL}	128	—	PIXCLK
Vertical Frontporch	t_{VFP}	10	—	PIXCLK
Vertical Valid Period	t_{VACT}	1	4095	Line

Note 1. The minimum values are the lowest of which this MCU circuit is capable. They are not values which guarantee the avoidance of overruns, vertical line number setting errors, or horizontal byte number setting errors.

51.3.3 Settings of the VCR and HCR Registers and the Captured Range

Figure 51.9 and Figure 51.10 show the relationship between the settings of the VCR and HCR registers and the captured range.

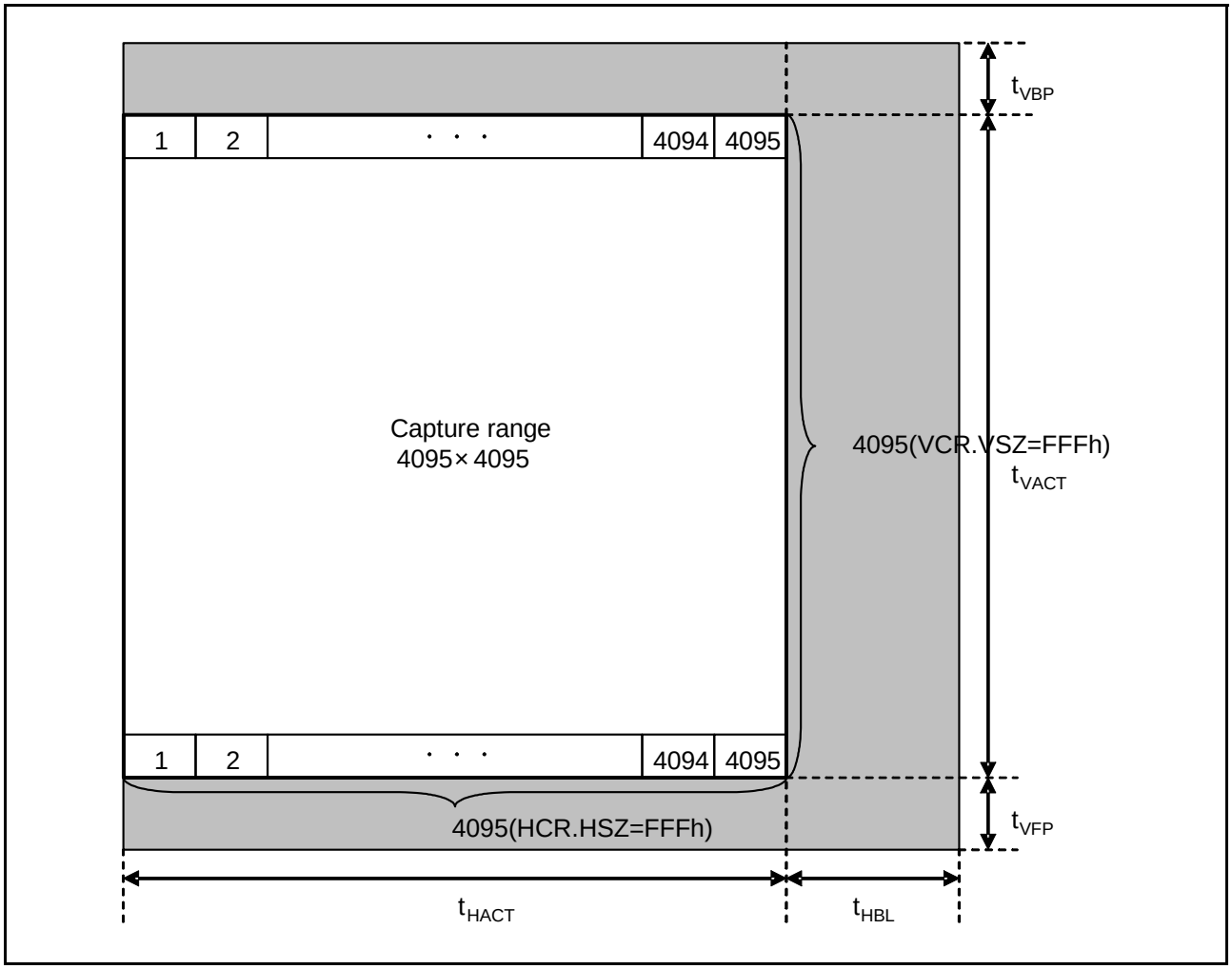


Figure 51.9 Settings of the VCR and HCR Registers and the Captured Range (VCR = 0FFF 0000h, HCR = 0FFF 0000h)

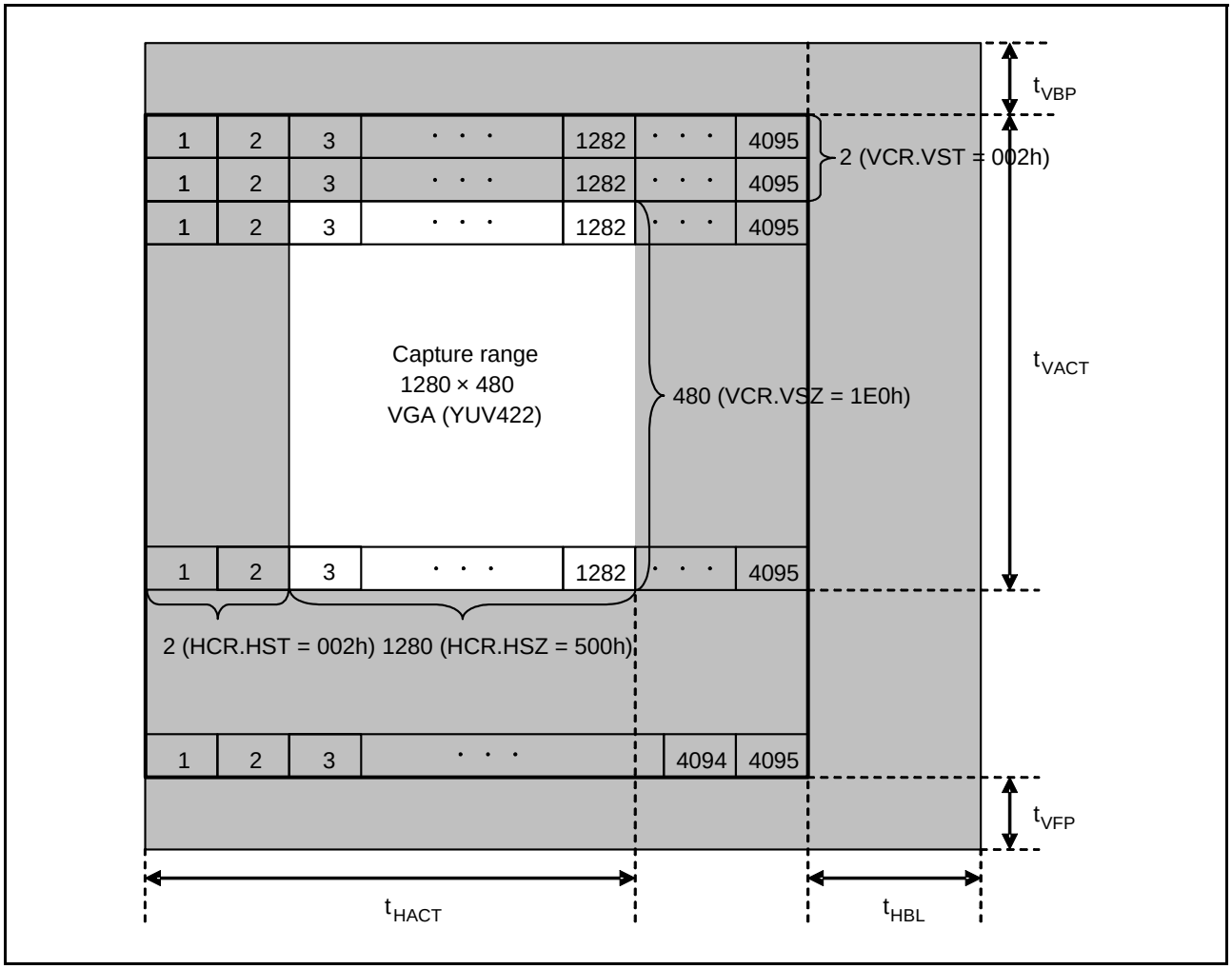


Figure 51.10 Settings of the VCR and HCR Registers and the Captured Range (VCR = 01E0 0002h, HCR = 0500 0002h)

51.3.4 Operations for Reception

Figure 51.11 shows an example of operations for reception when receive data ready interrupts (to start up the DTC or DMAC) and frame end interrupts are in use.

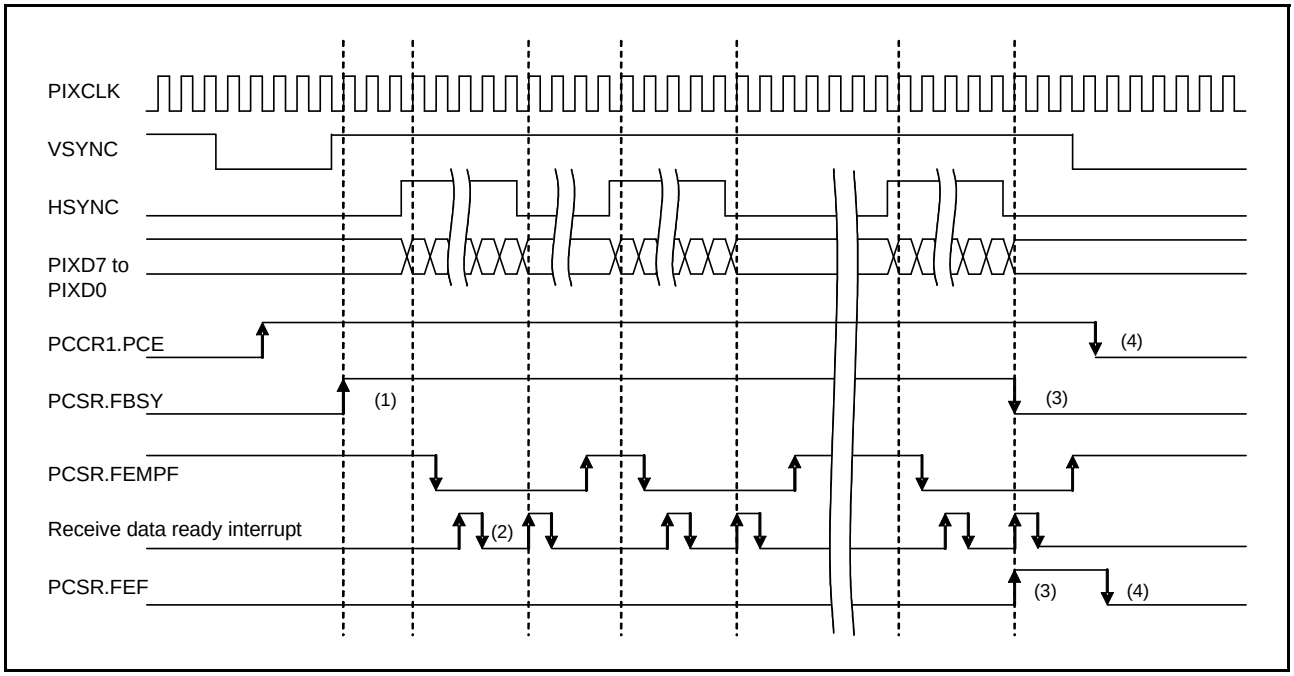


Figure 51.11 Example of Operations for Reception

The actual operations at the times indicated by (1), (2), (3), and (4) in Figure 51.11 are described below.

(1) When an effective edge of the VSYNC signal is detected after the PCE bit in the PCCR1 register is set to 1, the FEMPF flag in the PCSR register is set to 1 and the FIFO is initialized. Concurrently, the FBSY flag in the PCSR register is set to 1 and operations for reception start.

(2) When data within the range for capture set in the VCR and HCR registers is received, the data are stored in the FIFO. The PDC generates a receive data ready interrupt every time it receives 32 bytes of data, and the interrupt starts transfer of the captured data by the DTC or DMAC to the on-chip RAM or an external address space. The FIFO is likely to overrun if reading of the PCDR register takes more time than reception of the data. Check the OVRF flag in the PCSR register to verify an overrun.

(3) After reception of the last byte of data is completed, the FBSY flag in the PCSR register is cleared to 0 and the FEF flag in the PCSR register is set to 1 so that a receive data ready interrupt and frame end interrupt are generated.

(4) The FEMPF flag in the PCSR register is polled by the frame end interrupt, after which the completion of data transfer by the DTC or DMAC should be verified. After the PCE bit in the PCCR1 register is cleared to 0, the FEF flag in the PCSR register is also cleared to 0, and the reception of one frame of data is complete.
If an underrun occurs while the PCSR.FEMPF flag is being polled, setting of the PCSR.FEMPF flag may not proceed. Accordingly, also check the PCSR.UDRF flag during polling, and if an underrun occurs, run appropriate error processing.

If the FEF flag in the PCSR register has been set to 1 before the PCE bit in the PCCR1 register is set to 1, effective edges of the VSYNC signal are not detected and operations for reception will not be started. Clear the FEF flag in the PCSR register to 0 to start operations for data reception.

51.3.5 Operation during the Horizontal Blanking Period

If the horizontal blanking period begins but the number of the received data bytes has not reached 32 bytes since the previous receive data ready, the count of the number of received data bytes is retained and carried over to the next horizontal valid period. Figure 51.12 shows an example of operation during the horizontal blanking period.

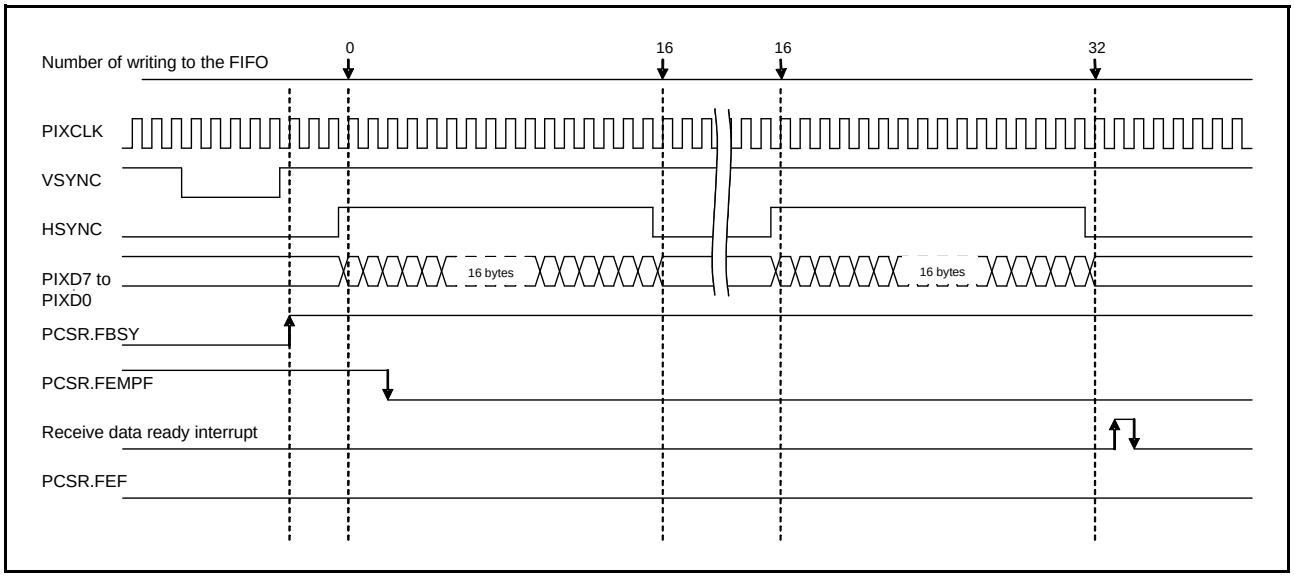


Figure 51.12 Example of Operation during the Horizontal Blanking Period

51.3.6 Operations for Continued Reception at the Time of Frame End

When the last of the data have been received but the number of bytes of data received since the previous receive data ready has not reached 32, the PDC continues to receive data (hereinafter referred to as “continued reception”) until the number reaches 32. After that, it generates a receive data ready and a frame end. The PIXCLK should be input during operations for continued reception. If the data stored in the FIFO are read out during operations for continued reception, the values read are undefined. Figure 51.13 shows an example of operations at the time of frame end.

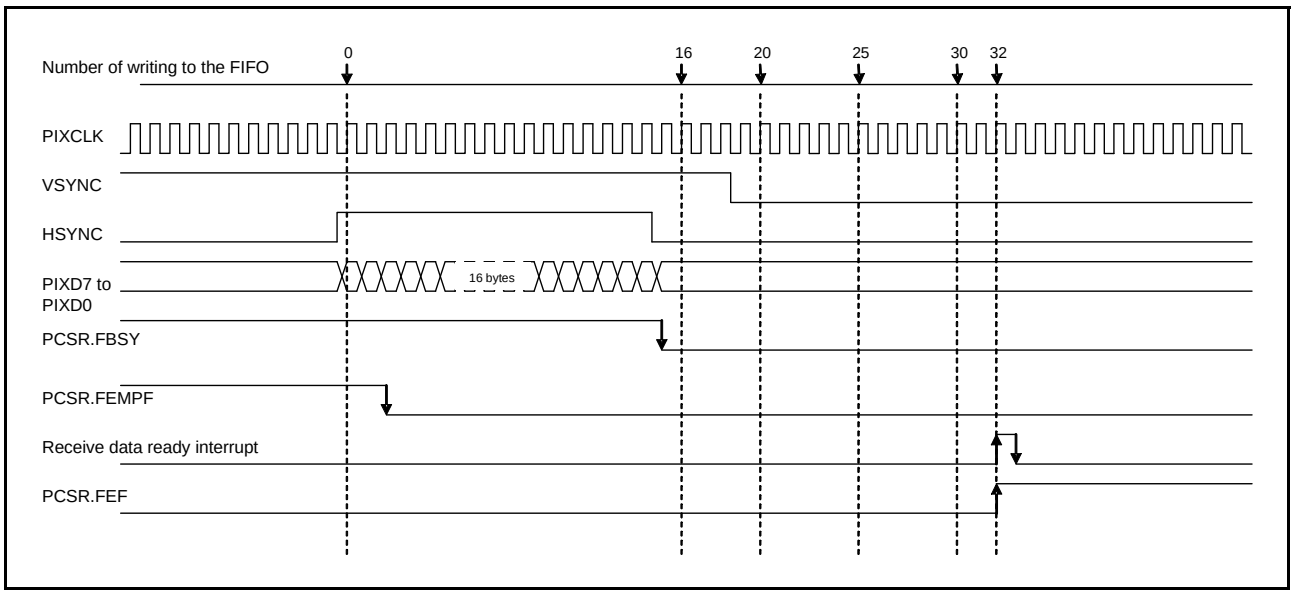


Figure 51.13 Example of Continued Reception at the Time of Frame End

51.3.7 Error Detection

The PDC has functions for error detection so that the software can apply control in response to errors during operations for reception. Table 51.4 summarizes the conditions under which each type of error is detected and the interrupt flags set in response.

Table 51.4 Error Detection

Error Factor	Conditions of Error Detection	Interrupt Flag	Example of Operation
Overrun	Data for reception arriving while the FIFO is full.*1	PCSR.OVRF	Figure 51.14
Underrun	The PCDR register being read while the FIFO is empty.	PCSR.UDRF	Figure 51.15
Vertical line number setting error	Negation of the VSYNC signal when the number of captured lines is less than the value set in the VCR register.	PCSR.VERF	Figure 51.16
Horizontal byte number setting error	Negation of the HSYNC signal when the number of bytes captured in a line is less than the value set in the HCR register.	PCSR.HERF	Figure 51.17

Note 1. This includes data reception during operations for continued reception.

When an error is detected, the PDC sets the corresponding interrupt flag to 1 to stop operations for reception. While the interrupt flag is set to 1, the PDC does not detect effective edges of the VSYNC signal and does not start operations for reception. Clear all error source interrupt flags to 0 to start operations for reception.
In addition, when an error occurs, data stored in the FIFO is disabled.

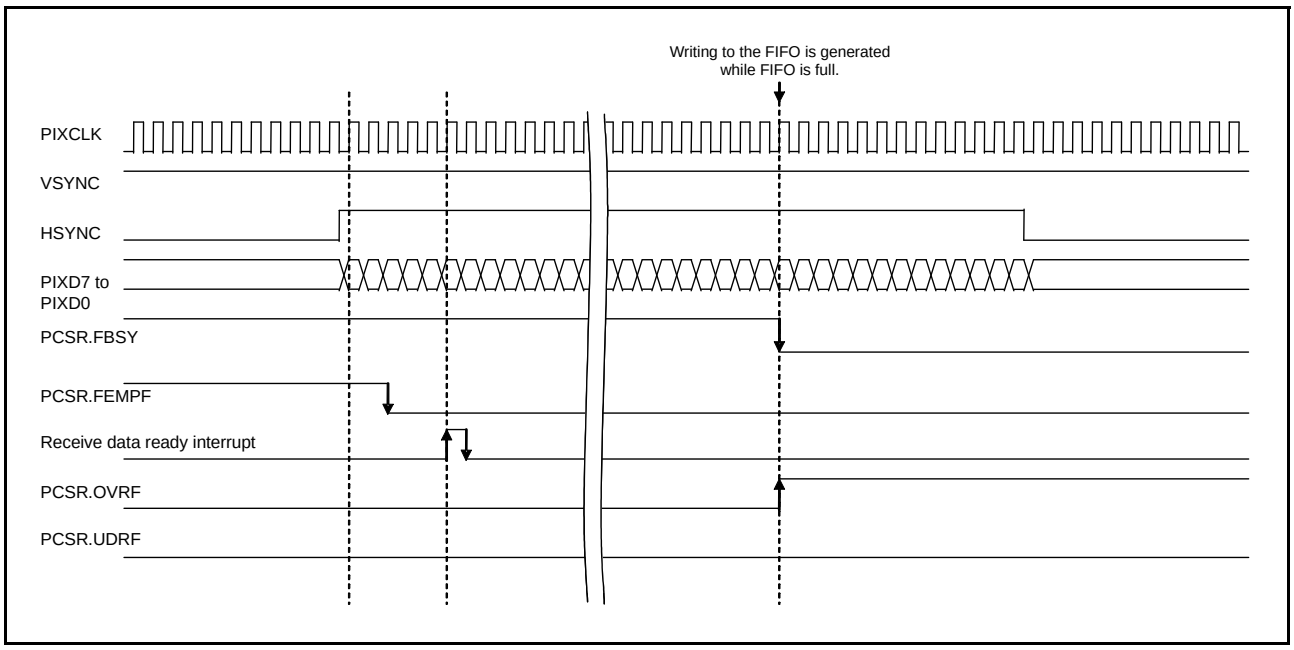


Figure 51.14 Overrun Detected

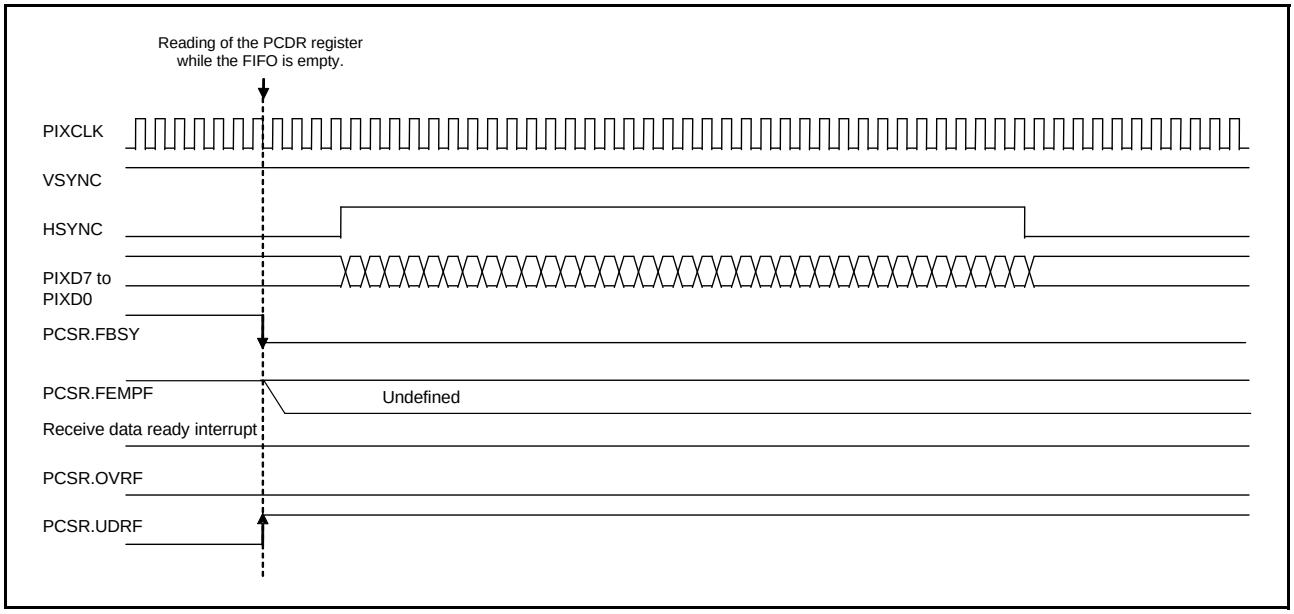


Figure 51.15 Underrun Detected

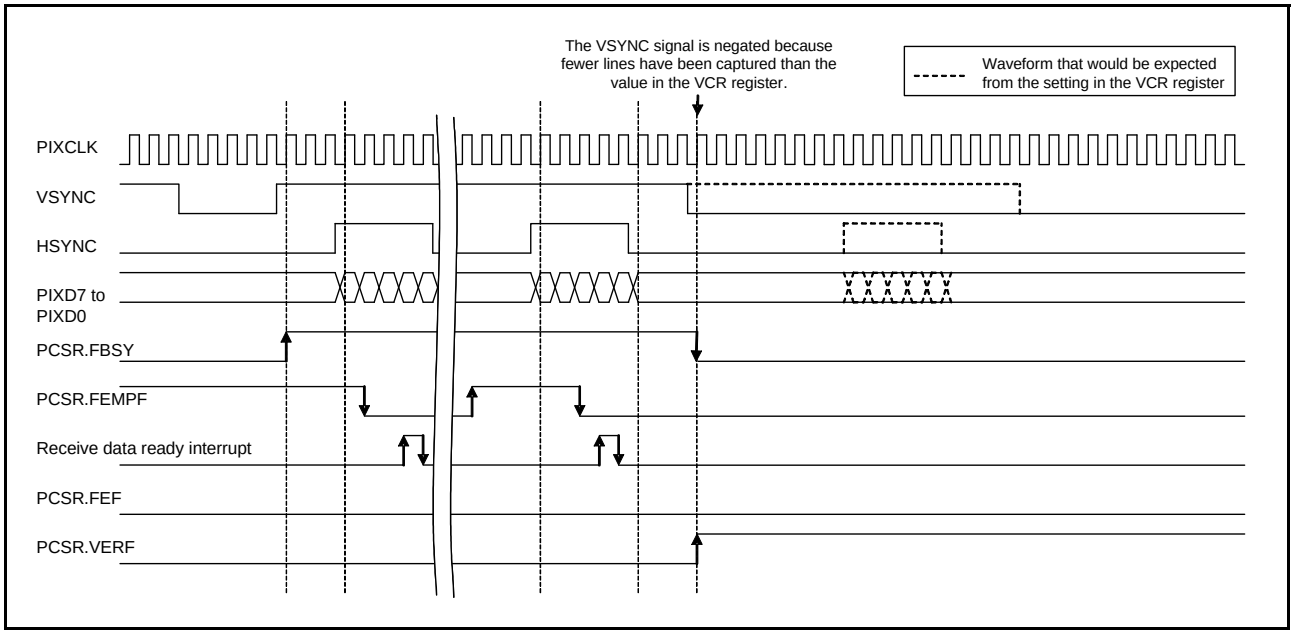


Figure 51.16 Vertical Line Number Setting Error Detected

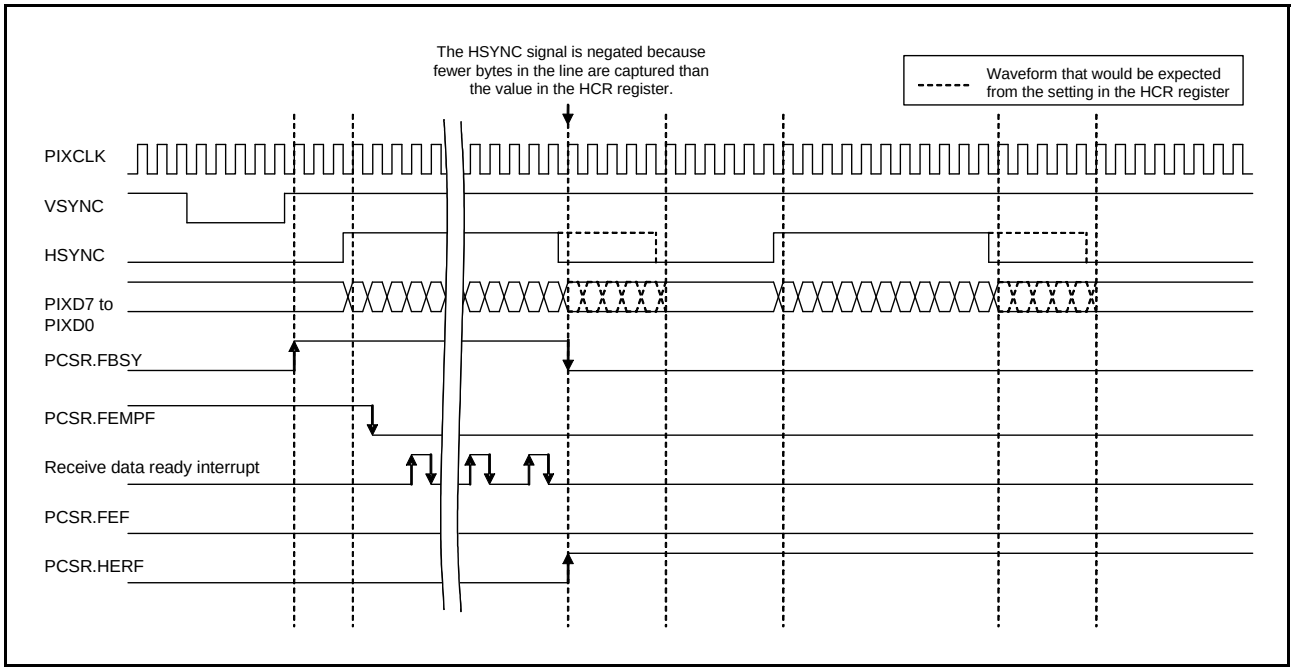


Figure 51.17 Horizontal Byte Number Setting Error Detected

51.3.8 Initial Settings

Figure 51.18 is a sample flowchart for initial settings. For a description of how to set up the input and output ports and the interrupt controller, see the descriptions given in the sections on the relevant blocks.

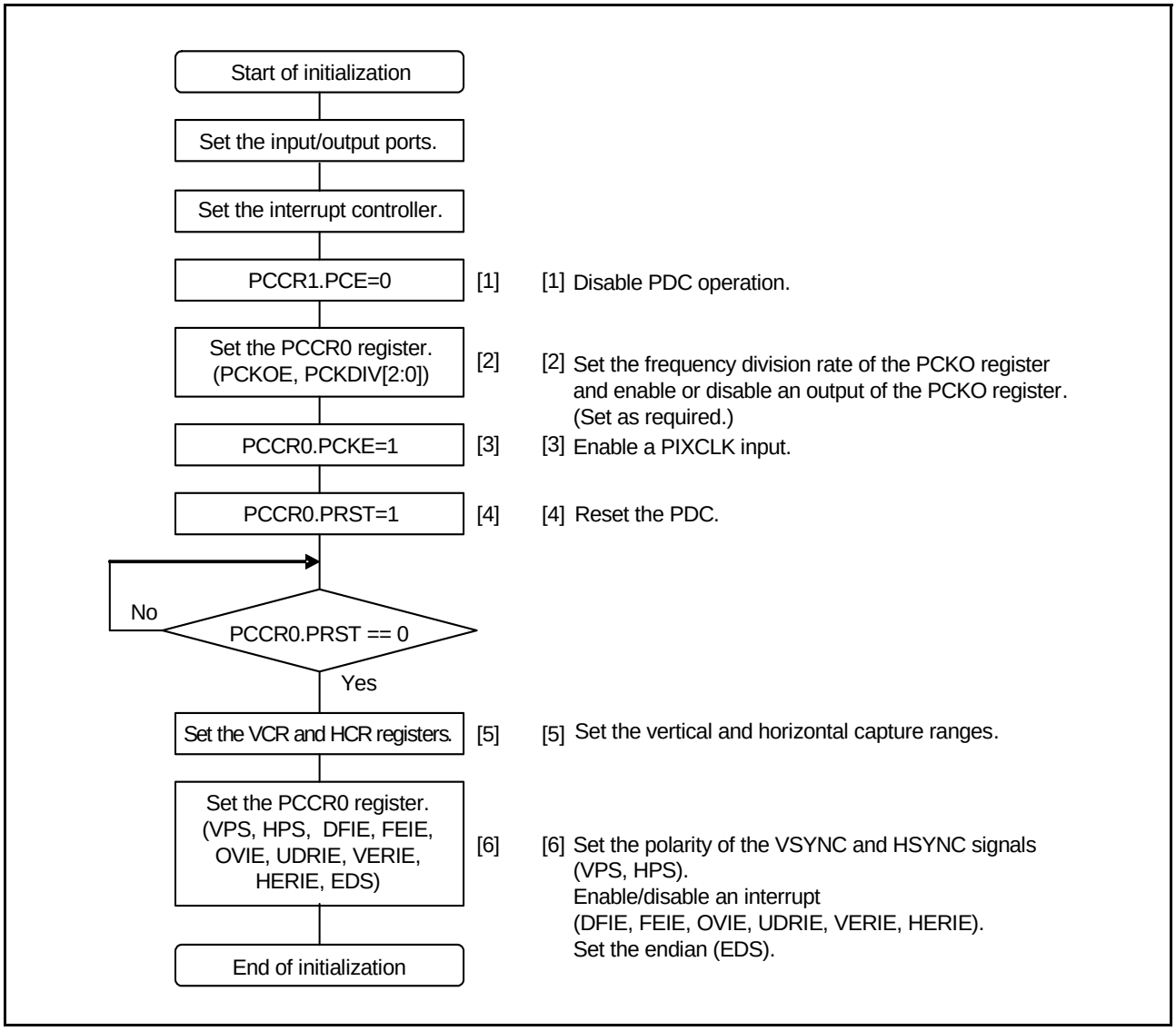
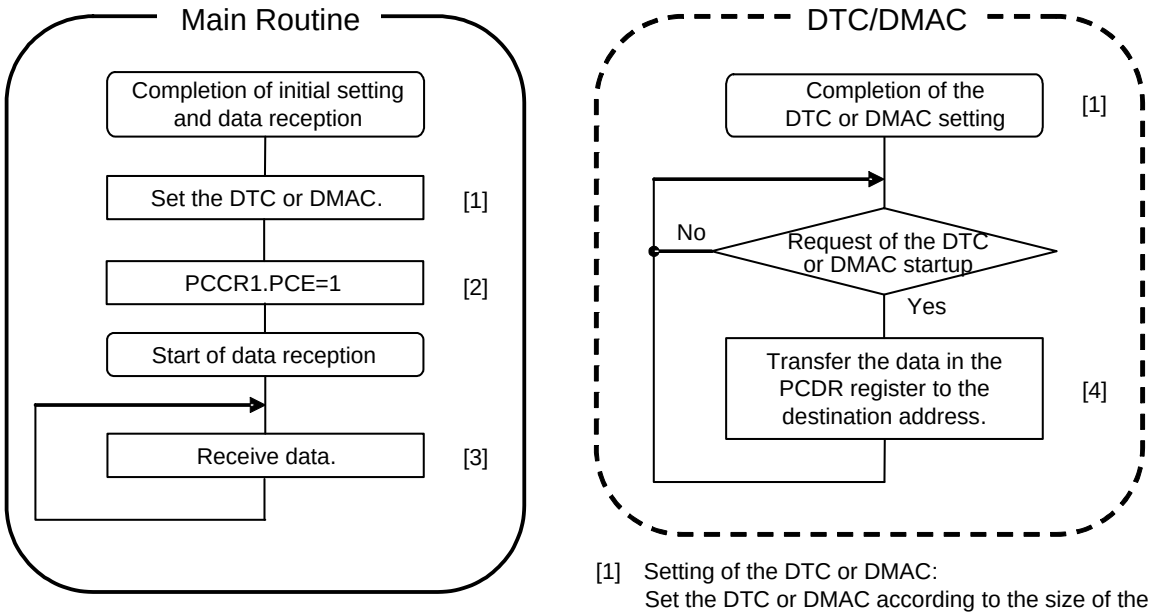
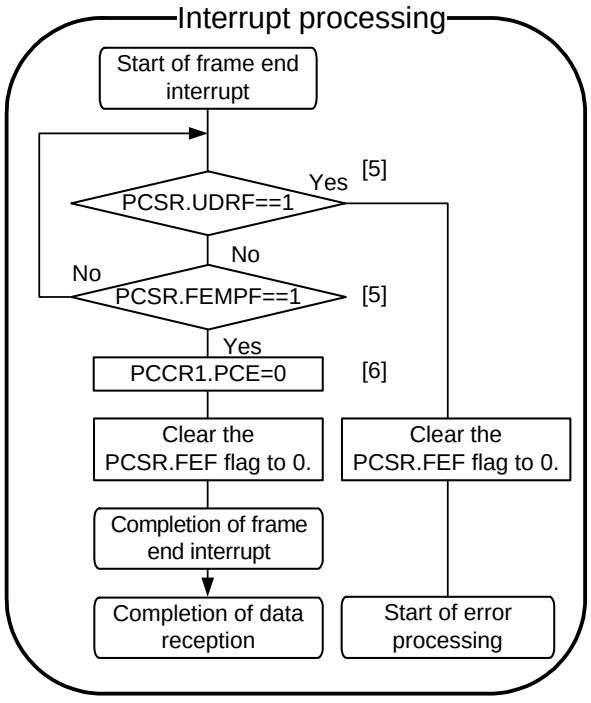


Figure 51.18 Sample Flowchart for Initial Setting of the PDC

51.3.9 Flows of Operations

Figure 51.19 shows an example of the flows of operations when the receive data ready interrupt (to start up the DTC or DMAC) and frame end interrupt are in use. For a description of how to set up the DTC or DMAC, see section 18, DMA Controller (DMACAa) and section 20, Data Transfer Controller (DTCa).





[1] Setting of the DTC or DMAC:
Set the DTC or DMAC according to the size of the captured image.

[2] Enabling of PDC operation:
Set the PCE bit in the PCCR1 register to 1 to enable PDC operation.

[3] Data reception:
Receive the data from the external I/O and wait for generation of a receive data ready interrupt (DTC/DMAC startup) or a frame end interrupt.

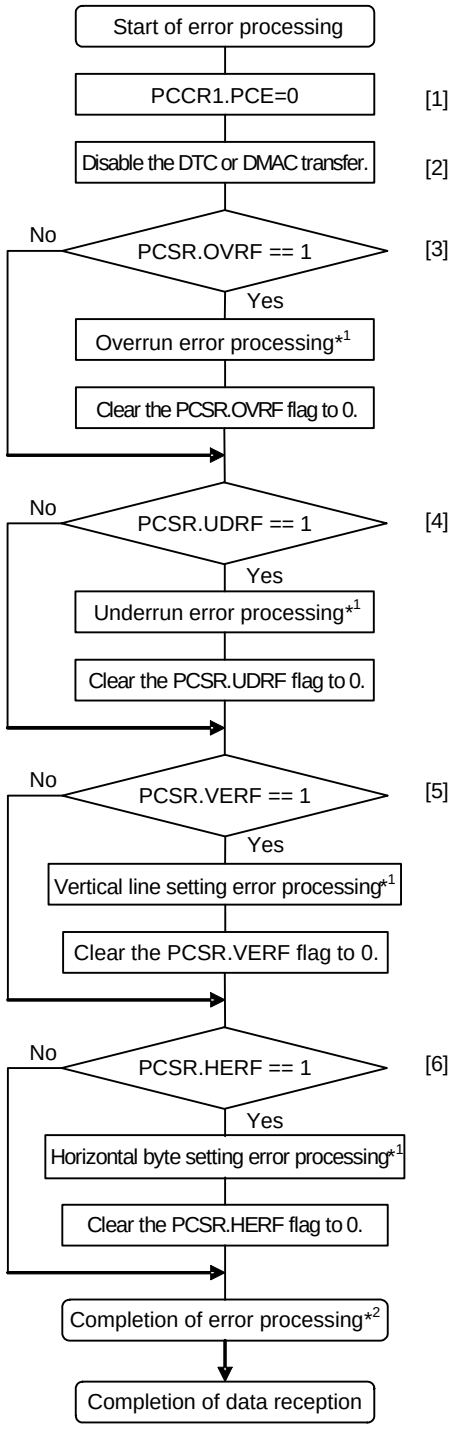
[4] Transfer by the DTC or DMAC:
Transfer the data in the PCDR register to the destination address in response to a receive data ready interrupt (DTC/DMAC startup).
Read the PCDR register in longword access and transfer 32 bytes of data per receive data ready interrupt.

[5] Frame end interrupt:
Read the FEMPF flag in the PCSR register and verify completion of the DTC or DMAC transfer.
If an underrun occurs during DTC or DMAC transfer, setting of the FEMPF flag may not proceed. When an underrun occurs, clear the FEF flag to 0 and branch to the appropriate error processing routine.

[6] Completion of data reception:
After completion of the transfer, clear the PCE bit in the PCCR1 register to 0, and then clear the FEF flag in the PCSR register to 0. This completes handling of the frame end interrupt and data reception.

Figure 51.19 Example of Flows of Operations

Figure 51.20 shows an example of the flow of operations in response to an error interrupt.



[1] Disabling of PDC operation:
Clear the PCE bit in the PCCR1 register to 0 to disable PDC operation.

[2] Disabling of the DTC or DMAC transfer:
Disable the transfer of the DTC or DMAC.

[3] Checking the overrun flag:
Read the PCSR register to check the state of The OVRF flag. If the flag is set, processing to handle the overrun error proceeds and the flag is then cleared.

[4] Checking the underrun flag:
Read the PCSR register to check the state of The UDRF flag. If the flag is set, processing to handle the underrun error proceeds and the flag is then cleared.

[5] Checking the vertical line setting error flag:
Read the PCSR register to check the state of The VERF flag. If the flag is set, processing to handle the vertical line setting error proceeds and the flag is then cleared.

[6] Checking the horizontal byte setting error flag:
Read the PCSR register to check the status of The HERF flag. If the flag is set, processing to handle the horizontal byte setting error proceeds and the flag is then cleared.

Note 1. Processing for each type of error is defined by the user, so the user needs to prepare error processing for each.

Note 2. If the CPU, DTC, or DMAC reads the PCDR register after an error interrupt has occurred in response to an interrupt source other than an underrun, an underrun may occur. If an underrun occurs on completion of error processing, run the error processing again and clear the UDRF flag.
Since an underrun after error processing still requires detection, do not prohibit the underrun interrupt before the error processing.

Figure 51.20 Example of Error Processing Flow

51.3.10 Interrupt Source

The interrupt sources of the PDC are receive data ready, frame end, overrun, underrun, vertical line number setting error, and horizontal byte number setting error. The PDC can start up the DTC or DMAC in response to a receive data ready interrupt request for the transfer of data.

Table 51.5 summarizes the interrupt sources of the PDC. When an interrupt condition listed in Table 51.5 is satisfied, the corresponding interrupt is generated. In the case of receive data ready, the interrupt source flag is cleared by reading of the PCDR register. For the interrupt source flag for frame end, the FEF flag in the PCSR register is cleared. For an overrun, underrun, vertical line number setting error, or a horizontal byte number setting error, the flags must be checked to identify the error source flag because their interrupt vectors are allocated to the same address by the PCERI. Once identified, the corresponding error interrupt source flag (the OVRF, UDRF, VERF, or HERF flag) in the PCSR register should be cleared.

When the DTC or DMAC is to handle data transfer, whichever is selected should be set up first. After enabling the block to handle transfer, set up the PDC. For descriptions of how to set up the DTC and DMAC, see section 18, DMA Controller (DMACAa) and section 20, Data Transfer Controller (DTCa).

In addition, for receive data ready, if the ICU.IRn.IR flag is 1, the interrupt request signal is retained internally but an interrupt request is not output to the ICU, even if the interrupt condition is satisfied. In this case, however, only one interrupt request signal can be retained internally. If the ICU.IRn.IR flag is 0, the retained interrupt request signal is output to the ICU. After completion of output, the request flag is automatically cleared. An interrupt request signal retained internally can also be cleared by setting the corresponding interrupt enable bit (the DFIE bit in the PCCR0 register) to 0.

When the receive data ready interrupt is disabled from the enabled state, set the ICU.IRn.IR flag to 0 according to the following procedure.

1. Set the IERn.IENj bit to the value that disables interrupt requests.
2. Set the receive data ready interrupt enable bit (the DFIE bit) to “disabled” and read out the register after writing to verify that the bit has the new value.
3. Set the ICU.IRn.IR flag to 0.

Table 51.5 PDC Interrupt Source

Interrupt Source	Abbre -viation	Interrupt Conditions	DTC/DMAC Startup
Receive Data Ready	PCDFI	Receive data ready occurring while the DFIE bit in the PCCR0 register is 1.	Possible
Frame End	PCFEI	Frame end occurring while the FEIE bit in the PCCR0 register is 1.	Impossible
Errors	PCERI	An overrun occurring while the OVIE bit in the PCCR0 register is 1. An underrun occurring while the UDRIE bit in the PCCR0 register is 1. A vertical line number setting error occurring while the VERIE bit in the PCCR0 register is 1. A horizontal byte number setting error occurring while the HERIE bit in the PCCR0 register is 1.	Impossible

51.3.11 Reset State

Resetting of the PDC is divided into two types, PDC reset and other reset. Other resets include RES# pin reset, power-on reset, voltage monitoring reset 0, voltage monitoring reset 1, voltage monitoring reset 2, deep software standby reset, independent watchdog timer reset, watchdog timer reset, and software reset. Table 51.6 shows the range and states following the two types of reset.

Table 51.6 Reset State

	PDC Reset	Other reset
PCCR0	Retained	Reset
PCCR1	Retained	Reset
PCSR	Reset	Reset
PCMONR	Retained	Reset
PCDR	Retained	Reset
VCR	Retained	Reset
HCR	Retained	Reset

51.4 Usage Notes

51.4.1 Setting of the Module Stop Function

The module stop control register B (MSTPCRB) enables or disables operation of the PDC. Operation of the PDC is stopped when the value is that following a reset. The register becomes accessible following release from the module stop state. For details, see section 11, Low Power Consumption.

51.4.2 Notes on the Power Consumption Reduction Function

When power consumption by the PDC is to be reduced by using the power consumption reduction function, set the PCE bit in the PCCR1 register to 0 to disable operations for reception, and set the PCKE bit in the PCCR0 register to 0 to disable input through the PIXCLK pin. After that, use the power consumption reduction function.
If the setting of the PCKOE bit in the PCCR0 register is 1, it should be set to 0 to stop output of the PCKO signal, and input through the PIXCLK pin should be disabled in the way described above. After that, use the power consumption reduction function.
On return from the low power consumption state, set the PCKE bit to 1, and after enabling input through the PIXCLK pin, use the PRST bit to initialize the PDC.

51.4.3 Notes on Error Interrupts

When an error interrupt occurs, there is a possibility that the DTC or DMAC might be transmitting parallel data depending on their operation state. Therefore, transmitting executed by the DTC or DMAC should be prohibited after the PDC operation is prohibited at the top of the error interrupt processing routine (PCCR1.PCE = 0).

51.4.4 Notes on Use of the DTC

When the DTC is used with the receive data ready interrupt, set the DISEL bit in the MRB register to 0 and the SZ bit in the MRA register to 10b.
The maximum number of blocks the DTC can transfer in block transfer mode is 65536. If 32 bytes are transferred per block transfer, this represents a total of up to 2097152 bytes. If more data are to be transferred, set up the DTC again during the horizontal blanking period.

51.4.5 Notes on Use of the DMAC

When the DMAC is used with the receive data ready interrupt, set the DISEL bit in the DMCSL register to 0, set the SZ bit in the DMTMD register to 10b, and set the IPRn.IPR[3:0] bit responding to an ICU receive data ready interrupt to level 0 (the interrupt is prohibited).
The maximum number of blocks the DMAC can transfer in block transfer mode is 65536. If 32 bytes are transferred per block transfer, this represents a total of up to 2097152 bytes. If more data are to be transferred, set up the DMAC again during the horizontal blanking period.

51.4.6 Notes on Start of Transfer

When transfer starts while the ICU.IRn.IR flag is 1, an interrupt request signal is internally retained after the start of transfer. This may cause the ICU.IRn.IR flag to behave in an unexpected way.
In such cases, therefore, clear the interrupt request flag by following the procedures below before enabling operations (i.e. before setting the PCE bit in the PCCR1 to 1).

(1) Ensure that transfer is stopped (the PCE bit in the PCCR1 is 0).
(2) Set the corresponding interrupt enable bit (the DFIE bit in the PCCR0 register) to 0.
(3) Read out the corresponding interrupt enable bit (the DFIE bit in the PCCR0 register) to ensure that it is 0.
(4) Set the ICU.IRn.IR flag to 0.

52. Boundary Scan

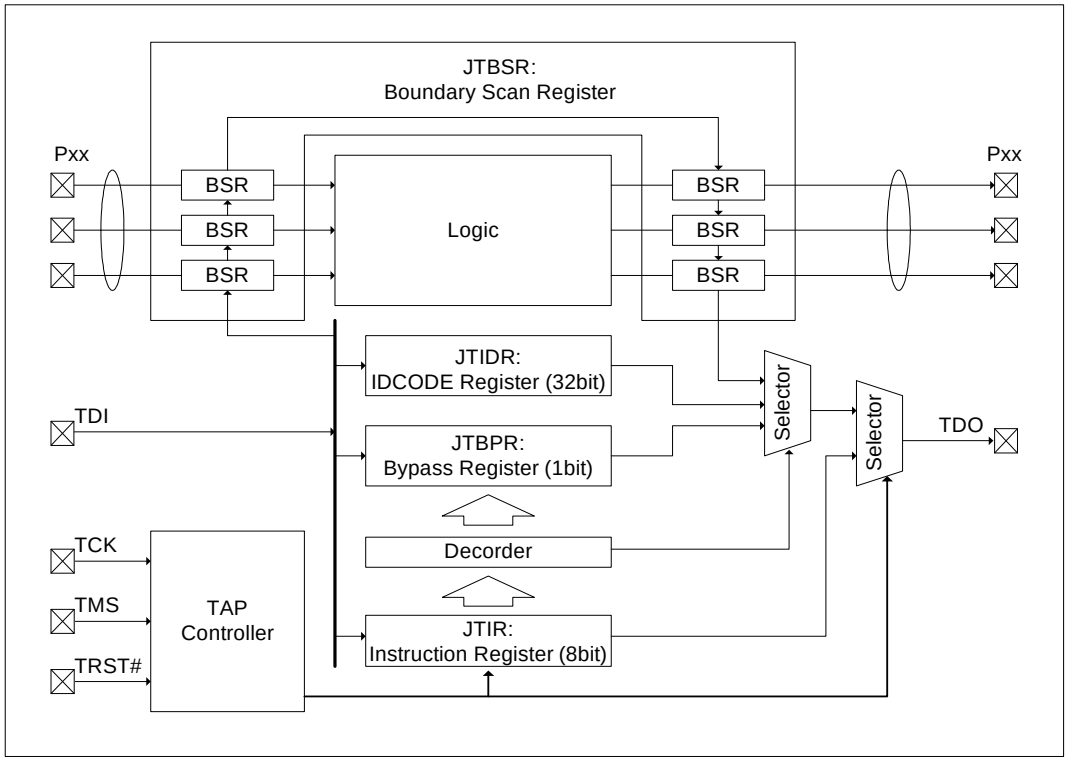
This MCU has boundary scan function.
The boundary scan is a serial I/O interface based on the JTAG (Joint Test Action Group, IEEE Std.1149.1 and IEEE Standard Test Access Port and Boundary-Scan Architecture).

52.1 Overview

Table 52.1 lists the specifications of boundary scan.
Figure 52.1 shows a block diagram of the boundary scan function.

Table 52.1 Specifications of Boundary Scan

Item	Description
Boundary scan enabled/disabled	Boundary scan is enabled when the RES# pin and the BSCANP pin are driven high and the EMLE pin is driven low.
Dedicated boundary scan pins	The following pins are dedicated for the JTAG, when boundary scan function is enabled (TDO/TCK/TDI/TMS/TRST#). 177-pin TFLGA/176-pin LFBGA: PF0/PF1/PF2/PF3/PF4 145-pin TFLGA: P26/P27/P30/P31/P34
Six test modes	• BYPASS mode • EXTEST mode • SAMPLE/PRELOAD mode • CLAMP mode • HIGHZ mode • IDCODE mode



JTBPR: Bypass register
JTBSR: Boundary scan register
JTIR: Instruction register
JTIDR: ID code register

Figure 52.1 Block Diagram of JTAG

Table 52.2 shows the I/O pins used in the boundary scan function.

Table 52.2 Pin Configuration

Pin Name	I/O	Description
TCK	Input	Test clock input pin Clock signal for boundary scan. Input the clock the duty cycle of which is 50 percent when boundary scan function is used.
TMS	Input	Test mode select pin
TDI	Input	Test data input pin
TDO	Output	Test data output pin
TRST#	Input	Test reset input pin

52.2 Register Descriptions

Table 52.3 lists the boundary scan registers.

Table 52.3 List of Boundary Scan Registers

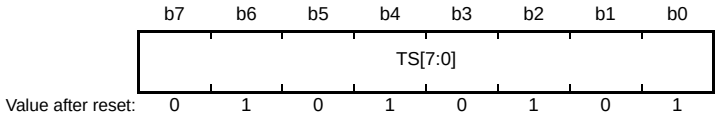
Register Name	Symbol	Value after Reset
Instruction register	JTIR	55h
ID code register	JTIDR	0819 3447h
Bypass register	JTBPR	Undefined
Boundary scan register	JTBSR	Undefined

Instructions can be input to the JTIR register via the TDI pin by serial transfer.
The JTBPR register, which is a 1-bit register, is connected between the TDI and TDO pins in BYPASS mode.
The JTBSR register, which is configured according to Table 52.6 and Table 52.7, is connected between the TDI and TDO pins when test data are being shifted in.
None of the registers is accessible from the CPU.
Table 52.4 shows the availability of serial transfer for the registers.

Table 52.4 Serial Transfer for the Registers

Register Name	Serial Input	Serial Output
Instruction register (JTIR)	Available	Available
ID code register (JTIDR)	Available	Available
Bypass register (JTBPR)	Available	Available
Boundary scan register (JTBSR)	Available	Available

52.2.1 Instruction Register (JTIR)



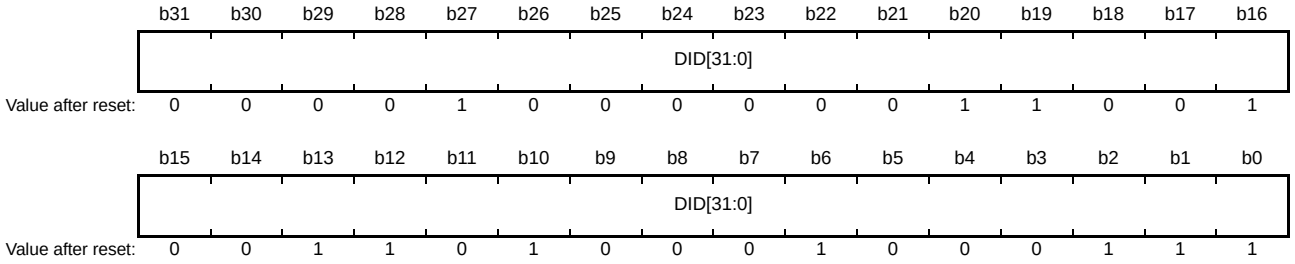
Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TS[7:0]	Test Bit Set	The command configuration is as shown in Table 52.5.	—

Table 52.5 Command Configuration

TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0	Instruction
0	0	0	0	0	0	0	0	EXTEST
0	1	0	0	0	0	0	0	SAMPLE/PRELOAD
0	1	0	1	0	1	0	1	IDCODE (initial value)
1	1	0	1	0	0	0	0	CLAMP
1	0	0	0	0	0	0	0	HIGHZ
1	1	1	1	1	1	1	1	BYPASS
Other than above								Reserved

JTAG instructions can be transferred to the JTIR register by serial input from the TDI pin.
The JTIR register is initialized when the TRST# pin is driven low, or when the TAP controller is in the Test-Logic-Reset state.

52.2.2 ID Code Register (JTIDR)



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	DID[31:0]	Reserved	JTIDR is a register with the fixed value that indicates the device IDCODE.	—

JTIDR data is output from the TDO pin when the IDCODE instruction has been executed.

52.2.3 Bypass Register (JTBPR)

The JTBPR register is a 1-bit register and is connected between the TDI and TDO pins when the JTIR register is set to BYPASS mode.
The JTBPR register cannot be read from or written to by the CPU.

52.2.4 Boundary Scan Register (JTBSR)

The JTBSR register is a shift register to control the external input and output pins of this LSI and is distributed across the pads.
The EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ instructions are issued to apply the JTBSR register in boundary-scan testing.
Table 52.6 and Table 52.7 show the correspondence between the JTBSR bits and the pins of this LSI.
The value after reset is undefined.

Table 52.6 Boundary Scan Register 177-Pin TFLGA/176-Pin LFBGA (1/9)

Pin No.	Pin Name	Input/Output	Bit Name
From TDI			
B1	P05	Output	390
		Output enable	389
		Input	388
D3	P03	Output	387
		Output enable	386
		Input	385
D2	P02	Output	384
		Output enable	383
		Input	382
D1	P01	Output	381
		Output enable	380
		Input	379
D4	P00	Output	378
		Output enable	377
		Input	376
E3	PF5	Output	375
		Output enable	374
		Input	373
E1	PJ5	Output	372
		Output enable	371
		Input	370
F3	PJ3	Output	369
		Output enable	368
		Input	367
H4	P35	Input	363
J3	P34	Output	362
		Output enable	361
		Input	360
K1	P33	Output	359
		Output enable	358
		Input	357
K2	P32	Output	356
		Output enable	355
		Input	354
L1	P31	Output	347
		Output enable	346
		Input	345
L2	P30	Output	344
		Output enable	343
		Input	342
M1	P27	Output	335
		Output enable	334
		Input	333

Table 52.6 Boundary Scan Register 177-Pin TFLGA/176-Pin LFBGA (2/9)

Pin No.	Pin Name	Input/Output	Bit Name
M2	P26	Output	332
		Output enable	331
		Input	330
L4	P25	Output	329
		Output enable	328
		Input	327
M3	P24	Output	326
		Output enable	325
		Input	324
N2	P23	Output	323
		Output enable	322
		Input	321
N3	P22	Output	320
		Output enable	319
		Input	318
R1	P21	Output	317
		Output enable	316
		Input	315
R2	P20	Output	308
		Output enable	307
		Input	306
P2	P17	Output	305
		Output enable	304
		Input	303
P3	P87	Output	302
		Output enable	301
		Input	300
R3	P16	Output	299
		Output enable	298
		Input	297
M4	P86	Output	296
		Output enable	295
		Input	294
N4	P15	Output	293
		Output enable	292
		Input	291
P4	P14	Output	290
		Output enable	289
		Input	288
R4	P13	Output	287
		Output enable	286
		Input	285
N5	P12	Output	284
		Output enable	283
		Input	282

Table 52.6 Boundary Scan Register 177-Pin TFLGA/176-Pin LFBGA (3/9)

Pin No.	Pin Name	Input/Output	Bit Name
R8	P11	Output	272
		Output enable	271
		Input	270
P8	P10	Output	269
		Output enable	268
		Input	267
R9	P53	Output	266
		Output enable	265
		Input	264
P9	P52	Output	263
		Output enable	262
		Input	261
N9	P51	Output	260
		Output enable	259
		Input	258
M9	P50	Output	257
		Output enable	256
		Input	255
P10	P83	Output	254
		Output enable	253
		Input	252
N10	PC7	Output	251
		Output enable	250
		Input	249
P11	PC6	Output	248
		Output enable	247
		Input	246
M10	PC5	Output	245
		Output enable	244
		Input	243
N11	P82	Output	242
		Output enable	241
		Input	240
M11	P81	Output	239
		Output enable	238
		Input	237
R12	P80	Output	236
		Output enable	235
		Input	234
P12	PC4	Output	233
		Output enable	232
		Input	231
N12	PC3	Output	230
		Output enable	229
		Input	228

Table 52.6 Boundary Scan Register 177-Pin TFLGA/176-Pin LFBGA (4/9)

Pin No.	Pin Name	Input/Output	Bit Name
M12	P77	Output	227
		Output enable	226
		Input	225
R13	P76	Output	224
		Output enable	223
		Input	222
P13	PC2	Output	221
		Output enable	220
		Input	219
P14	P75	Output	218
		Output enable	217
		Input	216
R14	P74	Output	215
		Output enable	214
		Input	213
R15	PC1	Output	212
		Output enable	211
		Input	210
N13	PC0	Output	209
		Output enable	208
		Input	207
N14	P73	Output	206
		Output enable	205
		Input	204
M13	PB7	Output	203
		Output enable	202
		Input	201
L12	PB6	Output	200
		Output enable	199
		Input	198
M14	PB5	Output	197
		Output enable	196
		Input	195
M15	PB4	Output	194
		Output enable	193
		Input	192
L13	PB3	Output	191
		Output enable	190
		Input	189
K12	PB2	Output	188
		Output enable	187
		Input	186
L14	PB1	Output	185
		Output enable	184
		Input	183

Table 52.6 Boundary Scan Register 177-Pin TFLGA/176-Pin LFBGA (5/9)

Pin No.	Pin Name	Input/Output	Bit Name
L15	P72	Output	182
		Output enable	181
		Input	180
K13	P71	Output	179
		Output enable	178
		Input	177
K15	PB0	Output	176
		Output enable	175
		Input	174
J14	PA7	Output	173
		Output enable	172
		Input	171
J15	PA6	Output	170
		Output enable	169
		Input	168
J12	PA5	Output	167
		Output enable	166
		Input	165
H12	PA4	Output	164
		Output enable	163
		Input	162
H13	PA3	Output	161
		Output enable	160
		Input	159
H15	PG7	Output	158
		Output enable	157
		Input	156
H14	PA2	Output	155
		Output enable	154
		Input	153
G13	PG6	Output	152
		Output enable	151
		Input	150
G14	PA1	Output	149
		Output enable	148
		Input	147
G12	PG5	Output	146
		Output enable	145
		Input	144
F14	PA0	Output	143
		Output enable	142
		Input	141
F13	PG4	Output	140
		Output enable	139
		Input	138

Table 52.6 Boundary Scan Register 177-Pin TFLGA/176-Pin LFBGA (6/9)

Pin No.	Pin Name	Input/Output	Bit Name
E15	P67	Output	137
		Output enable	136
		Input	135
E14	PG3	Output	134
		Output enable	133
		Input	132
F12	P66	Output	131
		Output enable	130
		Input	129
E13	PG2	Output	128
		Output enable	127
		Input	126
D15	P65	Output	125
		Output enable	124
		Input	123
D14	PE7	Output	122
		Output enable	121
		Input	120
E12	PE6	Output	119
		Output enable	118
		Input	117
C15	P70	Output	116
		Output enable	115
		Input	114
D12	PE5	Output	113
		Output enable	112
		Input	111
C13	PE4	Output	110
		Output enable	109
		Input	108
B15	PE3	Output	107
		Output enable	106
		Input	105
A15	PE2	Output	104
		Output enable	103
		Input	102
A14	PE1	Output	101
		Output enable	100
		Input	99
B14	PE0	Output	98
		Output enable	97
		Input	96
B13	P64	Output	95
		Output enable	94
		Input	93

Table 52.6 Boundary Scan Register 177-Pin TFLGA/176-Pin LFBGA (7/9)

Pin No.	Pin Name	Input/Output	Bit Name
A13	P63	Output	92
		Output enable	91
		Input	90
C12	P62	Output	89
		Output enable	88
		Input	87
D11	P61	Output	86
		Output enable	85
		Input	84
A12	P60	Output	83
		Output enable	82
		Input	81
D10	PD7	Output	80
		Output enable	79
		Input	78
B11	PG1	Output	77
		Output enable	76
		Input	75
A11	PD6	Output	74
		Output enable	73
		Input	72
C10	PG0	Output	71
		Output enable	70
		Input	69
D9	PD5	Output	68
		Output enable	67
		Input	66
B10	PD4	Output	65
		Output enable	64
		Input	63
A10	P97	Output	62
		Output enable	61
		Input	60
C9	PD3	Output	59
		Output enable	58
		Input	57
B9	P96	Output	56
		Output enable	55
		Input	54
C8	PD2	Output	53
		Output enable	52
		Input	51
D7	P95	Output	50
		Output enable	49
		Input	48

Table 52.6 Boundary Scan Register 177-Pin TFLGA/176-Pin LFBGA (8/9)

Pin No.	Pin Name	Input/Output	Bit Name
B8	PD1	Output	47
		Output enable	46
		Input	45
A8	P94	Output	44
		Output enable	43
		Input	42
C7	PD0	Output	41
		Output enable	40
		Input	39
D6	P93	Output	38
		Output enable	37
		Input	36
B7	P92	Output	35
		Output enable	34
		Input	33
B6	P91	Output	32
		Output enable	31
		Input	30
C6	P90	Output	29
		Output enable	28
		Input	27
B5	P47	Output	26
		Output enable	25
		Input	24
A5	P46	Output	23
		Output enable	22
		Input	21
C5	P45	Output	20
		Output enable	19
		Input	18
D5	P44	Output	17
		Output enable	16
		Input	15
C4	P43	Output	14
		Output enable	13
		Input	12
A4	P42	Output	11
		Output enable	10
		Input	9
B4	P41	Output	8
		Output enable	7
		Input	6
B3	P40	Output	5
		Output enable	4
		Input	3

Table 52.6 Boundary Scan Register 177-Pin TFLGA/176-Pin LFBGA (9/9)

Pin No.	Pin Name	Input/Output	Bit Name
B2	P07	Output	2
		Output enable	1
		Input	0
To TDO			

Table 52.7 Boundary Scan Register 145-Pin TFLGA (1/8)

Pin No.	Pin Name	Input/Output	Bit Name
From TDI			
B3	P05	Output	390
		Output enable	389
		Input	388
D3	P03	Output	387
		Output enable	386
		Input	385
C2	P02	Output	384
		Output enable	383
		Input	382
D4	P01	Output	381
		Output enable	380
		Input	379
D1	P00	Output	378
		Output enable	377
		Input	376
D2	PF5	Output	375
		Output enable	374
		Input	373
E3	PJ5	Output	372
		Output enable	371
		Input	370
F3	PJ3	Output	369
		Output enable	368
		Input	367
H4	P35	Input	363
J2	P33	Output	359
		Output enable	358
		Input	357
J3	P32	Output	356
		Output enable	355
		Input	354
L1	P25	Output	329
		Output enable	328
		Input	327
L4	P24	Output	326
		Output enable	325
		Input	324
L2	P23	Output	323
		Output enable	322
		Input	321
M1	P22	Output	320
		Output enable	319
		Input	318

Table 52.7 Boundary Scan Register 145-Pin TFLGA (2/8)

Pin No.	Pin Name	Input/Output	Bit Name
N1	P21	Output	317
		Output enable	316
		Input	315
N2	P20	Output	308
		Output enable	307
		Input	306
M2	P17	Output	305
		Output enable	304
		Input	303
N3	P87	Output	302
		Output enable	301
		Input	300
L3	P16	Output	299
		Output enable	298
		Input	297
M3	P86	Output	296
		Output enable	295
		Input	294
K4	P15	Output	293
		Output enable	292
		Input	291
N4	P14	Output	290
		Output enable	289
		Input	288
L5	P13	Output	287
		Output enable	286
		Input	285
M4	P12	Output	284
		Output enable	283
		Input	282
L6	P56	Output	281
		Output enable	280
		Input	279
N7	P55	Output	278
		Output enable	277
		Input	276
K5	P54	Output	275
		Output enable	274
		Input	273
K6	P53	Output	266
		Output enable	265
		Input	264

Table 52.7 Boundary Scan Register 145-Pin TFLGA (3/8)

Pin No.	Pin Name	Input/Output	Bit Name
L7	P52	Output	263
		Output enable	262
		Input	261
K7	P51	Output	260
		Output enable	259
		Input	258
M7	P50	Output	257
		Output enable	256
		Input	255
L8	P83	Output	254
		Output enable	253
		Input	252
N9	PC7	Output	251
		Output enable	250
		Input	249
M8	PC6	Output	248
		Output enable	247
		Input	246
L9	PC5	Output	245
		Output enable	244
		Input	243
N10	P82	Output	242
		Output enable	241
		Input	240
M9	P81	Output	239
		Output enable	238
		Input	237
K9	P80	Output	236
		Output enable	235
		Input	234
L10	PC4	Output	233
		Output enable	232
		Input	231
N11	PC3	Output	230
		Output enable	229
		Input	228
M10	P77	Output	227
		Output enable	226
		Input	225
K10	P76	Output	224
		Output enable	223
		Input	222
L11	PC2	Output	221
		Output enable	220
		Input	219

Table 52.7 Boundary Scan Register 145-Pin TFLGA (4/8)

Pin No.	Pin Name	Input/Output	Bit Name
N12	P75	Output	218
		Output enable	217
		Input	216
N13	P74	Output	215
		Output enable	214
		Input	213
M12	PC1	Output	212
		Output enable	211
		Input	210
M11	PC0	Output	209
		Output enable	208
		Input	207
L12	P73	Output	206
		Output enable	205
		Input	204
K11	PB7	Output	203
		Output enable	202
		Input	201
K12	PB6	Output	200
		Output enable	199
		Input	198
K13	PB5	Output	197
		Output enable	196
		Input	195
J11	PB4	Output	194
		Output enable	193
		Input	192
J10	PB3	Output	191
		Output enable	190
		Input	189
J12	PB2	Output	188
		Output enable	187
		Input	186
J13	PB1	Output	185
		Output enable	184
		Input	183
H10	P72	Output	182
		Output enable	181
		Input	180
H11	P71	Output	179
		Output enable	178
		Input	177
H12	PB0	Output	176
		Output enable	175
		Input	174

Table 52.7 Boundary Scan Register 145-Pin TFLGA (5/8)

Pin No.	Pin Name	Input/Output	Bit Name
H13	PA7	Output	173
		Output enable	172
		Input	171
G11	PA6	Output	170
		Output enable	169
		Input	168
G10	PA5	Output	167
		Output enable	166
		Input	165
G13	PA4	Output	164
		Output enable	163
		Input	162
F10	PA3	Output	161
		Output enable	160
		Input	159
F13	PA2	Output	155
		Output enable	154
		Input	153
F12	PA1	Output	149
		Output enable	148
		Input	147
E10	PA0	Output	143
		Output enable	142
		Input	141
E13	P67	Output	137
		Output enable	136
		Input	135
E11	P66	Output	131
		Output enable	130
		Input	129
E12	P65	Output	125
		Output enable	124
		Input	123
D10	PE7	Output	122
		Output enable	121
		Input	120
D13	PE6	Output	119
		Output enable	118
		Input	117
C12	P70	Output	116
		Output enable	115
		Input	114
D12	PE5	Output	113
		Output enable	112
		Input	111

Table 52.7 Boundary Scan Register 145-Pin TFLGA (6/8)

Pin No.	Pin Name	Input/Output	Bit Name
B13	PE4	Output	110
		Output enable	109
		Input	108
A13	PE3	Output	107
		Output enable	106
		Input	105
B12	PE2	Output	104
		Output enable	103
		Input	102
A12	PE1	Output	101
		Output enable	100
		Input	99
C11	PE0	Output	98
		Output enable	97
		Input	96
D9	P64	Output	95
		Output enable	94
		Input	93
C10	P63	Output	92
		Output enable	91
		Input	90
A11	P62	Output	89
		Output enable	88
		Input	87
B11	P61	Output	86
		Output enable	85
		Input	84
D8	P60	Output	83
		Output enable	82
		Input	81
C9	PD7	Output	80
		Output enable	79
		Input	78
A9	PD6	Output	74
		Output enable	73
		Input	72
D7	PD5	Output	68
		Output enable	67
		Input	66
B9	PD4	Output	65
		Output enable	64
		Input	63
C8	PD3	Output	59
		Output enable	58
		Input	57

Table 52.7 Boundary Scan Register 145-Pin TFLGA (7/8)

Pin No.	Pin Name	Input/Output	Bit Name
A8	PD2	Output	53
		Output enable	52
		Input	51
C7	PD1	Output	47
		Output enable	46
		Input	45
B8	PD0	Output	41
		Output enable	40
		Input	39
D6	P93	Output	38
		Output enable	37
		Input	36
A7	P92	Output	35
		Output enable	34
		Input	33
B7	P91	Output	32
		Output enable	31
		Input	30
A6	P90	Output	29
		Output enable	28
		Input	27
B6	P47	Output	26
		Output enable	25
		Input	24
C5	P46	Output	23
		Output enable	22
		Input	21
A5	P45	Output	20
		Output enable	19
		Input	18
E5	P44	Output	17
		Output enable	16
		Input	15
B5	P43	Output	14
		Output enable	13
		Input	12
A4	P42	Output	11
		Output enable	10
		Input	9
C4	P41	Output	8
		Output enable	7
		Input	6
A3	P40	Output	5
		Output enable	4
		Input	3

Table 52.7 Boundary Scan Register 145-Pin TFLGA (8/8)

Pin No.	Pin Name	Input/Output	Bit Name
A2	P07	Output	2
		Output enable	1
		Input	0
To TDO			

52.3 Operations

The boundary scan functionality is valid when the RES# pin is driven high, the EMLE pin is driven low, and the BSCANP pin is driven high.

52.3.1 TAP Controller

Figure 52.2 shows the state transition diagram of the TAP controller.

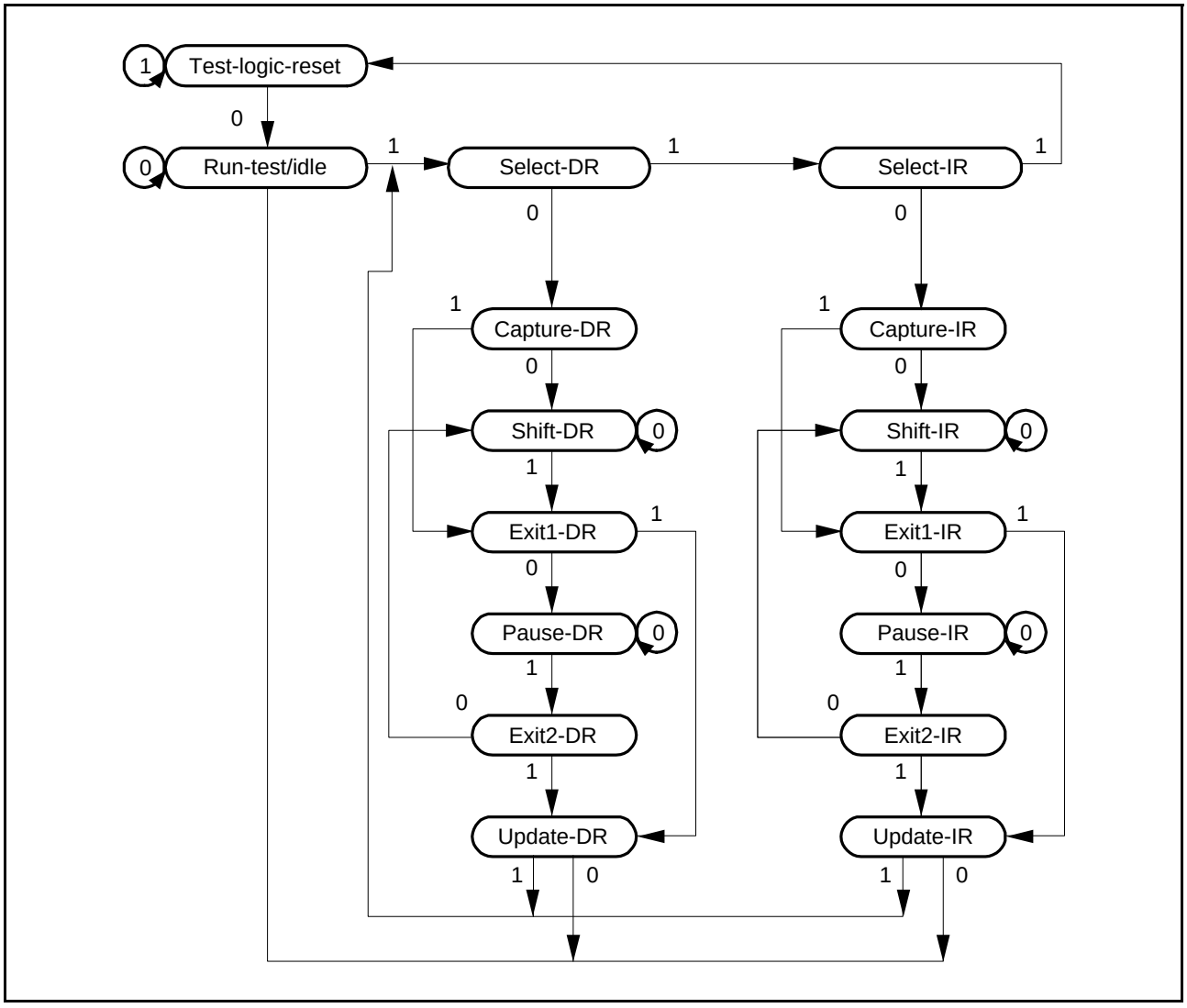


Figure 52.2 State Transition Diagram of TAP Controller

52.3.2 List of Commands

(1) BYPASS [Instruction Code: 1111 1111b]

The BYPASS instruction is an instruction that drives the bypass register (JTBPR). This instruction shortens the shift path, facilitating the transfer of serial data to other LSIs on a printed-circuit board at higher speeds. While this instruction is being executed, the test circuit has no effect on the system circuits.
The bypass register (JTBPR) is connected between the TDI and TDO pins. Bypass operation is initiated from shift-DR operation. The TDO is low in the first clock cycle in the shift-DR state; in the subsequent clock cycles, the TDI signal is output on the TDO pin.

(2) EXTEST [Instruction Code: 0000 0000b]

The EXTEST instruction is used to test external circuits when this LSI is installed on the printed circuit board. If this instruction is executed, output pins are used to output test data (specified by the SAMPLE/PRELOAD instruction) from the boundary scan register to the print circuit board, and input pins are used to input test result.

(3) SAMPLE/PRELOAD [Instruction Code: 0100 0000b]

The SAMPLE/PRELOAD instruction is used to input data from the LSI internal circuits to the boundary scan register, output data from scan path, and reload the data to the scan path. While this instruction is executed, input signals are directly input to the LSI and output signals are also directly output to the external circuits. The LSI system circuit is not affected by this instruction.
In SAMPLE operation, the boundary scan register latches the snap shot of data transferred from input pins to internal circuit or data transferred from internal circuit to output pins. The latched data is read from the scan path. The scan register latches the snap data at the rising edge of the TCK in Capture-DR state. The scan register latches snap shot without affecting the LSI normal operation.
In PRELOAD operation, initial value is written from the scan path to the parallel output latch of the boundary scan register prior to the EXTEST instruction execution. If the EXTEST is executed without executing this PRELOAD operation, undefined values are output from the beginning to the end (transfer to the output latch) of the EXTEST sequence. (In EXTEST instruction, output parallel latches are always output to the output pins.)

(4) IDCODE [Instruction Code: 0101 0101b]

When the IDCODE instruction is selected, IDCODE register value is output to the TDO in Shift-DR state of the TAP controller. In this case, IDCODE register value is output from the LSB. During this instruction execution, test circuit does not affect the system circuit. JTIR is initialized by the IDCODE instruction in Test-Logic-Reset state of the TAP controller.

(5) CLAMP [Instruction Code: 1101 0000b]

When the CLAMP instruction is selected, output pins output the boundary scan register value which was specified by the SAMPLE/PRELOAD instruction in advance. While the CLAMP instruction is selected, the status of boundary scan register is maintained regardless of the TAP controller state.
BYPASS is connected between TDI and TDO, the same operation as BYPASS instruction can be achieved.

(6) HIGHZ [Instruction Code: 1000 0000b]

When the HIGHZ instruction is selected, all output pins enter high-impedance state. While the HIGHZ instruction is selected, the status of boundary scan register is maintained regardless of the state of the TAP controller.
BYPASS is connected between TDI and TDO pins, leading to the same operation as when the BYPASS instruction has been selected.

52.4 Usage Notes

(1) Pin serial transfer, data are input or output in LSB order (see Figure 52.3).

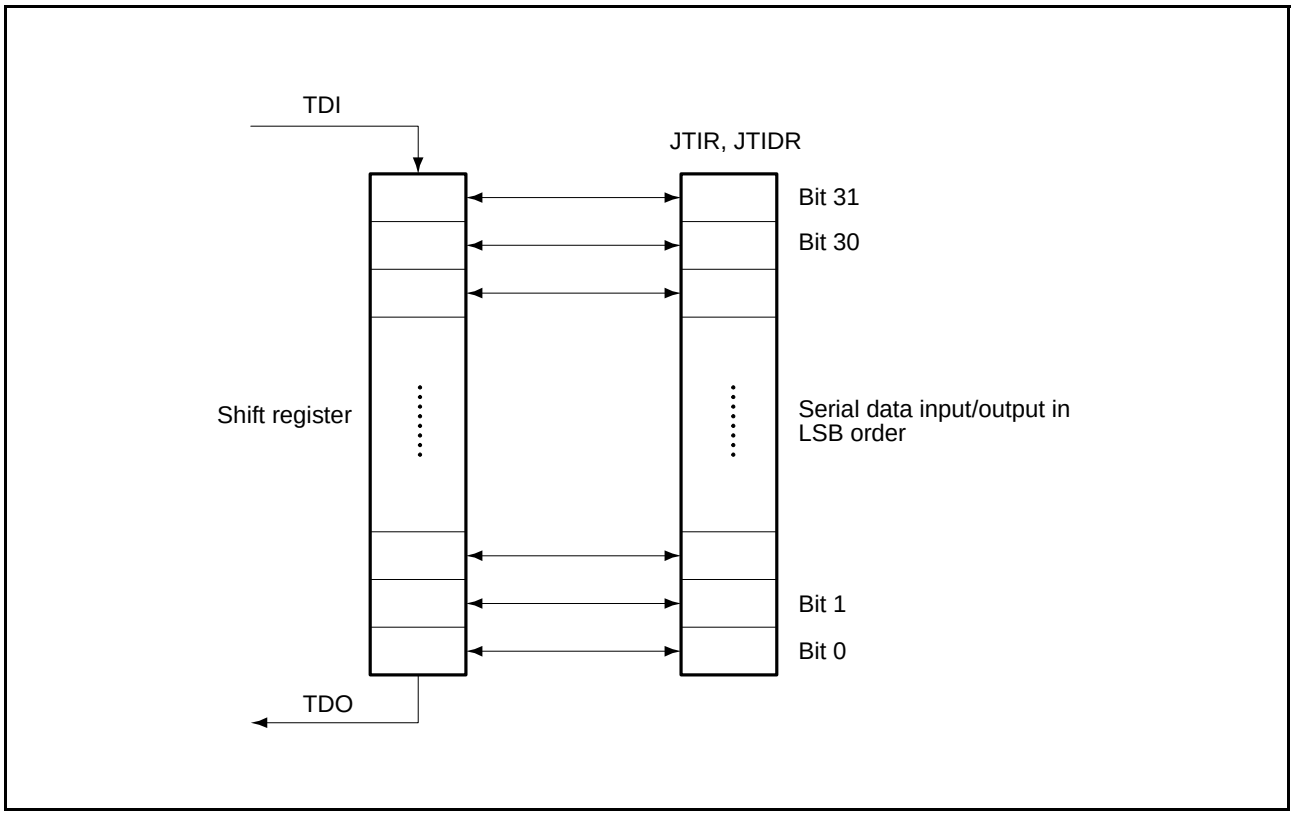


Figure 52.3 Serial Data Input/Output

(2) Pins of the boundary scan (TCK, TDI, TMS, and TRST#) have to be pulled up by pull-up resistors.
However, handle the #TRST pin in the way described in the manual for the given on-chip emulator if an on-chip emulator is in use.
If the #TRST pin is pulled down but a boundary scan is to proceed, ensure that the #TRST pin is also controllable.

(3) Power supply pins (VCC, VCL, VSS, AVCC0, AVCC1, AVSS0, AVSS1, VCC_USB, VSS_USB, AVCC_USBA, AVSS_USBA, PVSS_USBA, VCC_USBA VSS1_USBA, and VSS2_USBA) cannot be boundary-scanned.

(4) Analog reference pins (VREFH0, VREFL0, VBATT, USBA_RREF) cannot be boundary-scanned.

(5) Clock pins (EXTAL, XTAL, XCIN, and XCOUT) cannot be boundary-scanned.

(6) Reset signal (RES#) cannot be boundary-scanned.

(7) USB dedicated pins (USB0_DP, USB0_DM, USBA_DP, USBA_DM) cannot be boundary-scanned.

(8) The on-chip emulator enable pin (EMLE) cannot be boundary-scanned.

(9) The boundary-scan pin (BSCANP) cannot be boundary-scanned.

(10) The boundary-scan pins (TCK, TMS, TRST#, TDI, and TDO) cannot be boundary-scanned.

(11) The boundary-scan facility is not available when the chip is in the states below.

- Reset state
- Software standby or deep software standby

(12) For a pin that incorporates open-drain functionality and for which the open-drain function is enabled, if the boundary-scan function sets the corresponding bit in the output scan register and output enable register to 1, executing an EXTEST, CLAMP, or SAMPLE/PRELOAD instruction makes the pin output the high level rather than placing it in the high-impedance state.

(13) Be sure to satisfy the standards for the boundary scan function when multiplex ports are used. Figure 52.4 (1) shows the pin configuration of the multiplex port pins in combination with the RIIC pins (P12, P13, P16, and P17).

When the boundary scan function is used with pins P12, P13, P16, and P17 to be used as RIIC pins (SCL0[FM+], SCL2, SDA0[FM+], and SDA2), the conflict with open-drain output or sneak current might be generated.

(14) Figure 52.4 (2) shows the pin configuration of the pins P00 to P02, P40 to P47, P90 to P93, PD0 to PD7, and PE0 to PE7. When the boundary scan function is used with pins P00 to P02, P40 to P47, P90 to P93, PD0 to PD7, and PE0 to PE7 to be used as AD input pins (AN000 to AN007, ANEX0, ANEX1, and AN100 to AN120), the conflict with open-drain output or sneak current might be generated.

(15) Figure 52.4 (3) shows the pin configuration of pins P03 and P05. When the boundary scan function is used with pins P03 and P05 to be used as DA output pins (DA0 and DA1), the conflict with the DA output or sneak current might be generated.

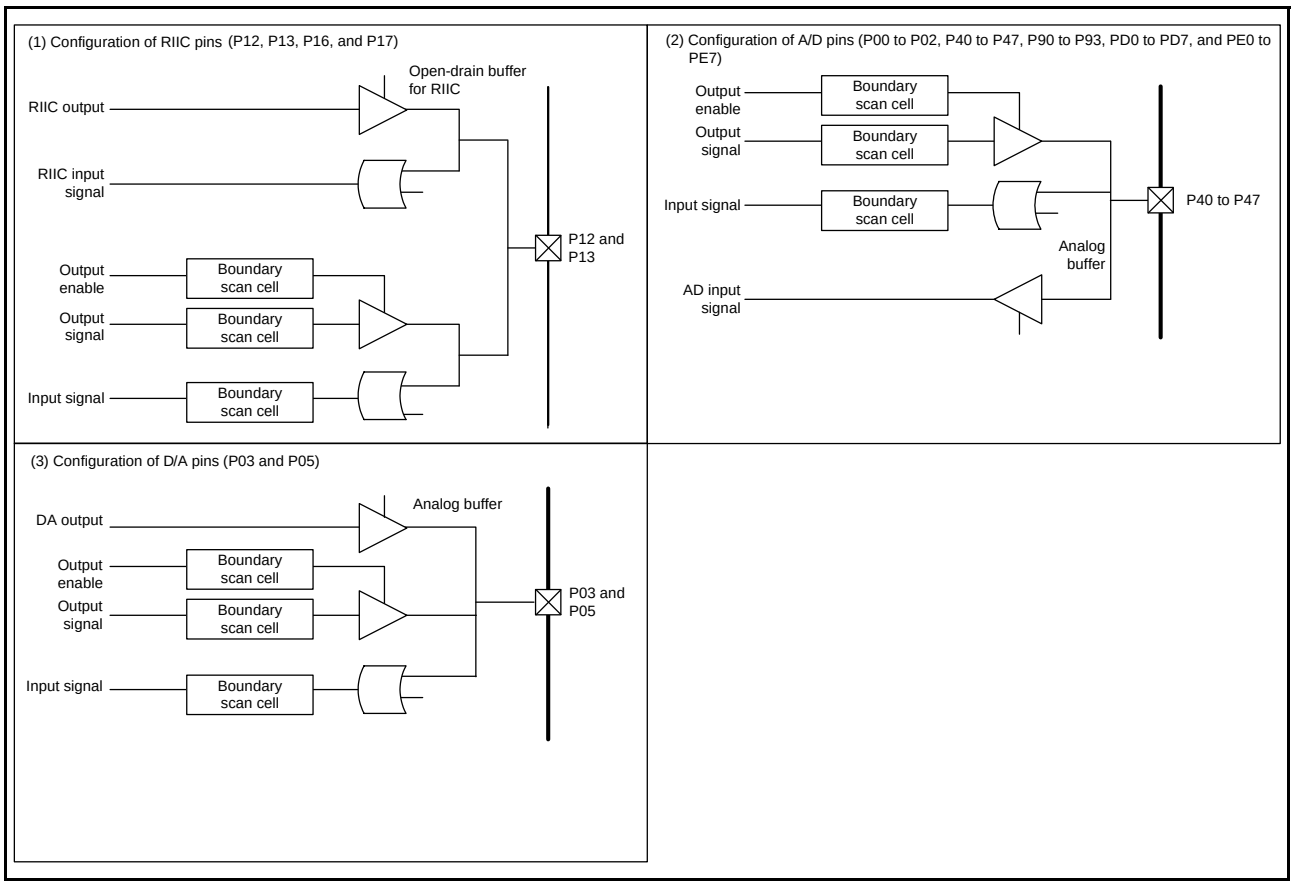


Figure 52.4 Pin Configuration

53. AES

Regarding the public release of this section, an exchange of non-disclosure agreement is necessary.
For details, contact your Renesas sales agency.

54. DES

Regarding the public release of this section, an exchange of non-disclosure agreement is necessary.
For details, contact your Renesas sales agency.

55. SHA

Regarding the public release of this section, an exchange of non-disclosure agreement is necessary.
For details, contact your Renesas sales agency.

56. RNG

Regarding the public release of this section, an exchange of non-disclosure agreement is necessary.
For details, contact your Renesas sales agency.

57. 12-Bit A/D Converter (S12ADC)

57.1 Overview

This MCU incorporates two units of a 12-bit successive approximation A/D converter. In unit 0, up to eight analog input channels are selectable. In unit 1, up to 21 analog input channels, temperature sensor output, and internal reference voltage are selectable for conversion.

The 12-bit A/D converter converts a maximum of 8 and 21 analog input channels (unit 0 and 1, respectively), temperature sensor output, and internal reference voltage, which have been selected, into a 12-bit digital value through successive approximation.

The A/D converter has three operating modes: single scan mode in which the analog inputs of up to 8 (unit 0) and 21 (unit 1) channels arbitrarily selected are converted for only once in ascending channel order; continuous scan mode in which the analog inputs of up to 8 (unit 0) and 21 (unit 1) channels arbitrarily selected are continuously converted in ascending channel order; and group scan mode in which up to 8 (unit 0) and 21 (unit 1) channels of the analog inputs are arbitrarily divided into two groups (group A and group B) and converted in ascending channel order in each group.

In group scan mode, the conditions for scanning start of group A and group B (synchronous trigger) can be independently selected, thus allowing A/D conversion of group A and group B to be started independently. When group A priority control is selected along with operation as described above, if a request to start scanning for group A (a synchronous or asynchronous trigger) is received during A/D conversion for group B, the conversion operation for group B is discontinued and the conversion for group A starts, which is given priority.

In double trigger mode, one analog input channel arbitrarily selected is converted in single scan mode or group scan mode (group A), and the resulting data of A/D conversion started by the first and second synchronous triggers are stored into different registers (duplication of A/D conversion data).

Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages internally generated in the 12-bit A/D converter is converted.

The temperature sensor output and internal reference voltage can be selected at the same time as the analog input of channels, and A/D conversion of the analog input of channels, the temperature sensor output, and the internal reference voltage is performed in that order.

A/D conversion of the extended analog input is independently performed.

Table 57.1 lists the specifications of the 12-bit A/D converter and Table 57.2 indicates the functions of the 12-bit A/D converter. Figure 57.1 shows a block diagram of the 12-bit A/D converter (unit 0) and Figure 57.2 shows a block diagram of the 12-bit A/D converter (unit 1).

Table 57.1 Specifications of 12-Bit A/D Converter (1/2)

Item	Specifications
Number of units	Two units
Input channels	Unit 0: Up to eight channels Unit 1: Up to 21 channels + one extended
Extended analog function	Temperature sensor output, internal reference voltage
A/D conversion method	Successive approximation method
Resolution	12 bits
Conversion time	0.48 µs per channel (for 12-bit conversion) 0.45 µs per channel (for 10-bit conversion) 0.42 µs per channel (for 8-bit conversion) (when A/D conversion clock ADCLK = 60 MHz)
A/D conversion clock	Peripheral module clock PCLKB*1 and A/D conversion clock ADCLK*1 can be set so that the frequency division ratio should be one of the following. PCLKB to ADCLK frequency division ratio = 1:1, 1:2, 1:4, 1:8 ADCLK is set using the clock generation circuit (CPG).
Data registers	• 29 registers for analog input (8 for unit 0 and 21 for unit 1), 1 for A/D-converted data duplication in double trigger mode in each unit, and 2 for A/D-converted data duplication during extended operation in double trigger mode in each unit • One register for temperature sensor output (in unit 1 only) • One register for internal reference voltage (in unit 1 only) • The results of A/D conversion are stored in 12-bit A/D data registers. • 8-, 10-, and 12-bit accuracy output for the results of A/D conversion • The value obtained by adding up A/D-converted results is stored as a value in the number of bit for conversion accuracy + 2 bits in the A/D data registers in A/D-converted value addition mode. • Double trigger mode (selectable in single scan and group scan modes): The first piece of A/D-converted analog-input data on one selected channel is stored in the data register for the channel, and the second piece is stored in the duplication register. • Extended operation in double trigger mode (available for specific triggers): A/D-converted analog-input data on one selected channel is stored in the duplication register that is prepared for each type of trigger.
Operating modes	• Single scan mode: A/D conversion is performed only once on the analog inputs of up to 8 (unit 0) and 21 (unit 1) channels arbitrarily selected, on the temperature sensor output (in unit 1 only), and on the internal reference voltage (in unit 1 only). A/D conversion is performed only once on the extended analog input (in unit 1 only). • Continuous scan mode: A/D conversion is performed repeatedly on the analog inputs of up to 8 (unit 0) and 21 (unit 1) channels arbitrarily selected, on the temperature sensor output (in unit 1 only), and on the internal reference voltage (in unit 1 only). A/D conversion is performed repeatedly on the extended analog input (in unit 1 only). • Group scan mode: Analog inputs of up to 8 (unit 0) and 21 (unit 1) channels arbitrarily selected, the temperature sensor output (in unit 1 only), and the internal reference voltage (in unit 1 only) are divided into group A and group B, and A/D conversion of the analog input selected on a group basis is performed only once. The conditions for scanning start of group A and group B (synchronous trigger) can be independently selected, thus allowing A/D conversion of group A and group B to be started independently. • Group scan mode (when group A is given priority): If a group A trigger (synchronous or asynchronous) is input during A/D conversion on group B, the A/D conversion on group B is stopped and A/D conversion is performed on group A. Restart (rescan) of A/D conversion on group B after completion of A/D conversion on group A can be set.
Conditions for A/D conversion start	• Software trigger • Synchronous trigger Trigger by the multi-function timer pulse unit (MTU), the general-purpose PWM timer (GPT), the event link controller (ELC), 8-bit timer (TMR), and 16-bit timer pulse unit (TPU). • Asynchronous trigger A/D conversion can be triggered by the external trigger pins, ADTRG0# (unit 0) and ADTRG1# (unit 1).

Table 57.1 Specifications of 12-Bit A/D Converter (2/2)

Item	Specifications
Function	• Sample-and-hold function • Channel-dedicated sample-and-hold function (3 channels for unit 0 only; continuous sampling can be set) • Variable sampling state count • Self-diagnosis of 12-bit A/D converter • Selectable A/D-converted value addition mode or average mode • Analog input disconnection detection function (Discharge function/precharge function) • Double trigger mode (duplication of A/D conversion data) • Switching function of 8-, 10-, and 12-bit conversion*2 • Automatic clear function of A/D data registers • Extended analog input function • Digital comparison (window function can be selected)
Interrupt source	• In the modes except double trigger mode and group scan mode, A/D scan end interrupt (S12ADI) request can be generated on completion of single scan. • In double trigger mode, A/D scan end interrupt (S12ADI) request can be generated on completion of double scan. • In group scan mode, an A/D scan end interrupt (S12ADI) request can be generated on completion of group A scan, whereas an A/D scan end interrupt for group B (S12GBADI) request can be generated on completion of group B scan. • When double trigger mode is selected in group scan mode, A/D scan end interrupt (S12ADI) request can be generated on completion of double scan of group A, whereas A/D scan end interrupt specially for group B (S12GBADI) request can be generated on completion of group B scan. • A compare interrupt (S12CMPI) can be generated in response to matches with a condition for comparison by digital comparison. • The S12ADI and S12GBADI interrupts can activate the DMA controller (DMAC) and the data transfer controller (DTC).
Event link function	• An ELC event is generated on completion of scans other than group B scan in group scan mode. • Scan can be started by a trigger output by ELC.
Low power consumption function	• Module stop state can be specified.*3

Note 1. Peripheral module clock PCLKB is set according to the setting of the SCKCR.PCKB[3:0] bits and A/D conversion clock ADCLK is set according to the setting of the SCKCR.PCKC[3:0] bits and the SCKCR.PCKD[3:0] bits in unit 0 and unit 1, respectively.
Note 2. When A/D conversion accuracy is modified, A/D conversion time is also changed. See section 57.3.7, Analog Input Sampling and Scan Conversion Time for details.
Note 3. See section 11, Low Power Consumption for details.

Table 57.2 Functions of 12-Bit A/D Converter (1/2)

Item			Unit 0 (S12AD)	Unit 1 (S12AD1)
Analog input channel			AN000 to AN007	AN100 to AN120 Internal reference voltage Temperature sensor output Extended input
Conditions for A/D conversion start	Software	Software trigger	Enabled	Enabled
	External trigger	Trigger input pin	ADTRG0#	ADTRG1#
	Synchronous trigger (trigger from MTU)	Compare match with or input capture to MTU0.TGRA	TRGA0N	TRGA0N
		Compare match with or input capture to MTU1.TGRA	TRGA1N	TRGA1N
		Compare match with or input capture to MTU2.TGRA	TRGA2N	TRGA2N
		Compare match with or input capture to MTU3.TGRA	TRGA3N	TRGA3N
		Compare match with or input capture to MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	TRGA4N	TRGA4N
		Compare match with or input capture to MTU6.TGRA	TRGA6N	TRGA6N
		Compare match with or input capture to MTU7.TGRA or underflow (trough) of MTU7.TCNT in complementary PWM mode	TRGA7N	TRGA7N
		Compare match with MTU0.TGRE	TRG0N	TRG0N
		Compare match between MTU4.TADCORA and MTU4.TCNT	TRG4AN	TRG4AN
		Compare match between MTU4.TADCORB and MTU4.TCNT	TRG4BN	TRG4BN
		Compare match between MTU4.TADCORA and MTU4.TCNT or compare match between MTU4.TADCORB and MTU4.TCNT	TRG4AN or TRG4BN	TRG4AN or TRG4BN
		Compare match between MTU4.TADCORA and MTU4.TCNT and compare match between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is used)	TRG4ABN	TRG4ABN
		Compare match between MTU7.TADCORA and MTU7.TCNT	TRG7AN	TRG7AN
		Compare match between MTU7.TADCORB and MTU7.TCNT	TRG7BN	TRG7BN
		Compare match between MTU7.TADCORA and MTU7.TCNT or compare match between MTU7.TADCORB and MTU7.TCNT	TRG7AN or TRG7BN	TRG7AN or TRG7BN
		Compare match between MTU7.TADCORA and MTU7.TCNT and compare match between MTU7.TADCORB and MTU7.TCNT (when interrupt skipping function 2 is used)	TRG7ABN	TRG7ABN
	Synchronous trigger (trigger from GPT)	Compare match with GTP0.GTADTRA	GTADTRA0N	GTADTRA0N
		Compare match with GPT0.GTADTRB	GTADTRB0N	GTADTRB0N
		Compare match with GPT1.GTADTRA	GTADTRA1N	GTADTRA1N
		Compare match with GPT1.GTADTRB	GTADTRB1N	GTADTRB1N
		Compare match with GPT2.GTADTRA	GTADTRA2N	GTADTRA2N
		Compare match with GPT2.GTADTRB	GTADTRB2N	GTADTRB2N
		Compare match with GPT3.GTADTRA	GTADTRA3N	GTADTRA3N
		Compare match with GPT3.GTADTRB	GTADTRB3N	GTADTRB3N
		Compare match with GTP0.GTADTRA or compare match with GTP0.GTADTRB	GTADTRA0N or GTADTRB0N	GTADTRA0N or GTADTRB0N
		Compare match with GTP1.GTADTRA or compare match with GTP1.GTADTRB	GTADTRA1N or GTADTRB1N	GTADTRA1N or GTADTRB1N
		Compare match with GTP2.GTADTRA or compare match with GTP2.GTADTRB	GTADTRA2N or GTADTRB2N	GTADTRA2N or GTADTRB2N
		Compare match with GTP3.GTADTRA or compare match with GTP3.GTADTRB	GTADTRA3N or GTADTRB3N	GTADTRA3N or GTADTRB3N
	Synchronous trigger (trigger from TMR)*1	Compare match between TMR0.TCORA and TMR0.TCNT	TMTRG0AN_0	TMTRG0AN_0
		Compare match between TMR2.TCORA and TMR2.TCNT	TMTRG0AN_1	TMTRG0AN_1
	Synchronous trigger (trigger from TPU)*1	Compare match with or input capture to TPU0.TGRA, compare match with or input capture to TPU1.TGRA, compare match with or input capture to TPU2.TGRA, compare match with or input capture to TPU3.TGRA, or compare match with or input capture to TPU4.TGRA	TPTRGAN_0	TPTRGAN_0
		Compare match with or input capture to TPU0.TGRA0	TPTRG0AN_0	TPTRG0AN_0

Table 57.2 Functions of 12-Bit A/D Converter (2/2)

Item			Unit 0 (S12AD)	Unit 1 (S12AD1)
Conditions for A/D conversion start	Synchronous trigger (trigger from ELC)	ELC trigger	ELCTRG0	ELCTRG1
Channel-dedicated sample-and-hold function	Target channel		AN000 to AN002	—
Interrupt			S12ADI S12GBADI S12CMPI	S12ADI1 S12GBADI1 S12CMPI1
Setting of module stop function*2			MSTPCRA.MSTPA17 bit	MSTPCRA.MSTPA16 bit

Note 1. “0” and “1” added to synchronous triggers indicate the number of unit. For the settings to output synchronous triggers, see the sections related to A/D converter startup of the responding modules.

Note 2. See section 11, Low Power Consumption for details.

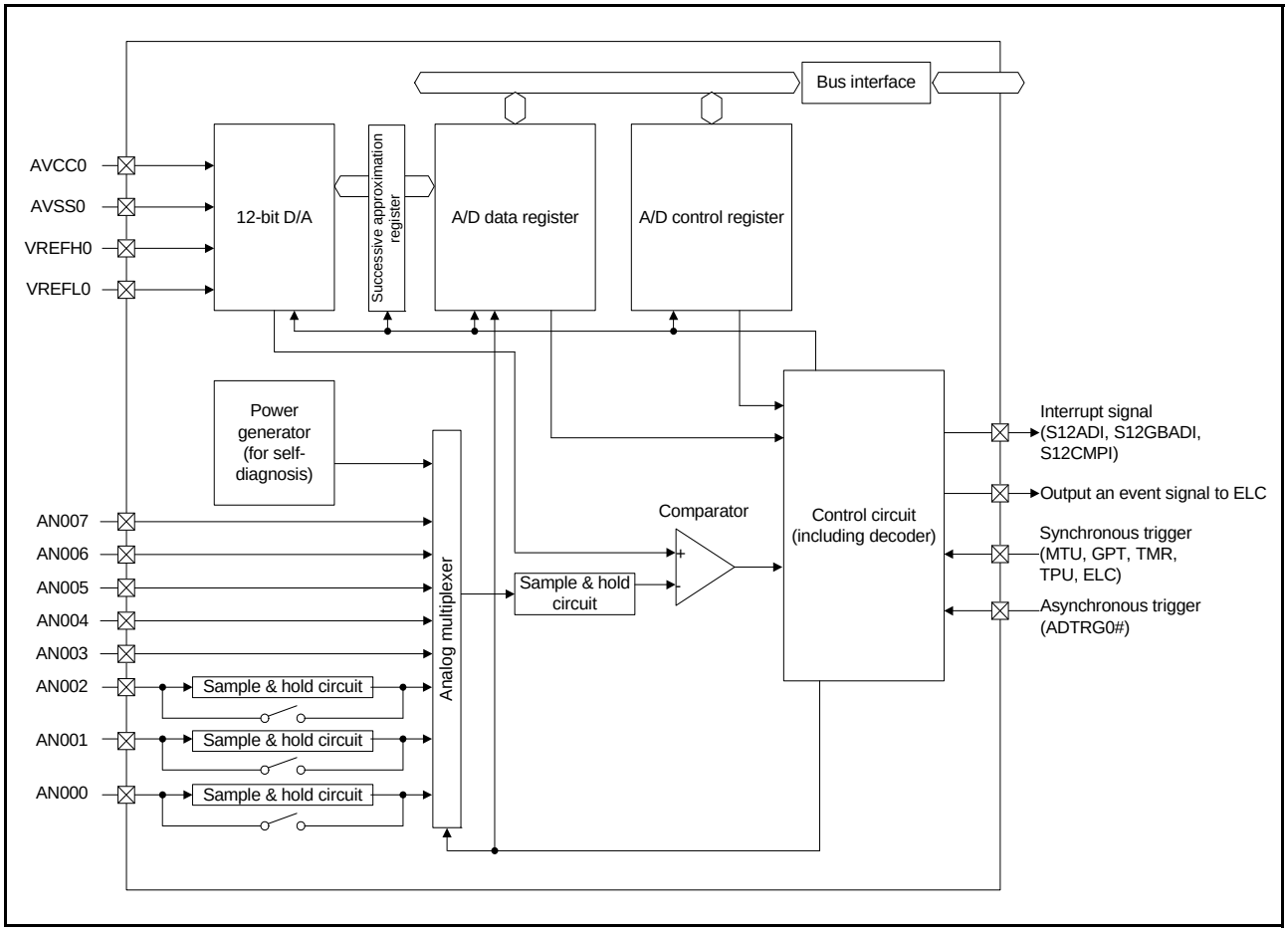


Figure 57.1 Block Diagram of 12-Bit A/D Converter (Unit 0)

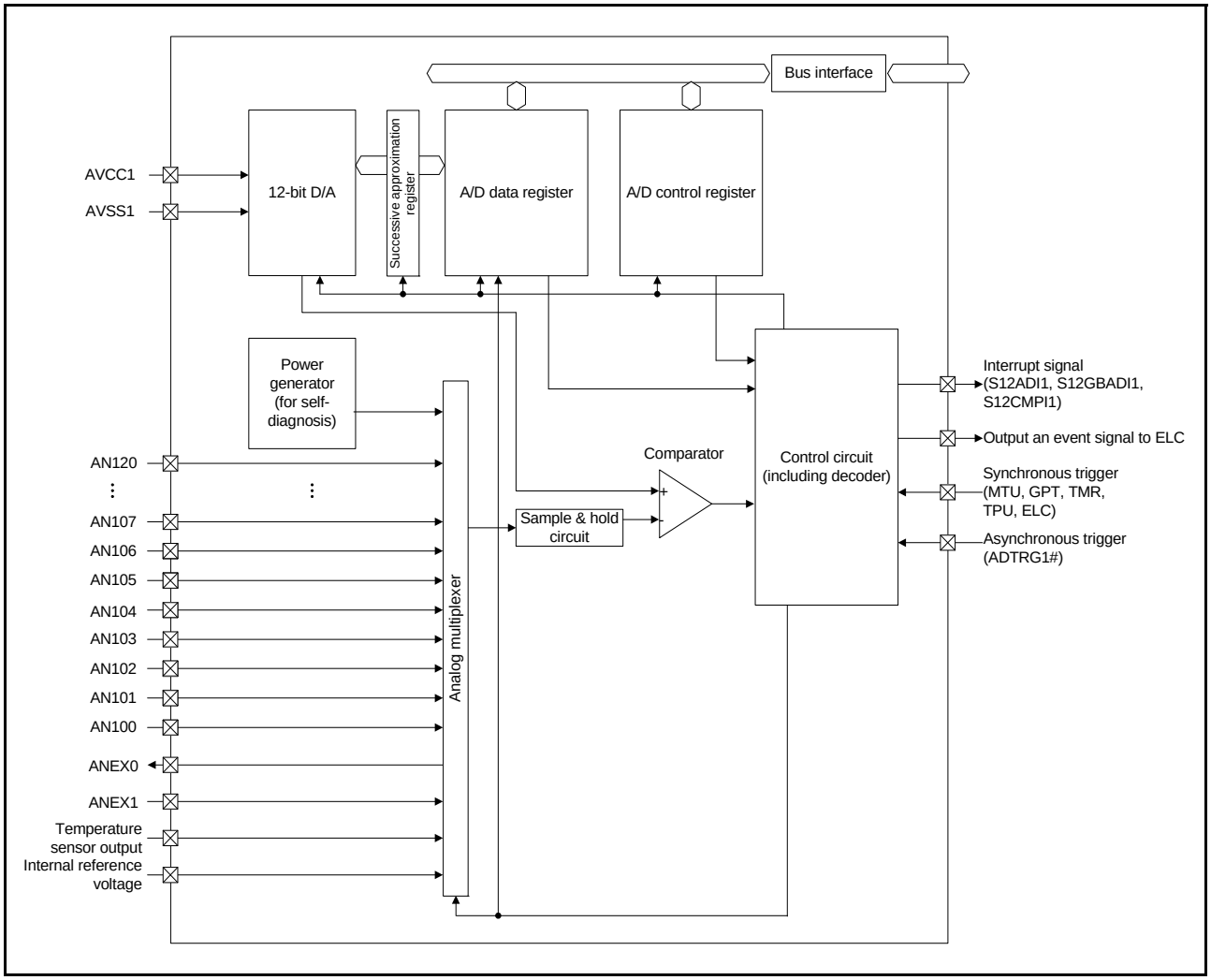


Figure 57.2 Block Diagram of 12-Bit A/D Converter (Unit 1)

Table 57.3 lists the I/O pins of the 12-bit A/D converter.

Table 57.3 I/O Pins of 12-Bit A/D Converter

Unit	Pin Name	I/O	Function
Unit 0	AVCC0	Input	Analog block power supply pin
	AVSS0	Input	Analog block ground pin
	VREFH0	Input	Reference power supply pin
	VREFL0	Input	Reference power supply ground pin
	AN000 to AN007	Input	Analog input pin 0 to 7
	ADTRG0#	Input	External trigger input pin for starting A/D conversion
Unit 1	AVCC1	Input	Multiplexed analog block power supply and reference power supply pin functions
	AVSS1	Input	Multiplexed analog block ground and reference power supply ground pin functions
	AN100 to AN120	Input	Analog input pin 8 to 28
	ANEX0	Output	Extended analog output pin
	ANEX1	Input	Extended analog input pin
	ADTRG1#	Input	External trigger input pin for starting A/D conversion

57.2 Register Descriptions

57.2.1 A/D Data Registers y (ADDRy), A/D Data Duplication Register (ADDBLDR), A/D Data Duplication Register A (ADDBLDRA), A/D Data Duplication Register B (ADDBLDRB), A/D Temperature Sensor Data Register (ADTSDR), A/D Internal Reference Voltage Data Register (ADOCDR)

ADDRy registers (y = 0 to 7 in unit 0; y = 0 to 20 in unit 1) are 16-bit read-only registers for storing the result of A/D conversion. The ADDBLDR register is a 16-bit read-only register for storing the result of A/D conversion in response to the second trigger in double trigger mode. The ADDBLDRA and ADDBLDRB are 16-bit read-only registers for storing the result of A/D conversion in response to the respective triggers during extended operation in double trigger mode. Register ADTSDR is a 16-bit read-only register for storing the A/D conversion result of temperature sensor output. Register ADOCDR is a 16-bit read-only register for storing the A/D result of internal reference voltage.
The formats for data in the ADDRy, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, and ADOCDR registers vary according to the following conditions.

- The setting of the A/D data register format select bit (ADCER.ADRFT) (determining whether the data are flush-left or flush-right in the registers)
- The setting of the A/D data register bit-accuracy specification bits (ADCER.ADPRC[1:0]) (8-, 10-, or 12-bit is selectable.)
- The setting of the addition frequency select bits (ADADC.ADC[1:0]) (once, twice, or three times is selectable.)
- The setting of the average mode enable bit (ADADC.AVEE) (Addition or average is selectable.)

The data formats for each given condition are shown below.

(1) When A/D-converted value addition/average mode is not selected

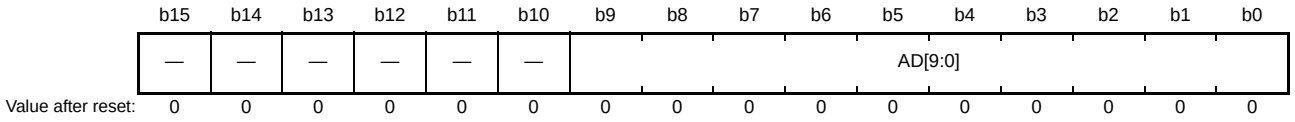
- The settings for flush-right data with 12-bit accuracy

Address(es): S12AD.ADDR0 0008 9020h to S12AD.ADDR7 0008 902Eh,
S12AD.ADDBLDR 0008 9018h, S12AD.ADDBLDRA 0008 9084h, S12AD.ADDBLDRB 0008 9086h,
S12AD1.ADDR0 0008 9120h to S12AD1.ADDR20 0008 9148h,
S12AD1.ADDBLDR 0008 9118h, S12AD1.ADDBLDRA 0008 9184h, S12AD1.ADDBLDRB 0008 9186h,
S12AD1.ADTSDR 0008 911Ah, S12AD1.ADOCDR 0008 911Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	AD[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

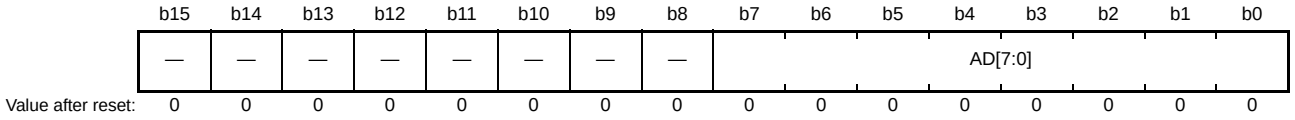
Bit	Symbol	Bit Name	Description	R/W
b11 to b0	AD[11:0]	Converted Value 11 to 0	12-bit A/D-converted value	R
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- The settings for flush-right data with 10-bit accuracy



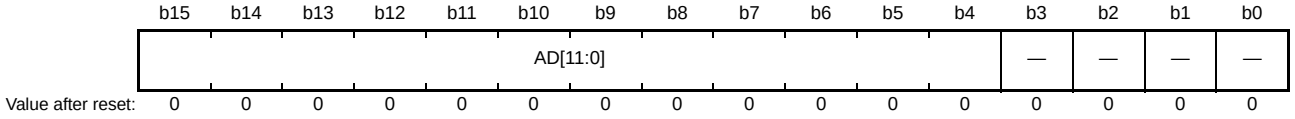
Bit	Symbol	Bit Name	Description	R/W
b9 to b0	AD[9:0]	Converted Value 9 to 0	10-bit A/D-converted value	R
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- The settings for flush-right data with 8-bit accuracy



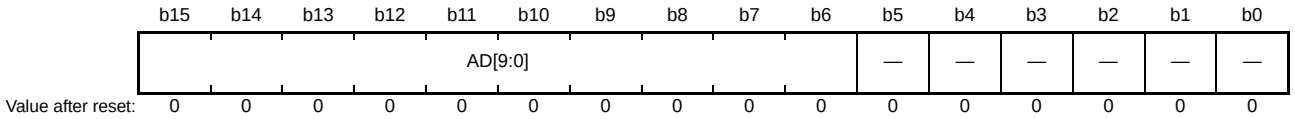
Bit	Symbol	Bit Name	Description	R/W
b7 to b0	AD[7:0]	Converted Value 7 to 0	8-bit A/D-converted value	R
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- The settings for flush-left data with 12-bit accuracy



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b4	AD[11:0]	Converted Value 11 to 0	12-bit A/D-converted value	R

- The settings for flush-left data with 10-bit accuracy



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b6	AD[9:0]	Converted Value 9 to 0	10-bit A/D-converted value	R

- The settings for flush-left data with 8-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	AD[7:0]	Converted Value 7 to 0	8-bit A/D-converted value	R

(2) When A/D-converted value average mode is selected

When A/D-converted value average mode is selected, the AD[11:0] bits indicate the mean of A/D-converted values on a specific channel. Even if A/D-converted value average mode is selected, the value is stored in the A/D data register according to the settings of the A/D data register format select bit in the same way as normal A/D conversion.

(3) When A/D-converted value addition mode is selected

When A/D-converted value addition mode is selected, the AD[13:0] bits indicate the value that is obtained by adding up A/D-converted values on a specific channel. In A/D-converted value addition mode, the value obtained by adding up of A/D conversion results is retained in the A/D data register as a 2-bit-extended value of the conversion accuracy specified. Even if A/D-converted value addition mode is selected, the value is stored in the A/D data register according to the settings of the A/D data register format select bits.
The data formats for each given condition are shown below.

- The settings for flush-right data with 12-bit accuracy
  (when A/D-converted value addition mode is selected)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AD[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

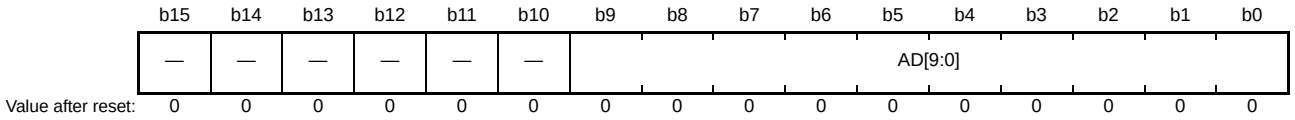
Bit	Symbol	Bit Name	Description	R/W
b13 to b0	AD[13:0]	Added Value 13 to 0	14-bit value obtained by adding up of A/D conversion results	R
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- The settings for flush-right data with 10-bit accuracy
  (when A/D-converted value addition mode is selected)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	AD[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	AD[11:0]	Added Value 11 to 0	12-bit value obtained by adding up of A/D conversion results	R
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- The settings for flush-right data with 8-bit accuracy
  (when A/D-converted value addition mode is selected)

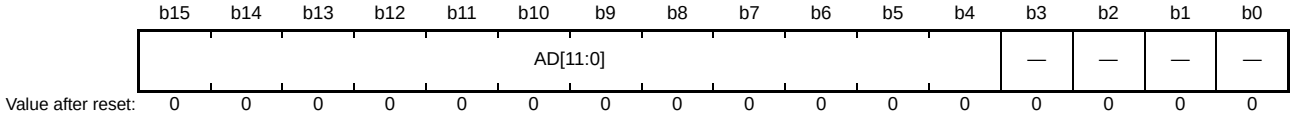


Bit	Symbol	Bit Name	Description	R/W
b9 to b0	AD[9:0]	Added Value 9 to 0	10-bit value obtained by adding up of A/D conversion results	R
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- The settings for flush-left data with 12-bit accuracy
  (when A/D-converted value addition mode is selected)

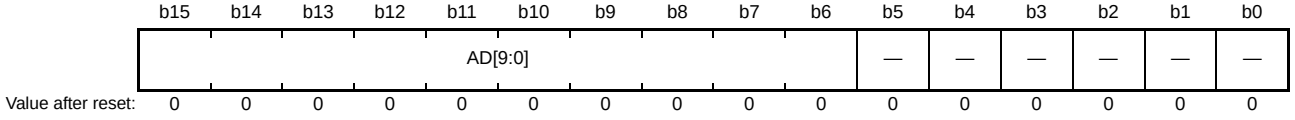
Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b2	AD[13:0]	Added Value 13 to 0	14-bit value obtained by adding up of A/D conversion results	R

- The settings for flush-left data with 10-bit accuracy
  (when A/D-converted value addition mode is selected)



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b4	AD[11:0]	Added Value 11 to 0	12-bit value obtained by adding up of A/D conversion results	R

- The settings for flush-left data with 8-bit accuracy
  (when A/D-converted value addition mode is selected)



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b6	AD[9:0]	Added Value 9 to 0	10-bit value obtained by adding up of A/D conversion results	R

57.2.2 A/D Self-Diagnosis Data Register (ADRD)

ADRD is a 16-bit read-only register that holds the A/D conversion results based on the 12-bit A/D converter's self-diagnosis. In addition to the AD bit indicating A/D-converted value, the self-diagnosis status bit (DIAGST[1:0]) is included in. In the ADRD register, the following different formats are used depending on the conditions below.

- The setting of the A/D data register format select bit (ADCER.ADRFT) (determining whether the data are flush-left or flush-right in the registers)
- The setting of the A/D data register bit-accuracy specify bits (ADCER.ADPRC[1:0]) (8-, 10-, or 12-bit is selectable.)

The A/D-converted value addition mode and A/D-converted value average mode cannot be applied to the A/D self-diagnosis function. For details of self-diagnosis, see section 57.2.11, A/D Control Extended Register (ADCER). The data formats for each given condition are shown below.

- The settings for flush-right data with 12-bit accuracy

Address(es): S12AD.ADRD 0008 901Eh, S12AD1.ADRD 0008 911Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DIAGST[1:0]		—	—	AD[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	AD[11:0]	Converted Value 11 to 0	12-bit A/D-converted value	R
b13, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	DIAGST[1:0]	Self-Diagnosis Status	b15 b14 0 0: Self-diagnosis has never been executed since power-on. 0 1: Self-diagnosis using the voltage of 0 V has been executed. 1 0: Self-diagnosis using the voltage of reference power supply*1 × 1/2 has been executed. 1 1: Self-diagnosis using the voltage of reference power supply*1 has been executed. For details of self-diagnosis, see section 57.2.11, A/D Control Extended Register (ADCER).	R

Note 1. "Reference voltage" refers to VREFH0 for unit 0 and to AVCC1 for unit 1.

- The settings for flush-right data with 10-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DIAGST[1:0]		—	—	—	—	AD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	AD[9:0]	Converted Value 9 to 0	10-bit A/D-converted value	R
b13 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	DIAGST[1:0]	Self-Diagnosis Status	b15 b14 0 0: Self-diagnosis has never been executed since power-on. 0 1: Self-diagnosis using the voltage of 0 V has been executed. 1 0: Self-diagnosis using the voltage of reference power supply*1 × 1/2 has been executed. 1 1: Self-diagnosis using the voltage of reference power supply*1 has been executed. For details of self-diagnosis, see section 57.2.11, A/D Control Extended Register (ADCER).	R

Note 1. “Reference voltage” refers to VREFH0 for unit 0 and to AVCC1 for unit 1.

- The settings for flush-right data with 8-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DIAGST[1:0]		—	—	—	—	—	—	AD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	AD[7:0]	Converted Value 7 to 0	8-bit A/D-converted value	R
b13 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	DIAGST[1:0]	Self-Diagnosis Status	b15 b14 0 0: Self-diagnosis has never been executed since power-on. 0 1: Self-diagnosis using the voltage of 0 V has been executed. 1 0: Self-diagnosis using the voltage of reference power supply*1 × 1/2 has been executed. 1 1: Self-diagnosis using the voltage of reference power supply*1 has been executed. For details of self-diagnosis, see section 57.2.11, A/D Control Extended Register (ADCER).	R

Note 1. “Reference voltage” refers to VREFH0 for unit 0 and to AVCC1 for unit 1.

- The settings for flush-left data with 12-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[11:0]												—	—	DIAGST[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DIAGST[1:0]	Self-Diagnosis Status	b1 b0 0 0: Self-diagnosis has never been executed since power-on. 0 1: Self-diagnosis using the voltage of 0 V has been executed. 1 0: Self-diagnosis using the voltage of reference power supply*1 × 1/2 has been executed. 1 1: Self-diagnosis using the voltage of reference power supply*1 has been executed. For details of self-diagnosis, see section 57.2.11, A/D Control Extended Register (ADCER).	R
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b4	AD[11:0]	Converted Value 11 to 0	12-bit A/D-converted value	R

Note 1. “Reference voltage” refers to VREFH0 for unit 0 and to AVCC1 for unit 1.

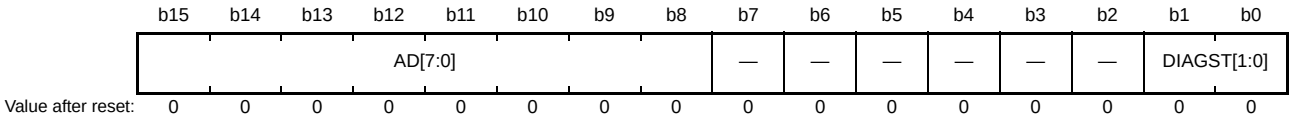
- The settings for flush-left data with 10-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[9:0]										—	—	—	—	DIAGST[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DIAGST[1:0]	Self-Diagnosis Status	b1 b0 0 0: Self-diagnosis has never been executed since power-on. 0 1: Self-diagnosis using the voltage of 0 V has been executed. 1 0: Self-diagnosis using the voltage of reference power supply*1 × 1/2 has been executed. 1 1: Self-diagnosis using the voltage of reference power supply*1 has been executed. For details of self-diagnosis, see section 57.2.11, A/D Control Extended Register (ADCER).	R
b5 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b6	AD[9:0]	Converted Value 9 to 0	10-bit A/D-converted value	R

Note 1. “Reference voltage” refers to VREFH0 for unit 0 and to AVCC1 for unit 1.

- The settings for flush-left data with 8-bit accuracy



Bit	Symbol	Bit Name	Description	R/W
b1, b0	DIAGST[1:0]	Self-Diagnosis Status	b1 b0 0 0: Self-diagnosis has never been executed since power-on. 0 1: Self-diagnosis using the voltage of 0 V has been executed. 1 0: Self-diagnosis using the voltage of reference power supply*1 × 1/2 has been executed. 1 1: Self-diagnosis using the voltage of reference power supply*1 has been executed. For details of self-diagnosis, see section 57.2.11, A/D Control Extended Register (ADCER).	R
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	AD[7:0]	Converted Value 7 to 0	8-bit A/D-converted value	R

Note 1. “Reference voltage” refers to VREFH0 for unit 0 and to AVCC1 for unit 1.

57.2.3 A/D Control Register (ADCSR)

Address(es): S12AD.ADCSR 0008 9000h, S12AD1.ADCSR 0008 9100h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADST	ADCS[1:0]		ADIE	—	—	TRGE	EXTRG	DBLE	GBADIE	—	DBLANS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DBLANS[4:0]	Double Trigger Channel Select	These bits select one analog input channel for double triggered operation. The setting is only effective while double trigger mode is selected.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	GBADIE	Group B Scan End Interrupt Enable	0: Disables S12GBADI interrupt generation upon group B scan completion. 1: Enables S12GBADI interrupt generation upon group B scan completion.	R/W
b7	DBLE	Double Trigger Mode Select	0: Deselects double trigger mode. 1: Selects double trigger mode.	R/W
b8	EXTRG	Trigger Select*1	0: A/D conversion is started by a synchronous trigger (MTU, GPT, TPU, TMR, ELC). 1: A/D conversion is started by the asynchronous trigger (ADTRG0# in unit 0; ADTRG1# in unit 1).	R/W
b9	TRGE	Trigger Start Enable	0: Disables A/D conversion to be started by the synchronous or asynchronous trigger. 1: Enables A/D conversion to be started by the synchronous or asynchronous trigger.	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	ADIE	Scan End Interrupt Enable	0: Disables S12ADI interrupt generation upon scan completion. 1: Enables S12ADI interrupt generation upon scan completion.	R/W
b14, b13	ADCS[1:0]	Scan Mode Select	b14 b13 0 0: Single scan mode 0 1: Group scan mode 1 0: Continuous scan mode 1 1: Setting prohibited	R/W
b15	ADST	A/D Conversion Start	0: Stops A/D conversion process. 1: Starts A/D conversion process.	R/W

Note 1. Starting A/D conversion using an external pin (asynchronous trigger)
After a high-level signal is input to the external pin (ADTRG0# in unit 0; ADTRG1# in unit 1), write 1 to both the TRGE and EXTRG bits in ADCSR register and change the signals of ADTRG0# in unit 0 and ADTRG1# in unit 1 to Low. Thus the falling edge of ADTRG0# in unit 0 and ADTRG1# in unit 1 are detected and the scan conversion process is started. In this case, the pulse width of the low-level input must be at least 1.5 clock cycles of PCLKB.

ADCSR sets double trigger mode, A/D conversion start trigger; enables/disables scan end interrupt; selects the scan mode; and starts or stops A/D conversion.

DBLANS[4:0] Bits (Double Trigger Channel Select)

The DBLANS[4:0] bits select one of the channels for A/D conversion data duplication in double trigger mode. The A/D conversion results of the analog input of the channel selected by the DBLANS[4:0] bits are stored into the A/D data register y when conversion is started by the first trigger, and into the A/D data duplication register when started by the second trigger. Table 57.4 shows selection of the channel for double triggered operation.
When double trigger mode is selected, the channels selected by ADANSA0 and ADANSA1 registers are invalid, and the channel selected by the DBLANS[4:0] bits is subjected to A/D conversion instead.
When double trigger mode is used, do not select A/D conversion for the self-diagnosis function, temperature sensor output, and internal reference voltage (temperature sensor output and internal reference voltage can be selected for A/D

conversion for group B in group scan mode). The DBLANS[4:0] bits should be set while the ADCSR.ADST bit is 0. (They should not be set simultaneously when 1 is written to the ADCSR.ADST bit.)
To enter A/D-converted value addition/average mode while double trigger mode is set, the channel selected by the DBLANS[4:0] bits should be selected in the ADANSA0 and ADANSA1 registers.

Table 57.4 Relationship between DBLANS[4:0] Bits Settings and Double Trigger Enabled Channels

Unit 0		Unit 1					
DBLANS[4:0]	Duplication channel	DBLANS[4:0]	Duplication channel	DBLANS[4:0]	Duplication channel	DBLANS[4:0]	Duplication channel
00000	AN000	00000	AN100	01000	AN108	10000	AN116
00001	AN001	00001	AN101	01001	AN109	10001	AN117
00010	AN002	00010	AN102	01010	AN110	10010	AN118
00011	AN003	00011	AN103	01011	AN111	10011	AN119
00100	AN004	00100	AN104	01100	AN112	10100	AN120
00101	AN005	00101	AN105	01101	AN113		
00110	AN006	00110	AN106	01110	AN114		
00111	AN007	00111	AN107	01111	AN115		

Note: A/D-converted data of the self-diagnosis function, temperature sensor output, and internal reference voltage cannot be used in double trigger mode.

GBADIE Bit (Group B Scan End Interrupt Enable)

The GBADIE bit enables or disables group B scan end interrupt (S12GBADI) in group scan mode.

DBLE Bit (Double Trigger Mode Select)

Double trigger mode has a function to store the resulting data of A/D conversion started by the first and second synchronous triggers into separate registers.
When double trigger mode is selected, the channels specified in the ADANS0 and ADANS1 registers are invalid and the channel selected by the DBLANS[4:0] bits is effective instead. Double trigger mode can be only operated by the synchronous trigger (MTU, ELC, GPT, TMR, TPU) selected by the ADSTRGR.TRSA[5:0] bits; it cannot be operated by the asynchronous or software trigger. The A/D conversion results started by the first trigger are stored into the A/D data register y and those started by the second trigger are stored into the A/D data duplication register. In this case, if the ADIE bit is set to 1, the interrupt is output not upon completion of the first conversion but upon completion of the second conversion.
In continuous scan mode, double trigger mode should not be selected. Also do not select double trigger mode for conversion of the self-diagnosis function, temperature sensor output, and internal reference voltage.
The DBLE bit should be set after the ADCSR.ADST bit has been set to 0.

EXTRG Bit (Trigger Select)

The EXTRG bit selects the synchronous trigger or the asynchronous trigger as the trigger for starting A/D conversion.

TRGE Bit (Trigger Start Enable)

The TRGE bit enables or disables A/D conversion by the synchronous trigger and the asynchronous trigger.
This bit should be set to 1 in group scan mode.

ADIE Bit (Scan End Interrupt Enable)

The ADIE bit enables or disables the A/D scan end interrupt (S12ADI) in scans except for group B scan in group scan mode.
With double trigger mode deselected, the S12ADI interrupt is generated after the first scan is completed if the ADIE bit is set to 1.

With the extended analog input selected, the S12ADI interrupt is generated after A/D conversion is completed if the ADIE bit is set to 1.
With double trigger mode selected, the S12ADI interrupt is generated after the second scan is completed if the ADIE bit is set to 1 as long as the scan is started by the synchronous trigger (MTU, ELC, GPT, TMR, TPU) selected by the ADSTRGR.TRSA[5:0] bits.
When a scan is started by a software trigger, even if double trigger mode is selected, the S12ADI interrupt is generated after the scan is completed as long as the ADIE bit is set to 1.

ADCS[1:0] Bits (Scan Mode Select)

The ADCS bit select the scan mode.
In single scan mode, A/D conversion is performed for the analog inputs of a maximum of eight channels in unit 0 and 21 channels in unit 1 selected with the ADANSA0 and ADANSA1 registers in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, the scan conversion is stopped.*1
In continuous scan mode, while the ADST bit in ADCSR register is 1, A/D conversion is performed for the analog inputs of a maximum of eight channels in unit 0 and 21 channels in unit 1 selected with the ADANSA0 and ADANSA1 registers in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is repeated from the first channel. If the ADST bit in ADCSR register is set to 0 during continuous scan, A/D conversion is stopped even if scanning is in progress.*1
In group scan mode, A/D conversion is performed for the analog inputs (group A) of a maximum of eight channels in unit 0 and 21 channels in unit 1 selected with the ADANSA0 and ADANSA1 registers in the ascending order of the channel number after scanning is started by the synchronous trigger (MTU, ELC, GPT, TMR, TPU) selected by the TRSA[5:0] bits in ADSTRGR register, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped.*1 A/D conversion is also performed for the analog inputs (group B) of a maximum of eight channels in unit 0 and 21 channels in unit 1 selected with the ADANSB0 and ADANSB1 registers in the ascending order of the channel number after scanning is started by the synchronous trigger (MTU, ELC, GPT, TMR, TPU) selected by the TRSB[5:0] bits in ADSTRGR register, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped.*1 If the conversion processes in group A and B occur at the same time, those conversion cannot be controlled separately. In this case, set the group A priority control setting bit (ADGSPCR.PGS) in the A/D group scan priority control register (ADGSPCR) to 1 in order to assign a priority to conversion of group A.*1
In group scan mode, different channels and triggers should be selected for group A and group B.
When the extended analog input is selected, single scan mode or continuous scan mode should be selected.
The ADCS[1:0] bits should be set while the ADCSR.ADST bit is 0 (it should not be set simultaneously when 1 is written to the ADCSR.ADST bit.)

Note 1. When the temperature sensor output or internal reference voltage is selected, A/D conversion of the designated analog input channels is followed by A/D conversion of the temperature sensor output and the internal reference voltage in that order.

Table 57.5 shows the selectable targets for A/D conversion depending on the settings of scan mode and double trigger mode.

Table 57.5 Selectable Targets for A/D Conversion Depending on the Settings of Scan Mode and Double Trigger Mode

Scan mode setting	Double Trigger Mode Setting	Targets for A/D conversion					
		Self-diagnosis	Analog input (including group A)	Analog input (group B)	Temperature sensor output	Internal reference voltage	Extended analog input
Single scan	DBLE = 0	○	○	×	○	○	○
	DBLE = 1	○	○ (1ch only)	×	×	×	×
Continuous scan	DBLE = 0	○	○	×	○	○	○
	DBLE = 1	×	×	×	×	×	×
Group scan	DBLE = 0	○	○	○	○	○	×
	DBLE = 1	○	○ (1ch only)	○	○ (group B only)	○ (group B only)	×

Note: ○: Selectable; ×: Not selectable
Note: When the extended analog input is selected as a target for A/D conversion, other A/D conversion targets should not be selected.

ADST Bit (A/D Conversion Start)

The ADST bit starts or stops A/D conversion process.
Before the ADST bit is set to 1, set the A/D conversion clock, the conversion mode, and conversion target analog input.
[Setting conditions]

- 1 is written by software.
- The synchronous trigger (MTU, ELC, GPT, TMR, TPU) selected by the ADSTRGR.TRSA[5:0] bits is detected with ADCSR.EXTRG and ADCSR.TRGE bits being set to 0 and 1, respectively.
- The synchronous trigger (MTU, ELC, GPT, TMR, TPU) selected by the ADSTRGR.TRSB[5:0] bits is detected with the ADCSR.TRGE bit being set to 1 in group scan mode.
- The asynchronous trigger is detected with the ADCSR.TRGE and ADCSR.EXTRG bits being set to 1 and the ADSTRGR.TRSA[5:0] bits being set to 000000b.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), a group B trigger is detected and A/D conversion of group B is started.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRSCN bit is set to 1 and A/D conversion of group B is restarted.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRP bit is set to 1 and A/D conversion of group B is started.

[Clearing conditions]

- 0 is written by software.
- The A/D conversion of all the selected channels, the temperature sensor output or the internal reference voltage is completed in single scan mode.
- The A/D conversion of the extended analog input is completed in single scan mode.
- Group A scan is completed in group scan mode.
- Group B scan is completed in group scan mode.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), a group A trigger is detected during group B A/D conversion and the scanning of group B is stopped.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRP bit is set to 1 and the scanning of group B started by a resumption trigger is completed.
- With group-A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1) the ADGSPCR.GBRSCN bit is set to 1 and the scanning of group B by a trigger is completed.

Note: When group-A priority control operation mode has been enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), do not set the ADST bit to 1.
Note: When group-A priority control operation mode has been enabled (ADCSR.ADCS[1:0] bits = 01b and

ADGSPCR.PGS bit = 1) and ADGSPCR.GBRP = 1, do not set the ADST bit to 0. When forcibly terminating A/D conversion, follow the procedure for clearing the ADST bit.

57.2.4 A/D Channel Select Register A0 (ADANSA0)

Address(es): S12AD.ADANSA0 0008 9004h, S12AD1.ADANSA0 0008 9104h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ANSA0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	ANSA0[15:0]	A/D Conversion Channels Select	0: AN000 to AN007 (unit 0) and AN100 to AN115 (unit 1) are not subjected to conversion. 1: AN000 to AN007 (unit 0) and AN100 to AN115 (unit 1) are subjected to conversion.	R/W

ADANSA0 selects analog input channels for A/D conversion among AN000 to AN007 (unit 0) and AN100 to AN115 (unit 1). In group scan mode, this register selects group A channels.

ANSA0[15:0] Bits (A/D Conversion Channels Select)

The ANSA0[15:0] bits select analog input channels for A/D conversion among AN000 to AN007 (unit 0) and AN100 to AN115 (unit 1). The channels to be selected and the number of channels can be arbitrarily set. In unit 0, the ANSA0[0] bit corresponds to AN000 and the ANSA0[7] bit corresponds to AN007. In unit 1, the ANSA0[0] bit corresponds to AN100 and the ANSA0[15] bit corresponds to AN115.

When double trigger mode is selected, the channel selected by the ANSA0[15:0] bits is invalid, and the channel selected by the ADCSR.DBLANS[4:0] bits is selected in group A instead.

When group scan mode is selected, do not select the channels specified in the A/D channel select register B0 (ADANSB0) and the A/D channel select register B1 (ADANSB1).

The ANSA0[15:0] bits should be set while the ADCSR.ADST bit is 0.

57.2.5 A/D Channel Select Register A1 (ADANSA1)

Address(es): S12AD1.ADANSA1 0008 9106h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	ANSA1[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	ANSA1[4:0]	A/D Conversion Channels Select	0: AN116 to AN120 (unit 1) are not subjected to conversion. 1: AN116 to AN120 (unit 1) are subjected to conversion.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADANSA1 selects analog input channels for A/D conversion among AN116 to AN120 (unit 1). In group scan mode, group A channels are to be selected. There is no register dedicated to unit 0.

ANSA1[4:0] Bits (A/D Conversion Channels Select)

The ANSA1[4:0] bits select analog input channels for A/D conversion among AN116 to AN120 (unit 1). The channels to be selected and the number of channels can be arbitrarily set. The ANSA1[0] bit corresponds to AN116 and the ANSA1[4] bit corresponds to AN120.

When double trigger mode is selected, the channel selected by the ANSA1[4:0] bits is invalid, and the channel selected by the ADCSR.DBLANS[4:0] bits is selected in group A instead.

When group scan mode is selected, do not select the channels specified in the A/D channel select register B0 (ADANSB0) and the A/D channel select register B1 (ADANSB1).

The ANSA1[4:0] bits should be set while the ADCSR.ADST bit is 0.

57.2.6 A/D Channel Select Register B0 (ADANSB0)

Address(es): S12AD.ADANSB0 0008 9014h, S12AD1.ADANSB0 0008 9114h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ANSB0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	ANSB0[15:0]	A/D Conversion Channels Select	0: AN000 to AN007 (unit 0) and AN100 to AN115 (unit 1) are not subjected to conversion. 1: AN000 to AN007 (unit 0) and AN100 to AN115 (unit 1) are subjected to conversion.	R/W

ADANSB0 selects analog input channels for A/D conversion among AN000 to AN007 (unit 0) and AN100 to AN115 (unit 1) in group B when group scan mode is selected. The ADANSB0 register is not used in any scan mode other than group scan mode.

ANSB0[15:0] Bits (A/D Conversion Channels Select)

The ANSB0[15:0] bits select analog input channels for A/D conversion among AN000 to AN007 (unit 0) and AN100 to AN115 (unit 1) in group B when group scan mode is selected. The ADANSB0 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the channels corresponding to group A, selected with the ADANSA0 and ADANSA1 registers and the ADCSR.DBLANS[4:0] bits in double trigger mode) should be excluded as the channels to be selected and the number of channels to be set.

In unit 0, the ANSB0[0] bit corresponds to AN000 and the ANSB0[7] bit corresponds to AN007. In unit 1, the ANSB0[0] bit corresponds to AN100 and the ANSB0[15] bit corresponds to AN115.

The ANSB0[15:0] bits should be set while the ADCSR.ADST bit is 0.

57.2.7 A/D Channel Select Register B1 (ADANSB1)

Address(es): S12AD1.ADANSB1 0008 9116h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	ANSB1[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	ANSB1[4:0]	A/D Conversion Channels Select	0: AN116 to AN120 (unit 1) are not subjected to conversion. 1: AN116 to AN120 (unit 1) are subjected to conversion.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADANSB1 selects analog input channels for A/D conversion among AN116 to AN120 (unit 1) in group B when group scan mode is selected. The ADANSB1 register is not used in any scan mode other than group scan mode. There is no register dedicated to unit 0.

ANSB1[4:0] Bits (A/D Conversion Channels Select)

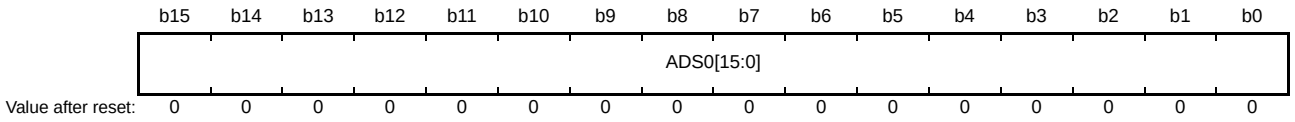
The ANSB1[4:0] bits select analog input channels for A/D conversion among AN116 to AN120 (unit 1) in group B when group scan mode is selected. The ADANSB1 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the channels corresponding to group A, selected with the ADANSA0 and ADANSA1 registers and the ADCSR.DBLANS[4:0] bits in double trigger mode) should be excluded as the channels to be selected and the number of channels to be set.

The ANSB1[0] bit corresponds to AN116 and the ANSB1[4] bit corresponds to AN120.

The ANSB1[4:0] bits should be set while the ADCSR.ADST bit is 0.

57.2.8 A/D-Converted Value Addition/Average Mode Select Register 0 (ADADS0)

Address(es): S12AD.ADADS0 0008 9008h, S12AD1.ADADS0 0008 9108h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	ADS0[15:0]	A/D-Converted Value Addition/ Average Channel Select	0: A/D-converted value addition/average mode for AN000 to AN007 (unit 0) and AN100 to AN115 (unit 1) is not selected. 1: A/D-converted value addition/average mode for AN000 to AN007 (unit 0) and AN100 to AN115 (unit 1) is selected.	R/W

ADADS0 selects the channels 0 to 15 on which A/D conversion is performed successively 2 to 4 times and then converted values are added (integrated) or averaged.

ADS0[15:0] Bits (A/D-Converted Value Addition/Average Channel Select)

When the ADS0[n] bit of the number that is the same as that of A/D-converted channel selected by the ANSA0[n] bits (n = 0 to 15) in ADANSA0 register or DBLANS[4:0] bits in ADCSR register and ANSB0[n] bits (n = 0 to 15) in ADANSB0 register is set to 1, A/D conversion of analog input of the selected channels is performed successively 2 to 4 times that is set with the ADC[1:0] bits in ADADC register. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register. As for the channel on which the A/D conversion is performed and addition/average mode is not selected, a normal one-time conversion is executed and the conversion result is stored to the A/D data register.

The ADS0[15:0] bits should be set while the ADCSR.ADST bit is 0.

57.2.9 A/D-Converted Value Addition/Average Mode Select Register 1 (ADADS1)

Address(es): S12AD1.ADADS1 0008 910Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	ADS1[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	ADS1[4:0]	A/D-Converted Value Addition/ Average Channel Select	0: A/D-converted value addition/average mode for AN116 to AN120 (unit 1) is not selected. 1: A/D-converted value addition/average mode for AN116 to AN120 (unit 1) is selected.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADADS1 selects the channels 16 to 20 on which A/D conversion is performed successively 2 to 4 times and then converted values are added (integrated) or averaged.

ADS1[4:0] Bits (A/D-Converted Value Addition/Average Channel Select)

When the ADS1[n] bit of the number that is the same as that of A/D-converted channel selected by the ANSA1[n] bits (n = 16 to 20) in ADANSA1 register or DBLANS[4:0] bits in ADCSR register and ANSB1[n] bits (n = 16 to 20) in ADANSB1 register is set to 1, A/D conversion of analog input of the selected channels is performed successively 2 to 4 times that is set with the ADC[1:0] bits in ADADC register. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register. As for the channel on which the A/D conversion is performed and addition/average mode is not selected, a normal one-time conversion is executed and the conversion result is stored to the A/D data register.
The ADS1[4:0] bits should be set while the ADCSR.ADST bit is 0.

Figure 57.3 shows a scanning operation sequence in which both the ADS[2] and ADS[6] bits are set to 1.
It is assumed that addition mode is selected (ADADS.AVEE = 0), the addition count is set to 4 (ADADC.ADC[1:0] = 11b), and the channels AN000 to AN007 are selected (ADANSA0.ANSA0[15:0] = 00FFh) in continuous scan mode (ADCSR.ADCS[1:0] = 10b). The conversion process begins with AN000. The AN002 conversion is performed successively 4 times, and the added (integrated) value is returned to the A/ D data register 2. After that the AN003 conversion process is started. The AN006 conversion is performed successively 4 times and the added (integrated) value is returned to the A/D data register 6. After conversion of AN007, the conversion operation is once again performed in the same sequence from AN000.

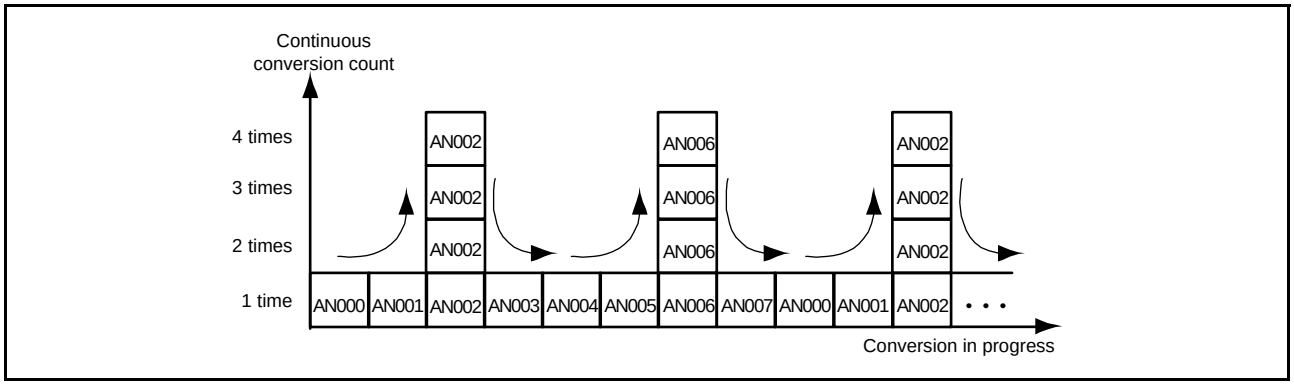


Figure 57.3 Scan Conversion Sequence with ADADC.ADC[1:0] = 11b, ADS[2] = 1, and ADS[6] = 1

57.2.10 A/D-Converted Value Addition/Average Count Select Register (ADADC)

Address(es): S12AD.ADADC 0008 900Ch, S12AD1.ADADC 0008 910Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	AVEE	—	—	—	—	—	ADC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	ADC[1:0]	Addition Count Select	b1 b0 0 0: 1-time conversion (no addition; same as normal conversion) 0 1: 2-time conversion (addition once) 1 0: 3-time conversion (addition twice)*1 1 1: 4-time conversion (addition three times)	R/W
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	AVEE	Average Mode Enable	0: Addition mode is selected. 1: Average mode is selected.	R/W

Note 1. When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to three times (ADADC.ADC[1:0] = 10b)

ADADC sets the addition count for A/D conversion of the channel, temperature sensor output, and internal reference voltage for which A/D-converted value addition/average mode is selected, and selects either addition or average mode.

ADC[1:0] Bits (Addition Count Select)

The ADC[1:0] bits set the addition count common to the channels for which A/D conversion and A/D-converted value addition/average mode is selected, including the channels selected in double trigger mode (by ADCSR.DBLANS[4:0] bits), and to A/D conversion of temperature sensor output and internal reference voltage.
When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to three times (ADADC.ADC[1:0] = 10b).
The ADC[1:0] bits should be set while the ADCSR.ADST bit is 0. When self-diagnosis is executed (ADCER.DIAGM = 1), do not set the ADC[1:0] bits to any value other than 00b.

AVEE Bit (Average Mode Enable)

The AVEE bit selects addition or average mode for A/D conversion of the channel for which A/D conversion and A/D-converted value addition/average mode is selected, including the channels selected in double trigger mode (by ADCSR.DBLANS[4:0] bits), temperature sensor output, and internal reference voltage.
When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to three times (ADADC.ADC[1:0] = 10b).
The AVEE bits should be set while the ADCSR.ADST bit is 0.

57.2.11 A/D Control Extended Register (ADCER)

Address(es): S12AD.ADCER 0008 900Eh, S12AD1.ADCER 0008 910Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADRFMT	—	—	—	DIAGM	DIAGLD	DIAGVAL[1:0]		—	—	ACE	—	—	ADPRC[1:0]		—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2, b1	ADPRC[1:0]	A/D Conversion Accuracy Specify	b2 b1 0 0: A/D conversion is performed with 12-bit accuracy. 0 1: A/D conversion is performed with 10-bit accuracy. 1 0: A/D conversion is performed with 8-bit accuracy. 1 1: Setting is prohibited.	R/W
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	ACE	A/D Data Register Automatic Clearing Enable	0: Disables automatic clearing. 1: Enables automatic clearing.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	DIAGVAL[1:0]	Self-Diagnosis Conversion Voltage Select	b9 b8 0 0: Setting prohibited when self-diagnosis is enabled 0 1: Uses the voltage of 0 V for self-diagnosis. 1 0: Uses the voltage of reference power supply*1 × 1/2 for self-diagnosis. 1 1: Uses the voltage of reference power supply*1 for self-diagnosis.	R/W
b10	DIAGLD	Self-Diagnosis Mode Select	0: Rotation mode for self-diagnosis voltage 1: Fixed mode for self-diagnosis voltage	R/W
b11	DIAGM	Self-Diagnosis Enable	0: Disables self-diagnosis of 12-bit A/D converter. 1: Enables self-diagnosis of 12-bit A/D converter.	R/W
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	ADRFMT	A/D Data Register Format Select	0: Right-alignment is selected for the A/D data register format. 1: Left-alignment is selected for the A/D data register format.	R/W

Note 1. “Reference voltage” refers to VREFH0 for unit 0 and to AVCC1 for unit 1.

ADCER sets self-diagnosis mode, format of the A/D data registers y (ADDRy), and automatic clearing of A/D data registers.

ADPRC[1:0] Bits (A/D Conversion Accuracy Specify)

These bits select the A/D conversion accuracy among 8-, 10-, or 12-bit accuracy. When the A/D conversion accuracy is changed, the bit width of effective data stored in the result register and A/D conversion time are also changed. See section 57.3.7, Analog Input Sampling and Scan Conversion Time for details. The ADPRC[1:0] bits should be set while the ADCSR.ADST bit is 0.

ACE Bit (A/D Data Register Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (all “0”) of ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, or ADOCDR registers after any of these registers have been read by the CPU, DTC, or DMACA. Automatic clearing of the A/D data register enables a failure which has not been updated in the A/D data register to be detected.

DIAGVAL[1:0] Bits (Self-Diagnosis Conversion Voltage Select)

These bits select the voltage value used in self-diagnosis voltage fixed mode. For details, refer to the descriptions of the ADCER.DIAGLD bit.

Self-diagnosis should not be executed by setting the ADCER.DIAGLD bit to 1 when the ADCER.DIAGVAL[1:0] bits are set to 00b.

DIAGLD Bit (Self-Diagnosis Mode Select)

The DIAGLD bit selects whether the three voltage values are rotated or the fixed voltage is used in self-diagnosis. Setting this bit (ADCER.DIAGLD) to 0 allows conversion of the voltages in rotation mode where 0, the reference power supply × 1/2, and the reference power supply are converted in this order. When self-diagnosis rotation mode is selected after a reset, self-diagnosis is performed from 0 V. When self-diagnosis voltage fixed mode is selected, the fixed voltage specified by the ADCER.DIAGVAL[1:0] bits is converted. In self-diagnosis voltage rotation mode, the self-diagnosis voltage value does not return to 0 when scan conversion is completed. When scan conversion is restarted, therefore, rotation starts at the voltage value following the previous value. If fixed mode is switched to rotation mode, rotation starts at the fixed voltage value.

The DIAGLD bit should be set while the ADCSR.ADST bit is 0.

DIAGM Bit (Self-Diagnosis Enable)

The DIAGM bit enables or disables self-diagnosis.

Self-diagnosis is used to detect a failure of the 12-bit A/D converter. Specifically, one of the internally generated voltage values 0, the reference power supply × 1/2, and the reference power supply is converted. When conversion is completed, information on the converted voltage and the conversion result is stored into the self-diagnosis data register (ADRD). ADRD can then be read out by software to determine whether the conversion result falls within the normal range (normal) or not (abnormal). Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages is converted. The execution time of self-diagnosis differs from the A/D conversion time of one channel.

When self-diagnosis is selected in double trigger mode, self-diagnosis is executed only for the first scan conversion by a synchronous trigger (MTU, ELC, GPT, TMR, TPU), not executed in the second scan. When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed in group A and B.

The DIAGM bit should be set while the ADCSR.ADST bit is 0.

ADRFMT Bit (A/D Data Register Format Select)

The ADRFMT bit specifies right-alignment or left-alignment for the data to be stored in ADDRy, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR, ADRD, or ADCMPDR registers.

For details on the format of each data register, see section 57.2.1, A/D Data Registers y (ADDRy), A/D Data Duplication Register (ADDBLDR), A/D Data Duplication Register A (ADDBLDRA), A/D Data Duplication Register B (ADDBLDRB), A/D Temperature Sensor Data Register (ADTSDR), A/D Internal Reference Voltage Data Register (ADOCDR), section 57.2.2, A/D Self-Diagnosis Data Register (ADRD), and section 57.2.26, A/D Compare Data Register y (ADCMPDRy) (y = 0, 1).

57.2.12 A/D Start Trigger Select Register (ADSTRGR)

Address(es): S12AD.ADSTRGR 0008 9010h, S12AD1.ADSTRG 0008 9110h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	TRSA[5:0]						—	—	TRSB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	TRSB[5:0]	A/D Conversion Start Trigger Select for Group B	Select the A/D conversion start trigger for group B in group scan mode.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	TRSA[5:0]	A/D Conversion Start Trigger Select	Select the A/D conversion start trigger in single scan mode and continuous mode. In group scan mode, the A/D conversion start trigger for group A is selected.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADSTRGR selects the A/D conversion start trigger.

TRSB[5:0] Bits (A/D Conversion Start Trigger Select for Group B)

The TRSB[5:0] bits select the trigger to start scanning of the analog input selected in group B. The TRSB[5:0] bits require to be set only in group scan mode and are not used in any other scan mode. For the scan conversion start trigger for group B, setting a software trigger or an asynchronous trigger is prohibited. Therefore, the TRSB[5:0] bits should be set to the value other than 000000b and the ADCSR.TRGE bit should be set to 1 in group scan mode.

When group A is given priority in group scan mode, setting the ADGSPCR.GBRP bit to 1 allows group B to continuously operate in single scan mode. When setting the ADGSPCR.GBRP bit to 1, set the TRSB[5:0] bits to 3Fh.

Note that the issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (tSCAN). If the issuance period is less than tSCAN, A/D conversion by the trigger may have no effect.

When the trigger from the module operated in 120 MHz (MTU/GPT) is selected as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. See section 57.3.7, Analog Input Sampling and Scan Conversion Time for details.

Table 57.6 lists the A/D conversion startup sources selected by the TRSB[5:0] bits.

TRSA[5:0] Bits (A/D Conversion Start Trigger Select)

The TRSA[5:0] bits select the trigger to start A/D conversion in single scan mode and continuous scan mode. In group scan mode, the trigger to start scanning of the analog input selected in group A is selected. When scanning is executed in group scan mode or double trigger mode, set the TRGE bit in ADCSR register to 1.

- When using the A/D conversion startup source of a synchronous trigger (MTU, ELC, GPT, TMR, TPU), set the TRGE bit in ADCSR register to 1.
- When using the asynchronous trigger (ADTRGn#), set the TRGE bit in ADCSR register to 1 and set the EXTRG bit in ADCSR register to 1.
- Software trigger (ADCSR.ADST) is enabled regardless of the settings of the ADCSR.TRGE bit, the ADCSR.EXTRG bit, and the TRSA[5:0] bits. Note that the issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (tSCAN). If the issuance period is less than tSCAN, A/D conversion by a trigger may have no effect. When the trigger from the module operated in 120 MHz (MTU/GPT) is selected as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. See section 57.3.7, Analog Input Sampling and Scan Conversion Time for details.

Table 57.7 lists the selection of A/D conversion start sources selected by the TRSA[5:0] bits.

Table 57.6 Selection of A/D Activation Sources by the TRSB[5:0] Bits (1/2)

Module	Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
Trigger source de-selection state			1	1	1	1	1	1
MTU	TRGA0N	Compare match with or input capture to MTU0.TGRA	0	0	0	0	0	1
	TRGA1N	Compare match with or input capture to MTU1.TGRA	0	0	0	0	1	0
	TRGA2N	Compare match with or input capture to MTU2.TGRA	0	0	0	0	1	1
	TRGA3N	Compare match with or input capture to MTU3.TGRA	0	0	0	1	0	0
	TRGA4N	Compare match with or input capture to MTU4.TGRA, or an underflow of MTU4.TCNT (in the trough) in complementary PWM mode	0	0	0	1	0	1
	TRGA6N	Compare match with or input capture to MTU6.TGRA	0	0	0	1	1	0
	TRGA7N	Compare match with or input capture to MTU7.TGRA, or an underflow of MTU7.TCNT (in the trough) in complementary PWM mode	0	0	0	1	1	1
	TRG0N	Compare match with MTU0.TGRE	0	0	1	0	0	0
	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	1	0	0	1
	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	0
	TRG4AN or TRG4BN	Compare match between MTU4.TADCORA and MTU4.TCNT, or between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	1
	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is in use)	0	0	1	1	0	0
	TRG7AN	Compare match between MTU7.TADCORA and MTU7.TCNT	0	0	1	1	0	1
	TRG7BN	Compare match between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	0
	TRG7AN or TRG7BN	Compare match between MTU7.TADCORA and MTU7.TCNT, or between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	1
	TRG7ABN	Compare match between MTU7.TADCORA and MTU7.TCNT, and between MTU7.TADCORB and MTU7.TCNT (when interrupt skipping function 2 is in use)	0	1	0	0	0	0
GPT	GTADTRA0N	Compare match with GPT0.GTADTRA	0	1	0	0	0	1
	GTADTRB0N	Compare match with GPT0.GTADTRB	0	1	0	0	1	0
	GTADTRA1N	Compare match with GPT1.GTADTRA	0	1	0	0	1	1
	GTADTRB1N	Compare match with GPT1.GTADTRB	0	1	0	1	0	0
	GTADTRA2N	Compare match with GPT2.GTADTRA	0	1	0	1	0	1
	GTADTRB2N	Compare match with GPT2.GTADTRB	0	1	0	1	1	0
	GTADTRA3N	Compare match with GPT3.GTADTRA	0	1	0	1	1	1
	GTADTRB3N	Compare match with GPT3.GTADTRB	0	1	1	0	0	0
	GTADTRA0N or GTADTRB0N	Compare match with GPT0.GTADTRA or with GPT0.GTADTRB	0	1	1	0	0	1
	GTADTRA1N or GTADTRB1N	Compare match with GPT1.GTADTRA or with GPT1.GTADTRB	0	1	1	0	1	0
	GTADTRA2N or GTADTRB2N	Compare match with GPT2.GTADTRA or with GPT2.GTADTRB	0	1	1	0	1	1
	GTADTRA3N or GTADTRB3N	Compare match with GPT3.GTADTRA or with GPT3.GTADTRB	0	1	1	1	0	0
TMR	TMTRG0AN_0	Compare match between TMR0.TCORA0 and TMR0.TCNT0 (unit 0, ch 0)	0	1	1	1	0	1
	TMTRG0AN_1	Compare match between TMR2.TCORA0 and TMR2.TCNT0 (unit 1, ch 0)	0	1	1	1	1	0
TPU	TPTRGAN	Compare match with or input capture to TPUn.TGRAn (n = 0 to 5)	0	1	1	1	1	1
	TPTRG0AN	Compare match with or input capture to TPU0.TGRA0	1	0	0	0	0	0

Table 57.6 Selection of A/D Activation Sources by the TRSB[5:0] Bits (2/2)

Module	Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
ELC	ELCTRG0N/ ELCTRG1N		1	1	0	0	0	0

Table 57.7 Selection of A/D Activation Sources by the TRSA[5:0] Bits (1/2)

Module	Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
Trigger source de-selection state			1	1	1	1	1	1
External	ADTRG0#	Input pin for the trigger	0	0	0	0	0	0
MTU	TRGA0N	Compare match with or input capture to MTU0.TGRA	0	0	0	0	0	1
	TRGA1N	Compare match with or input capture to MTU1.TGRA	0	0	0	0	1	0
	TRGA2N	Compare match with or input capture to MTU2.TGRA	0	0	0	0	1	1
	TRGA3N	Compare match with or input capture to MTU3.TGRA	0	0	0	1	0	0
	TRGA4N	Compare match with or input capture to MTU4.TGRA or, in complementary PWM mode, an underflow of MTU4.TCNT (in the trough)	0	0	0	1	0	1
	TRGA6N	Compare match with or input capture to MTU6.TGRA	0	0	0	1	1	0
	TRGA7N	Compare match with or input capture to MTU7.TGRA or, in complementary PWM mode, an underflow of MTU7.TCNT (in the trough)	0	0	0	1	1	1
	TRG0N	Compare match with MTU0.TGRE	0	0	1	0	0	0
	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	1	0	0	1
	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	0
	TRG4AN or TRG4BN	Compare match between MTU4.TADCORA and MTU4.TCNT, or between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	1
	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is in use)	0	0	1	1	0	0
	TRG7AN	Compare match between MTU7.TADCORA and MTU7.TCNT	0	0	1	1	0	1
	TRG7BN	Compare match between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	0
	TRG7AN or TRG7BN	Compare match between MTU7.TADCORA and MTU7.TCNT, or between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	1
	TRG7ABN	Compare match between MTU7.TADCORA and MTU7.TCNT, and between MTU7.TADCORB and MTU7.TCNT (when interrupt skipping function 2 is in use)	0	1	0	0	0	0
GPT	GTADTRA0N	Compare match with GPT0.GTADTRA	0	1	0	0	0	1
	GTADTRB0N	Compare match with GPT0.GTADTRB	0	1	0	0	1	0
	GTADTRA1N	Compare match with GPT1.GTADTRA	0	1	0	0	1	1
	GTADTRB1N	Compare match with GPT1.GTADTRB	0	1	0	1	0	0
	GTADTRA2N	Compare match with GPT2.GTADTRA	0	1	0	1	0	1
	GTADTRB2N	Compare match with GPT2.GTADTRB	0	1	0	1	1	0
	GTADTRA3N	Compare match with GPT3.GTADTRA	0	1	0	1	1	1
	GTADTRB3N	Compare match with GPT3.GTADTRB	0	1	1	0	0	0
	GTADTRA0N or GTADTRB0N	Compare match with GPT0.GTADTRA or with GPT0.GTADTRB	0	1	1	0	0	1
	GTADTRA1N or GTADTRB1N	Compare match with GPT1.GTADTRA or with GPT1.GTADTRB	0	1	1	0	1	0
	GTADTRA2N or GTADTRB2N	Compare match with GPT2.GTADTRA or with GPT2.GTADTRB	0	1	1	0	1	1
	GTADTRA3N or GTADTRB3N	Compare match with GPT3.GTADTRA or with GPT3.GTADTRB	0	1	1	1	0	0

Table 57.7 Selection of A/D Activation Sources by the TRSA[5:0] Bits (2/2)

Module	Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
TMR	TMTRG0AN_0	Compare match between TMR0.TCORA0 and TMR0.TCNT0 (unit 0, ch 0)	0	1	1	1	0	1
	TMTRG0AN_1	Compare match between TMR2.TCORA0 and TMR2.TCNT0 (unit 1, ch 0)	0	1	1	1	1	0
TPU	TPTRGAN	Compare match with or input capture to TPUn.TGRAn (n = 0 to 5)	0	1	1	1	1	1
	TPTRG0AN	Compare match with or input capture to TPU0.TGRA0	1	0	0	0	0	0
ELC	ELCTRG0N/ ELCTRG1N		1	1	0	0	0	0

57.2.13 A/D Conversion Extended Input Control Register (ADEXICR)

Address(es): S12AD1.ADEXICR 0008 9112h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EXOEN	EXSEL[1:0]		—	OCSB	TSSB	OCSA	TSSA	—	—	—	—	—	—	OCSAD	TSSAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSSAD	Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select	0: Temperature sensor output A/D-converted value addition/average mode is not selected. 1: Temperature sensor output A/D-converted value addition/average mode is selected.	R/W
b1	OCSAD	Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select	0: Internal reference voltage A/D-converted value addition/average mode is not selected. 1: Internal reference voltage A/D-converted value addition/average mode is selected.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TSSA	Temperature Sensor Output A/D Conversion Select	This bit is used for A/D conversion for group A in single scan mode, continuous scan mode, or group scan mode.	R/W
b9	OCSA	Internal Reference Voltage A/D Conversion Select	This bit is used for A/D conversion for group A in single scan mode, continuous scan mode, or group scan mode.	R/W
b10	TSSB	Temperature Sensor Output A/D Conversion Select	This bit is used for A/D conversion for group B in group scan mode.	R/W
b11	OCSB	Internal Reference Voltage A/D Conversion Select	This bit is used for A/D conversion for group B in group scan mode.	R/W
b12	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14, b13	EXSEL[1:0]	Extended Analog Input Select	b14 b13 0 0: Analog input channel (ANn) 0 1: ANEX1 1 0: Setting is prohibited. 1 1: Setting is prohibited.	R/W
b15	EXOEN	Extended Analog Output Control	0: Output is disabled. 1: Output is enabled.	R/W

TSSAD Bit (Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select)

When the TSSAD bit is set to 1, A/D conversion of the temperature sensor output is selected and performed successively 2 to 4 times that is set with the ADC[1:0] bits in ADADC register. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is returned to the A/D temperature sensor data register (ADTSDR). When the ADADC.AVEE bit is 1, the mean value is returned to ADTSDR.

The TSSAD bit should be set while the ADCSR.ADST bit is 0.

OCSAD Bit (Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select)

When the OCSAD bit is set to 1, A/D conversion of the internal reference voltage is selected and performed successively 2 to 4 times that is set with the ADC[1:0] bits in ADADC register. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is returned to the A/D internal reference voltage data register (ADOCDR). When the ADADC.AVEE bit is 1, the mean value is returned to ADOCDR.
The OCSAD bit should be set while the ADCSR.ADST bit is 0.

TSSA Bit (Temperature Sensor Output A/D Conversion Select)

This bit selects A/D conversion of the temperature sensor output for group A in single scan mode, continuous scan mode or group scan mode. When A/D conversion of the temperature sensor output is selected and performed, set the ADCSR.DBLE bit to 0.
The TSSA bit should be set while the ADCSR.ADST bit is 0.

OCSA Bit (Internal Reference Voltage A/D Conversion Select)

This bit selects A/D conversion of the internal reference voltage for group A in single scan mode, continuous scan mode or group scan mode. When A/D conversion of the internal reference voltage is selected and performed, set the ADCSR.DBLE bit to 0.
The OCSA bit should be set while the ADCSR.ADST bit is 0. After the OCSA bit has been set to 1, wait for at least 400 ns to start A/D conversion.

TSSB Bit (Temperature Sensor Output A/D Conversion Select)

This bit selects A/D conversion of the temperature sensor output for group B in group scan mode.
The TSSB bit should be set while the ADCSR.ADST bit is 0. Do not set the TSSB bit to 1 while the TSSA bit is 1.

OCSB Bit (Internal Reference Voltage A/D Conversion Select)

This bit selects A/D conversion of the internal reference voltage for group B in group scan mode.
The OCSB bit should be set while the ADCSR.ADST bit is 0. Do not set the OCSB bit to 1 while the OCSA bit is 1. After the OCSB bit has been set to 1, wait for at least 400 ns to start A/D conversion.

EXSEL[1:0] Bits (Extended Analog Input Select)

These bits select the extended analog input (ANEX1) instead of the analog input channels (ANn).
When ANEX1 is selected, ANEX0 should be input to ANEX1 via the external operational amplifier. In this case, only select AN100 to AN107; do not select AN108 to AN120. For details, see section 57.3.5.1, Usage of ANEX1.

EXOEN Bit (Extended Analog Output Control)

This bit controls the extended analog output (ANEX0). When the output is enabled, the multiplexed value of AN100 to AN107 among analog input channels in unit 1 is output to ANEX0. Do not enable an output when the EXSEL[1:0] bits are 00b.

57.2.14 A/D Sampling State Register n (ADSSTRn) (n = 0 to 7, L, T, O)

Address(es): S12AD.ADSSTR0 0008 9060h, S12AD.ADSSTR1 0008 9073h, S12AD.ADSSTR2 0008 9074h, S12AD.ADSSTR3 0008 9075h, S12AD.ADSSTR4 0008 9076h, S12AD.ADSSTR5 0008 9077h, S12AD.ADSSTR6 0008 9078h, S12AD.ADSSTR7 0008 9079h, S12AD1.ADSSTR0 0008 9160h, S12AD1.ADSSTR1 0008 9173h, S12AD1.ADSSTR2 0008 9174h, S12AD1.ADSSTR3 0008 9175h, S12AD1.ADSSTR4 0008 9176h, S12AD1.ADSSTR5 0008 9177h, S12AD1.ADSSTR6 0008 9178h, S12AD1.ADSSTR7 0008 9179h, S12AD1.ADSSTRL 0008 9161h, S12AD1.ADSSTRT 0008 9170h, S12AD1.ADSSTRO 0008 9171h

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	SST[7:0]	Sampling Time Setting	These bits set the sampling time in the range from 5 to 255 states.	R/W

ADSSTRn sets the sampling time for analog input.
If one state is one ADCLK (A/D conversion clock) cycle and the ADCLK clock is 60 MHz, one state is 16.7 ns. The initial value is 11 states. If the impedance of analog input signal source is too high to secure sufficient sampling time or if the ADCLK clock is slow, the sampling time can be adjusted. The SST[7:0] bits should be set while the ADCSR.ADST bit is 0. The sampling time must be set to a value that is 5 states or more and is 255 or less. Table 57.8 shows the relationship between the A/D sampling state register and the relevant channels.
For details, refer to section 57.3.7, Analog Input Sampling and Scan Conversion Time.

Table 57.8 Relationship between A/D Sampling State Register and Relevant Channels

	Corresponding Channels	
Bit Name	**Unit 0**	**Unit 1**
ADSSTR0.SST[7:0] bits*1	AN000	AN100
ADSSTR1.SST[7:0] bits	AN001	AN101
ADSSTR2.SST[7:0] bits	AN002	AN102
ADSSTR3.SST[7:0] bits	AN003	AN103
ADSSTR4.SST[7:0] bits	AN004	AN104
ADSSTR5.SST[7:0] bits	AN005	AN105
ADSSTR6.SST[7:0] bits	AN006	AN106
ADSSTR7.SST[7:0] bits	AN007	AN107
ADSSTRL.SST[7:0] bits	—	AN108 to AN120
ADSSTRT.SST[7:0] bits	—	Temperature sensor output (in unit 1 only)
ADSSTRO.SST[7:0] bits	—	Internal reference voltage (in unit 1 only)

Note 1. When self-diagnosis function is selected, the sampling time set by the ADSSTR0.SST[7:0] bits is applied to it.

57.2.15 A/D Sample-and-Hold Circuit Control Register (ADSHCR)

Address(es): S12AD.ADSHCR 0008 9066h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SHANS[2:0]			SSTSH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	SSTSH[7:0]	Channel-Dedicated Sample-and-Hold Circuit Sampling Time Setting	Set the sampling time (4 to 255 states).	R/W
b10 to b8	SHANS[2:0]	Channel-Dedicated Sample-and-Hold Circuit Bypass Select	Select whether to use or not use (bypass) AN000 to AN002 channel-dedicated sample-and-hold circuits. 0: Bypass the channel-dedicated sample-and-hold circuits. 1: Use the channel-dedicated sample-and-hold circuits.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADSHCR sets the parameters related to channel-dedicated sample-and-hold circuits.

SSTSH[7:0] Bits (Channel-Dedicated Sample-and-Hold Circuit Sampling Time Setting)

The SSTSH[7:0] bits set the sampling time for the channel-dedicated sample-and-hold circuits when the ADSHMSR.SHMD bit is 0. If one state is one ADCLK (A/D conversion clock) cycle and the ADCLK clock is 60 MHz, one state is 16.7 ns. The initial value is 24 states. If the impedance of analog input signal source is too high to secure sufficient sampling time or if the ADCLK clock is slow, the sampling time can be adjusted. The SSTSH[7:0] bits should be set while the ADCSR.ADST bit is 0. The sampling time must be set to a value that is 4 states or more and is 255 or less.

SHANS[2:0] Bits (Channel-Dedicated Sample-and-Hold Circuit Bypass Select)

The SHANS[2:0] bits select whether to use or not use (bypass) AN000 to AN002 channel-dedicated sample-and-hold circuits. The SHANS[0] bit selects AN000, SHANS[1] bit selects AN001, and SHANS[2] bit selects AN002. The SHANS[2:0] bits should be set while the ADCSR.ADST and ADSHMSR.SHMD bits are 0.

If any channel from among AN000 to AN002 is selected for group B while operation is in group scan mode under group A priority control, make the setting to bypass the channel's dedicated sample-and-hold circuit.

The channels in unit 1 do not include channel-dedicated sample-and-hold circuits.

57.2.16 A/D Sample-and-Hold Circuit Operating Mode Select Register (ADSHMSR)

Address(es): S12AD.ADSHMSR 0008 907Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SHMD
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SHMD	Channel-Dedicated Sample-and-Hold Circuit Operating Mode Select	0: Continuous sampling by the channel-dedicated sample-and-hold circuits is disabled. 1: Continuous sampling by the channel-dedicated sample-and-hold circuits is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADSHMSR enables or disables continuous sampling by the channel-dedicated sample-and-hold circuits.

SHMD Bit (Channel-Dedicated Sample-and-Hold Circuit Operating Mode Select)

Setting the SHMD bit to 1 enables continuous sampling by the channel-dedicated sample-and-hold circuits selected by the ADSHCR.SHANS[2:0] bits. The SHMD bit should be set while the ADCSR.ADST bit is 0.

When continuous sampling is enabled, continuous sampling proceeds while the 12-bit A/D converter is in a waiting state and holding proceeds during A/D conversion.

57.2.17 A/D Disconnection Detection Control Register (ADDISCR)

Address(es): S12AD.ADDISCR 0008 907Ah, S12AD1.ADDISCR 0008 917Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ADNDIS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	ADNDIS[4:0]	Disconnection Detection Assist Setting	Disconnection detection assist function is set.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADDISCR sets the disconnection detection assist function.

ADNDIS[4:0] Bits (Disconnection Detection Assist Setting)

These bits selects either precharge or discharge and the period of precharge/discharge for the A/D disconnection detection assist function. Setting the ADNDIS[4] bit = 1 allows to select precharge and setting the ADNDIS[4] bit = 0 allows to select discharge. The period of precharge/discharge can be set with the ADNDIS[3:0] bits. When the ADNDIS[3:0] bits = 0000b, the disconnection detection assist function is not effective. Setting of the ADNDIS[3:0] bits to 0001b is prohibited. Except for the case of the ADNDIS[3:0] = 0000b or 0001b, the specified value indicates the number of states for the period of precharge/discharge. When the temperature sensor output or internal reference voltage is converted or self-diagnosis is used, the disconnection detection assistance cannot be used. In that case, the ADNDIS[3:0] bits should be set to 0000b. When the ADNDIS[3:0] bits are set to any values other than 0000b or 0001b, and the disconnection detection assistance is enabled, the disconnection detection assistance for the channel-dedicated sample-and-hold circuit used for analog inputs are also enabled.

57.2.18 A/D Group Scan Priority Control Register (ADGSPCR)

Address(es): S12AD.ADGSPCR 0008 9080h, S12AD1.ADGSPCR 0008 9180h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GBRP	—	—	—	—	—	—	—	—	—	—	—	—	—	GBRSCN	PGS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PGS	Group-A Priority Control Setting*1	0: Operation is without group A priority control 1: Operation is with group A priority control	R/W
b1	GBRSCN	Group B Restart Setting*2	(Enabled only when PGS = 1. Reserved when PGS = 0.) 0: Scanning for group B is not restarted after having been discontinued due to group A priority control. 1: Scanning for group B is restarted after having been discontinued due to group A priority control.	R/W
b14 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	GBRP	Group B Single Scan Continuous Start*3	(Enabled only when PGS = 1. Reserved when PGS = 0.) 0: Single scan for group B is not continuously activated. 1: Single scan for group B is continuously activated.	R/W

Note 1. When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode). If the bits are set to any other values, proper operation is not guaranteed.
Note 2. When the GBRSCN bit is set to 1, clock frequency ratio of PCLKB to ADCLK should be 1:1.
Note 3. When the GBRP bit has been set to 1, single scan is performed continuously for group B regardless of the setting of the GBRSCN bit.

ADGSPCR sets priority control of A/D conversion for group A in group scan mode.

PGS Bit (Group-A Priority Control Setting)

This bit sets the priority of operation on group A. Set this bit to 1 when giving priority to operation on group A.
When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode). If the bits are set to any other values, proper operation is not guaranteed.
When the PGS bit has been set to 0, clear operation must be performed by software according to section 57.5.2, Notes on Stopping A/D Conversion. When the PGS bit has been set to 1, make settings according to section 57.3.4.3, Operation under Group-A Priority Control.

GBRSCN Bit (Group B Restart Setting)

This bit controls the restarting of scan operation on group B when operation on group A is given priority.
If a scan operation on group B has been stopped by a group A trigger input with the GBRSCN bit set to 1, the scan operation is restarted on completion of the A/D conversion on group A. Also, if a group B trigger is input during A/D conversion on group A, the scan operation on group B is restarted on completion of the A/D conversion on group A.
If the GBRSCN bit has been set to 0, triggers that are input during A/D conversion are ignored. Also, the ADCSR.ADST bit must be 0 when the GBRSCN bit is to be set.
The setting of the GBRSCN bit has an effect when the PGS bit is set to 1.

GBRP Bit (Group B Single Scan Continuous Start)

This bit is set when a single scan operation is to be performed continuously on group B.
Setting the GBRP bit to 1 starts a single scan on group B. On completion of the scan, another single scan on group B is automatically started. If an A/D conversion on group B has been stopped due to an operation on group A that takes priority, single scan on group B is automatically restarted on completion of the A/D conversion on group A.
Disable group B trigger input before setting the GBRP bit to 1. Setting the GBRP bit to 1 invalidates the setting of the

GBRSCN bit. The ADCSR.ADST bit must be 0 when the GBRP bit is to be set.
The setting of the GBRP bit is valid when the PGS bit is 1.

57.2.19 A/D Compare Control Register (ADCMPCR)

Address(es): S12AD.ADCMPCR 0008 9090h, S12AD1.ADCMPCR 0008 9190h

	b7	b6	b5	b4	b3	b2	b1	b0
	CMPIE	WCMPE	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	WCMPE	Window Function	0: Window function disabled 1: Window function enabled	R/W
b7	CMPIE	Compare Interrupt Enable	0: Generation of an S12CMPI interrupt in response to matches with a condition for comparison is disabled. 1: Generation of an S12CMPI interrupt in response to matches with a condition for comparison is enabled.	R/W

ADCMPCR sets the compare function.

WCMPE Bit (Window Function)

The WCMPE bit selects enabling or disabling of Window function. This bit should be set while the ADCSR.ADST bit is 0.

CMPIE Bit (Compare Interrupt Enable)

The CMPIE bit enables or disables generation of a compare interrupt (S12CMPI) in response to matches with a condition for comparison.

57.2.20 A/D Compare Channel Select Register 0 (ADCMPANSR0)

Address(es): S12AD.ADCMPANSR0 0008 9094h, S12AD1.ADCMPANSR0 0008 9194h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPS0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	CMPS0[15:0]	Compare Channel Select	0: The corresponding channel from among AN000 to AN007 and AN100 to AN115 is not a target for comparison. 1: The corresponding channel from among AN000 to AN007 and AN100 to AN115 is a target for comparison.	R/W

ADCMPANSR0 selects analog input channels for comparison from among AN000 to AN007 and AN100 to AN115.

CMPS0[15:0] Bits (Compare Channel Select)

Setting the CMPS0[n] bit which has the same number as the A/D channel selected by the ADANSA0.ANSA0[n] (n = 0 to 7 for unit 0, n = 0 to 15 for unit 1) or ADANSB0.ANSB0[n] (n = 0 to 7 for unit 0, n = 0 to 15 for unit 1) bits to 1 enables comparison with that channel.

Set the CMPS0[15:0] bits while ADCSR.ADST bit is 0.

The CMPS0[15:8] bits of the unit 0 are reserved bits. Reading to these bits returns 0. Writing value should be 0.

57.2.21 A/D Compare Channel Select Register1 (ADCMPANSR1)

Address(es): S12AD1.ADCMPANSR1 0008 9196h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	CMPS1[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	CMPS1[4:0]	Compare Channel Select	0: The corresponding channel from among AN116 to AN120 is not a target for comparison. 1: The corresponding channel from among AN116 to AN120 is a target for comparison.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADCMPANSR1 selects analog input channels for comparison from among AN116 to AN120.

CMPS1[4:0] Bits (Compare Channel Select)

Setting the CMPS1[n] which has the same number as the A/D channel selected by the ADANSB1.ANSB1[n] (n = 0 to 4) bits to 1 enables comparison with that channel.

Set the CMPS1[4:0] bits while ADCSR.ADST bit is 0.

57.2.22 A/D Compare Channel Select Extended Register (ADCMPANSER)

Address(es): S12AD1.ADCMPANSER 0008 9192h

	b7	b6	b5	b4	b3	b2	b1	b0
			—	—	—	—	CMPSOC	CMPSTS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPSTS	Temperature Sensor Output Compare Select	0: Temperature sensor output is not a target for comparison. 1: Temperature sensor output is a target for comparison.	R/W
b1	CMPSOC	Internal Reference Voltage Compare Select	0: Internal reference voltage is not a target for comparison. 1: Internal reference voltage is a target for comparison.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPSTS Bit (Temperature Sensor Output Compare Select)

Setting the CMPSTS bit to 1 while ADEXICR.TSSA or ADEXICR.TSSB = 1 enables comparison. This bit should be set while the ADCSR.ADST bit is 0.

CMPSOC Bit (Internal Reference Voltage Compare Select)

Setting the CMPSOC bit to 1 while ADEXICR.OCSA or ADEXICR.OCSB = 1 enables comparison. This bit should be set while the ADCSR.ADST bit is 0.

57.2.23 A/D Compare Level Register 0 (ADCMPLR0)

Address(es): S12AD.ADCMPLR0 0008 9098h, S12AD1.ADCMPLR0 0008 9198h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPL0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	CMPL0[15:0]	Compare Level Select	Set the condition for comparison with the selected channels from among AN000 to AN007 and AN100 to AN115. When Window function is disabled (ADCMPCR.WCMPE bit = 0): 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When Window function is enabled (ADCMPCR.WCMPE bit = 1): 0: AD-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W

ADCMPLR0 sets the condition for use in comparing the values of the ADCMPDR0 and ADCMPDR1 registers with results of A/D conversion.
Set the ADCMPLR0 register while ADCSR.ADST is 0.

CMPL0[15:0] Bits (Compare Level Select)

The CMPL0[15:0] bits set the condition for use in comparison with the selected channel from among AN000 to AN007 and AN100 to AN115. A condition can be set for individual comparison of each analog input.
The CMPL0[0] bit is used for AN000 (unit0) and AN100 (unit 1), the CMPL0[7] bit is used for AN007 (unit 0) and AN107 (unit 1), and the CMPL0[15] bit is used for AN115 (unit 1).
When the result of comparison matches the set condition, ADCMPSR0.CMPF0n is set to 1 and a compare interrupt (S12CMPI) is generated.
The CMPL0[15:8] bits of the unit 0 are reserved. Reading to these bits return 0. Writing value should be 0.

57.2.24 A/D Compare Level Register1 (ADCMPLR1)

Address(es): S12AD1.ADCMPLR1 0008 919Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	CMPL1[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	CMPL1[4:0]	Compare Level Select	Set the condition for comparison with the selected channels from among AN116 to AN120. When Window function is disabled (ADCMPCR.WCMPE bit = 0): 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When Window function is enabled (ADCMPCR.WCMPE bit = 1): 0: AD-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADCMPLR1 sets the condition for use in comparing the values of the ADCMPDR0 and ADCMPDR1 registers with results of A/D conversion.
Set the ADCMPLR1 register while ADCSR.ADST is 0.

CMPL1[4:0] Bits (Compare Level Select)

The CMPL1[4:0] bits set the condition for use in comparison with the selected channel from among AN116 to AN120. A condition can be set for individual comparison of each analog input.
The CMPL1[0] bit is used for AN116 and the CMPL1[4] bit is used for AN120.
When the result of comparison matches the set condition, ADCMPSR1.CMPF1n is set to 1 and a compare interrupt (S12CMPI) is generated.

57.2.25 A/D Compare Level Extended Register (ADCMPLER)

Address(es): S12AD1.ADCMPLER 0008 9193h

	b7	b6	b5	b4	b3	b2	b1	b0
			—	—	—	—	CMPLOC	CMPLTS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPLTS	Temperature Sensor Output Compare Level Select	When Window function is disabled (ADCMPCR.WCMPE bit = 0): 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When Window function is enabled (ADCMPCR.WCMPE bit = 1): 0: AD-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W
b1	CMPLOC	Internal Reference Voltage Compare Level Select	When Window function is disabled (ADCMPCR.WCMPE bit = 0): 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value When Window function is enabled (ADCMPCR.WCMPE bit = 1): 0: AD-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADCMPLER sets the condition for use in comparing the values of ADCMPDR0 and ADCMPDR1 registers with results of A/D conversion.

Set the ADCMPLER register while ADCSR.ADST is 0.

CMPLTS Bit (Temperature Sensor Output Compare Level Select)

This bit sets the condition for use in comparison with temperature sensor output.

When the result of comparison matches the set condition, ADCMPSER.CMPFTS is set to 1 and a compare interrupt (S12CMPI) is generated.

CMPLOC Bit (Internal Reference Voltage Compare Level Select)

This bit sets conditions for use in comparison with internal reference voltage.

When the result of comparison matches the set condition, ADCMPSER.CMPFOC is set to 1 and a compare interrupt (S12CMPI) is generated.

57.2.26 A/D Compare Data Register y (ADCMPDRy) (y = 0, 1)

Set the reference data for comparison with the selected channels.
This register is accessible even during A/D conversion which allows dynamic changing of the reference data. The ADCMPDR1 register is not used when the Window function is disabled.
The ADCMPDRy register uses different formats depending on the following conditions.

- The value of A/D data register format select bit (flush-right or flush-left)
- The value of A/D-conversion accuracy specification bit (12 bits, 10 bits, 8 bits)
- The value of A/D-converted value addition/average mode select register (A/D-converted value addition mode selected or not selected)

(1) When A/D-Converted Value Addition Mode is Not Selected

- Flush-right format with 12-bit precision

Address(es): S12AD.ADCMPDR0 0008 909Ch, S12AD.ADCMPDR1 0008 909Eh, S12AD1.ADCMPDR0 0008 919Ch, S12AD1.ADCMPDR1 0008 919Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMPD[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	CMPD[11:0]	—	12-bit reference value	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- Flush-right format with 10-bit precision

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

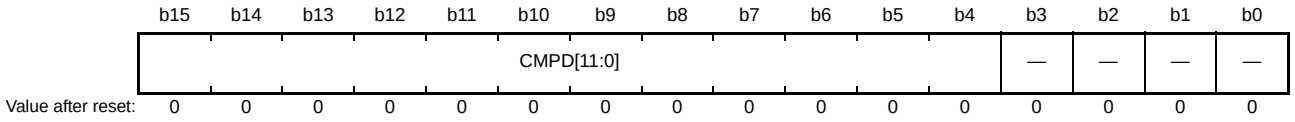
Bit	Symbol	Bit Name	Description	R/W
b9 to b0	CMPD[9:0]	—	10-bit reference value	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- Flush-right format with 8-bit precision

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CMPD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

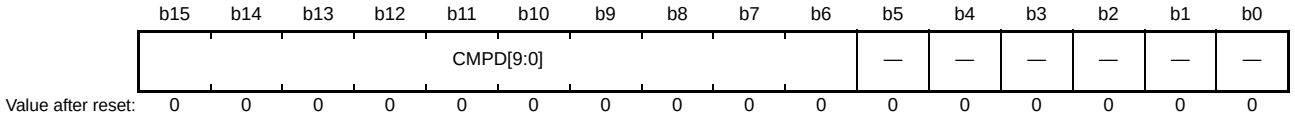
Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CMPD[7:0]	—	8-bit reference value	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- Flush-left data with 12-bit precision



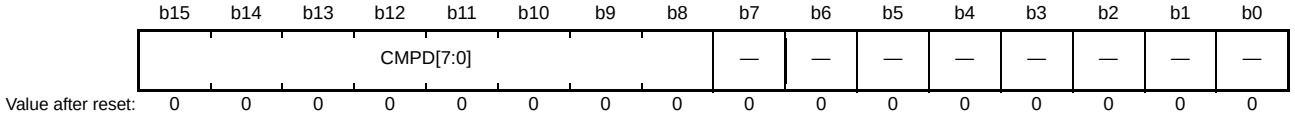
Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b4	CMPD[11:0]	—	12-bit reference value	R/W

- Flush-left data with 10-bit precision



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b6	CMPD[9:0]	—	10-bit reference value	R/W

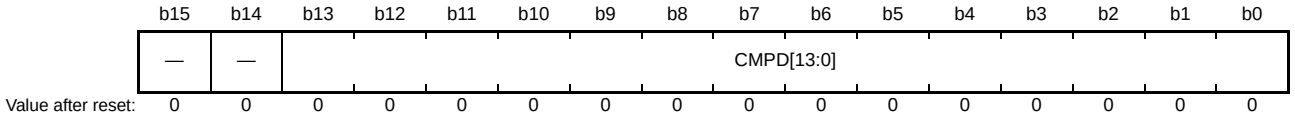
- Flush-left data with 8-bit precision



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	CMPD[7:0]	—	8-bit reference value	R/W

(2) When A/D-Converted Value Addition Mode is Selected

- Flush-right format with 12-bit precision (in A/D-converted value addition mode)



Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CMPD[13:0]	—	14-bit reference value	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- Flush-right format with 10-bit precision (in A/D-converted value addition mode)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMPD[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	CMPD[11:0]	—	12-bit reference value	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- Flush-right format with 8-bit precision (in A/D-converted value addition mode)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	CMPD[9:0]	—	10-bit reference value	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- Flush-left format with 12-bit precision (in A/D-converted value addition mode)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPD[13:0]														—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b2	CMPD[13:0]	—	14-bit reference value	R/W

- Flush-left format with 10-bit precision (in A/D-converted value addition mode)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPD[11:0]												—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b4	CMPD[11:0]	—	12-bit reference value	R/W

- Flush-left format with 8-bit precision (in A/D-converted value addition mode)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPD[9:0]										—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b6	CMPD[9:0]	—	10-bit reference value	R/W

57.2.27 A/D Compare Status Register0 (ADCMPSR0)

Address(es): S12AD.ADCMPSR0 0008 90A0h, S12AD1.ADCMPSR0 0008 91A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPF0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	CMPF0[15:0]	Compare Flag	Indicates compare results of AN000 to AN007 and AN100 to AN115. 0: Condition for comparison was not met. 1: Condition for comparison was met.	R/W

Compare results using compare function are stored in the ADCMPSR0 register.

CMPF0[15:0] Bits (Compare Flag)

These bits are status flags to indicate the results of comparison with the selected analog inputs from among AN000 to AN007 and AN100 to AN115. When the result of comparison on completion of A/D conversion matches the condition set in ADCMPLR0.CMPL0n, the corresponding flags are set to 1.

When the ADCMPCR.CMPIE bit is 1, a compare interrupt (S12CMPI) request is generated when the setting of the flag becomes 1.

The CMPF0[0] bit is used for AN000 (unit 0)/AN100 (unit 1), the CMPF0[7] bit is used for AN007 (unit 0)/AN107 (unit 1), and the CMPF0[15] bit is used for AN115 (unit 1).

The value 1 cannot be written to the CMPF0n bit.

The CMPF0[15:8] bits of the unit 0 are reserved. Reading to these bits return 0. Writing value should be 0.

[Setting condition]

- The condition set in ADCMPLR0.CMPL0n is met.

[Clearing condition]

- 0 is written after reading 1.

57.2.28 A/D Compare Status Register1 (ADCMPSR1)

Address(es): S12AD1.ADCMPSR1 0008 91A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	CMPF1[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	CMPF1[4:0]	Compare Flag	Indicates compare results of AN116 to AN120. 0: Condition for comparison was not met. 1: Condition for comparison was met.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Compare results using compare function are stored in the ADCMPSR1 register.

CMPF1[4:0] Bits (Compare Flag)

These bits are status flags to indicate the results of comparison with the selected analog inputs from among AN116 to AN120. When the result of comparison on completion of A/D conversion matches the condition set in ADCMPLR1.CMPL1n, the corresponding flags are set to 1.

When the ADCMPCR.CMPIE bit is 1, a compare interrupt (S12CMPI) request is generated when the setting of the flag becomes 1.

The CMPF1[0] bit is used for AN116 and the CMPF1[4] bit is used for AN120.

The value 1 cannot be written to the CMPF1n bit.

[Setting condition]

- The condition set in ADCMPLR1.CMPL1n is met.

[Clearing condition]

- 0 is written after reading 1.

57.2.29 A/D Compare Status Extended Register (ADCMPSER)

Address(es): S12AD1.ADCMPSER 0008 91A4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPFOC	CMPFTS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPFTS	Temperature Sensor Output Compare Flag	0: Condition for comparison was not met. 1: Condition for comparison was met.	R/W
b1	CMPFOC	Internal Reference Voltage Compare Flag	0: Condition for comparison was not met. 1: Condition for comparison was met.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Compare results using compare function are stored in the ADCMPSER register.

CMPFTS Bit (Temperature Sensor Output Compare Flag)

This bit is a status flag to indicate the result of comparison with temperature sensor output. When the result of comparison on completion of A/D conversion matches the condition set in ADCMPLER.CMPLTS bit, this flag is set to 1. When the ADCMPCR.CMPIE bit is 1, a compare interrupt (S12CMPI) request is generated when the setting of the flag becomes 1.

The value 1 cannot be written to the CMPFTS bit.

[Setting condition]

- The condition set in ADCMPLER.CMPLTS bit is met.

[Clearing condition]

- 0 is written after reading 1.

CMPFOC Bit (Internal Reference Voltage Compare Flag)

This bit is a status flag to indicate the result of comparison with internal reference voltage. When the result of comparison on completion of A/D conversion matches the condition set in ADCMPLER.CMPLOC bit, this flag is set to 1. When the ADCMPCR.CMPIE bit is 1, a compare interrupt (S12CMPI) request is generated when the setting of the flag becomes 1.

The value 1 cannot be written to the CMPFOC bit.

[Setting condition]

- The condition set in ADCMPLER.CMPLOC bit is met.

[Clearing condition]

- 0 is written after reading 1.

57.3 Operation

57.3.1 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.
A scan conversion is performed in three operating modes: single scan mode, continuous scan mode, and group scan mode. In single scan mode, one or more specified channels are scanned once. In continuous scan mode, one or more specified channels are scanned repeatedly until the ADST bit in ADCSR register is cleared to 0 from 1 by software. In group scan mode, the selected channels of group A and the selected channels of group B are scanned once after starting to be scanned according to the respective synchronous triggers (MTU, ELC, GPT, TMR, TPU).
In single scan mode and continuous scan mode, A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n. In group scan mode, A/D conversion is performed for ANn channels of group A selected by the ADANSA0 and ADANSA1 registers first, and then performed for ANn channels of group B selected by the ADANSB0 and ADANSB1 registers, respectively, starting from the channel with the smallest number n.
When self-diagnosis is selected, it is executed once at the beginning of each scan and one of the three voltages internally generated in the 12-bit A/D converter is converted.
The temperature sensor output and internal reference voltage can be selected at the same time as the analog input of channels and A/D conversion of the analog input of channels, the temperature sensor output, and the internal reference voltage is performed in that order.
When the extended analog input is selected, A/D conversion should be performed in single scan mode or continuous scan mode.
Double trigger mode is to be used with single scan mode or group scan mode. With double trigger mode being enabled, A/D conversion data of a channel selected by the DBLANS[4:0] bits in ADCSR register is duplicated only if the conversion is started by any of the synchronous triggers (MTU, ELC, GPT, TMR, TPU) selected by the TRSA[5:0] bits in ADSTRGR register.
When any of AN000 to AN002 channels is set as a channel-dedicated sample-and-hold circuit by the SHANS[2:0] bits in ADSHCR register, the target analog input specified is sampled and held before the first A/D conversion of each scan.

57.3.2 Single Scan Mode

57.3.2.1 Basic Operation (Without Channel-Dedicated Sample-and-Hold Circuits)

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as below.

(1) When the ADST bit in ADCSR register is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (MTU, ELC, GPT, TMR, TPU), or input of an asynchronous trigger. A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(2) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).
(3) When A/D conversion of all the selected channels is completed, an S12ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion is enabled).
(4) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a waiting state.

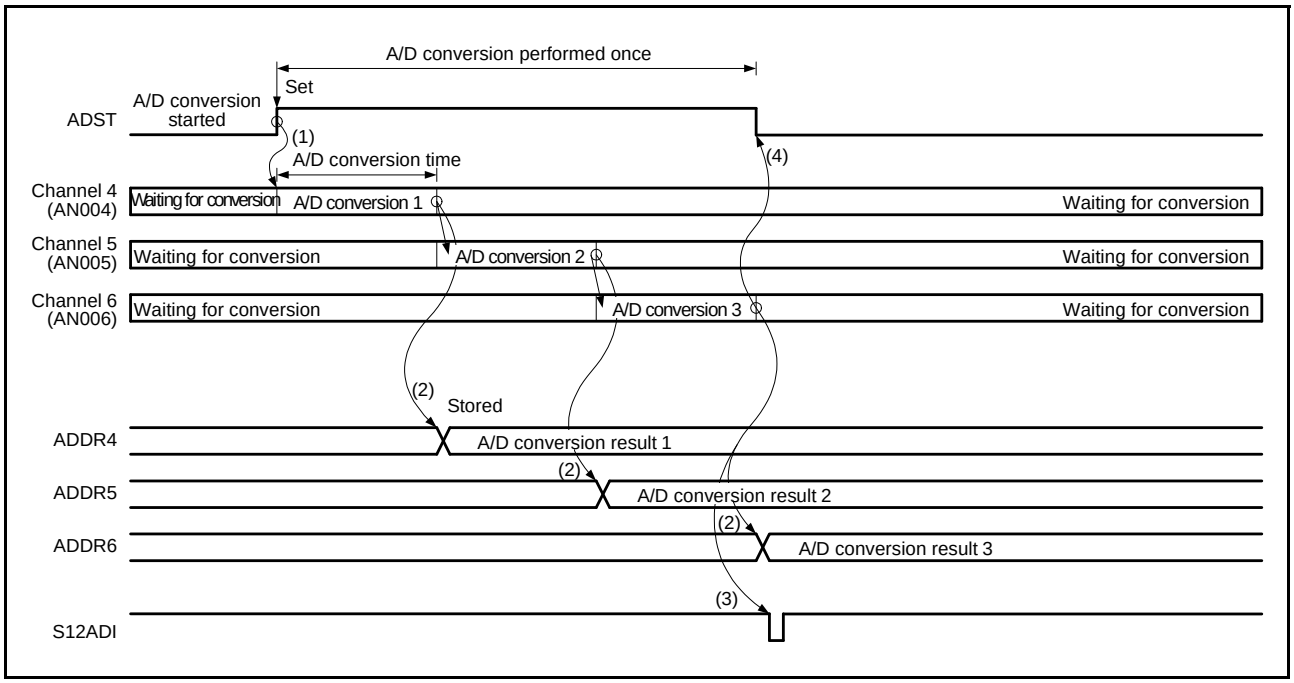


Figure 57.4 Example of Operation in Single Scan Mode (Basic Operation: AN004 to AN006 Selected)

57.3.2.2 Basic Operation (With Channel-Dedicated Sample-and-Hold Circuits, Continuous Sampling Disabled)

When a channel-dedicated sample-and-hold circuit is used while continuous sampling is disabled, sample-and-hold operations are performed first, and this is followed by A/D conversion once of the analog inputs on all selected channels. The ADSHCR.SHANS[2:0] bits are used to select the channels for which the channel-dedicated sample-and-hold circuits are to be used.

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (MTU, ELC, GPT, TMR, TPU), or input of an asynchronous trigger.
(2) After sample-and-hold operation, A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, an S12ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion is enabled).
(5) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a waiting state.

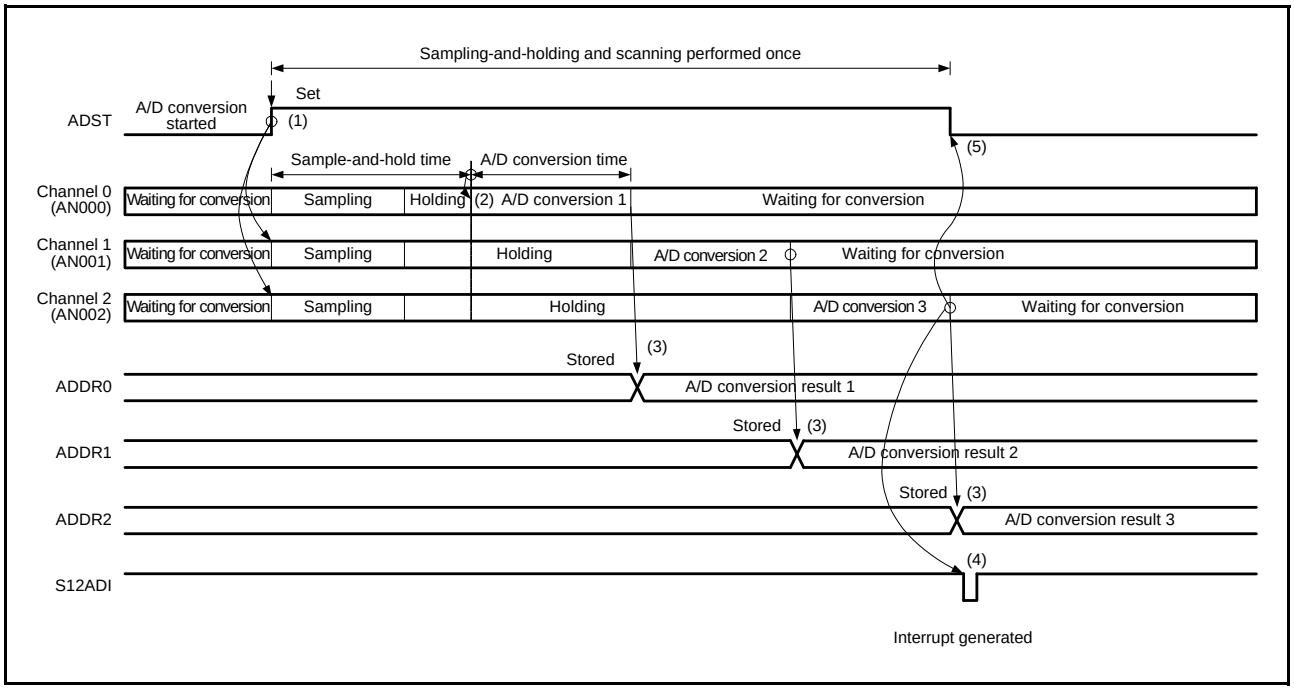


Figure 57.5 Example of Operation in Single Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used: AN000 to AN002 Selected; Continuous Sampling Disabled)

57.3.2.3 Basic Operation (With Channel-Dedicated Sample-and-Hold Circuits, Continuous Sampling Enabled)

When a channel-dedicated sample-and-hold circuit is used while continuous sampling is enabled, sample-and-hold operations are performed first, and this is followed by A/D conversion once of the analog inputs on all selected channels. The ADSHCR.SHANS[2:0] bits are used to select the channels for which the channel-dedicated sample-and-hold circuits are to be used.

(1) When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuits selected by the ADSHCR.SHANS[2:0] bits start continuous sampling.
(2) Analog input holding of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (MTU, GPT, TMR, TPU, ELC), or input of an asynchronous trigger. Set the ADCSR.ADST bit to 1 after at least 400 ns (when the permissible signal source impedance is 1 kΩ) has elapsed since the ADSHMSR.SHMD bit was set to 1.
(3) After the stabilization time of the sample-and-hold circuits has elapsed, A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(4) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy), and the sample-and-hold circuit restarts continuous sampling.
(5) When A/D conversion of all the selected channels is completed, an S12ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled).
(6) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a waiting state. If this is followed by single scanning, set the continuous sampling time for the sample-and-hold circuits to at least 400 ns (when the permissible signal source impedance is 1 kΩ).
(7) When the ADSHMSR.SHMD bit is set to 0, the sample-and-hold circuits are stopped.

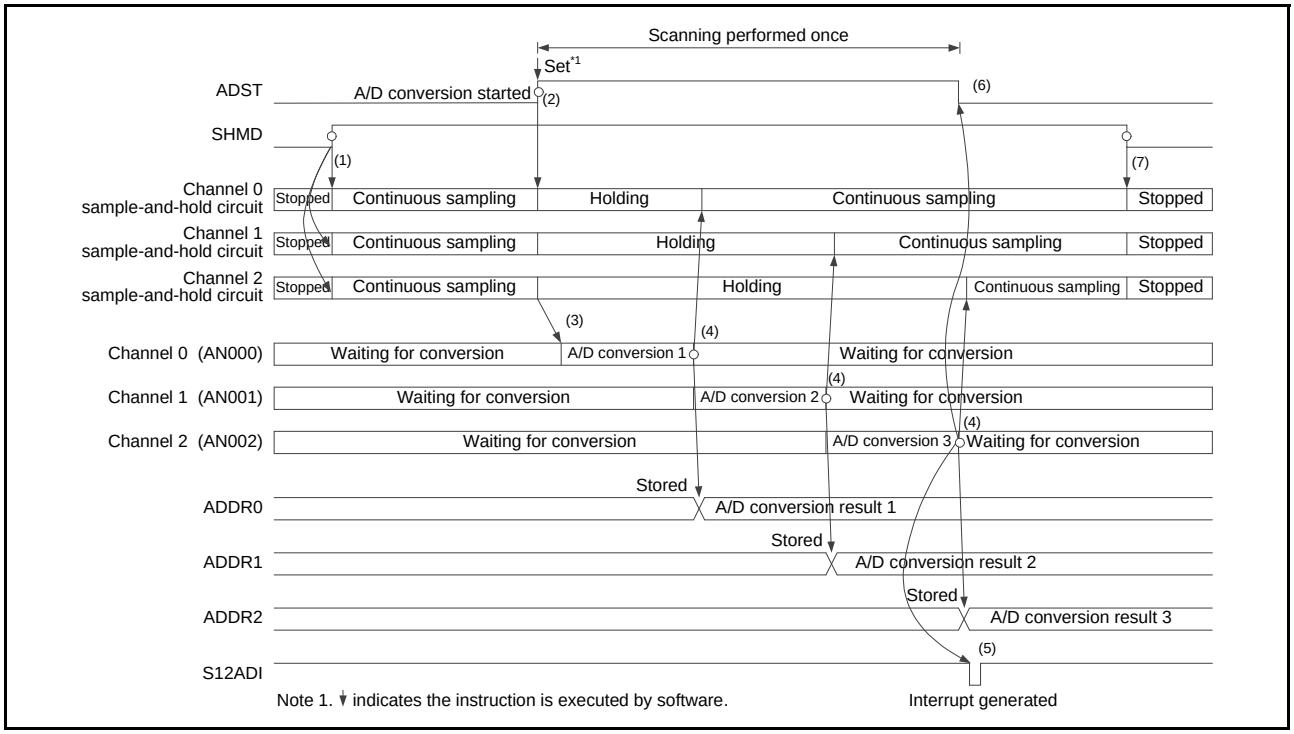


Figure 57.6 Example of Operation in Single Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used: AN000 to AN002 Selected; Continuous Sampling Enabled)

57.3.2.4 Channel Selection and Self-Diagnosis (Without Channel-Dedicated Sample-and-Hold Circuits)

When channels and self-diagnosis are selected, A/D conversion is first performed for the reference voltage VREFH0 (unit 0) or AVCC1 (unit 1) (×0, ×1/2, or ×1) supplied to the 12-bit A/D converter, and then A/D conversion is performed once on the analog input of the selected channels as below.

(1) A/D conversion for self-diagnosis is first started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (MTU, ELC, GPT, TMR, TPU), or input of an asynchronous trigger.
(2) When A/D conversion for self-diagnosis is completed, the result of A/D conversion is stored in the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, an S12ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled).
(5) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a waiting state.

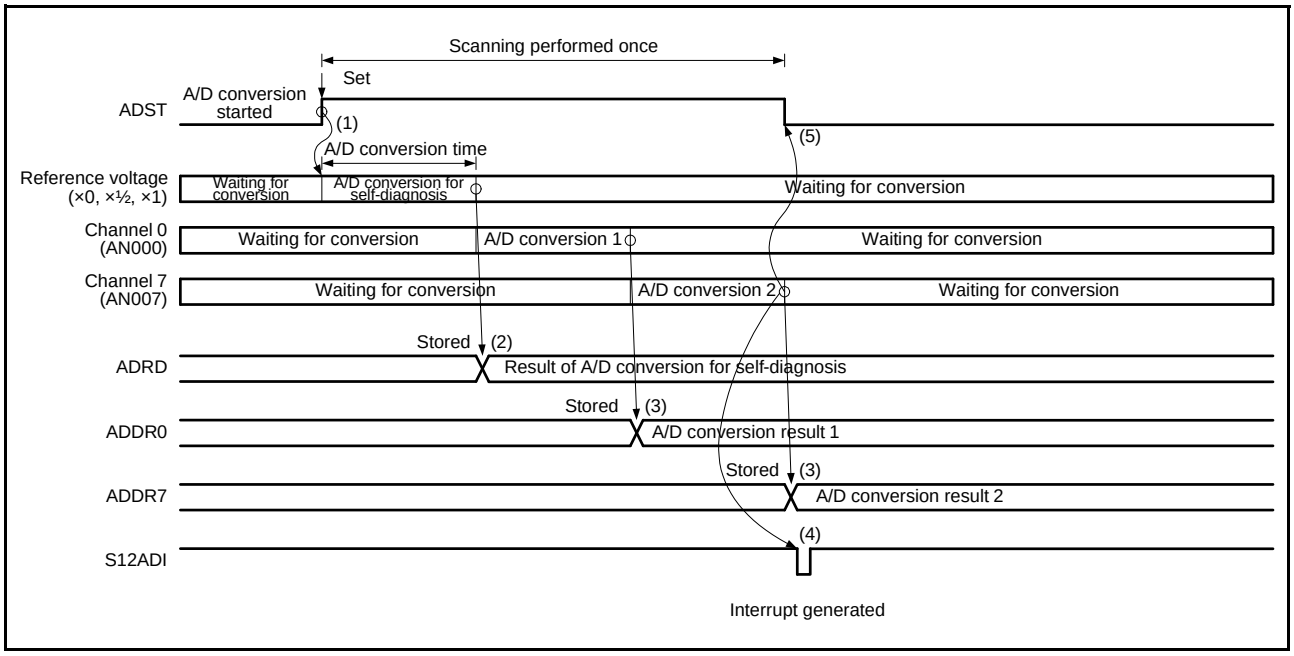


Figure 57.7 Example of Operation in Single Scan Mode (Basic Operation: AN000 and AN007 Selected + Self-Diagnosis)

57.3.2.5 Channel Selection and Self-Diagnosis (With Channel-Dedicated Sample-and-Hold Circuits, Continuous Sampling Disabled)

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used while continuous sampling is disabled, sample-and-hold operations are performed first, and this is followed by A/D conversion of the reference voltage VREFH0 (unit 0) or AVCC1 (unit 1) (×0, ×1/2, or ×1) supplied to the 12-bit A/D converter as below. After that, A/D conversion is performed only once on the analog input of the selected channels.

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (MTU, ELC, GPT, TMR, TPU), or input of an asynchronous trigger.
(2) After sample-and-hold operation, A/D conversion for self-diagnosis is started.
(3) When A/D conversion for self-diagnosis is completed, the result of A/D conversion is stored in the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(4) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).
(5) When A/D conversion of all the selected channels is completed, an S12ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled).
(6) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a waiting state.

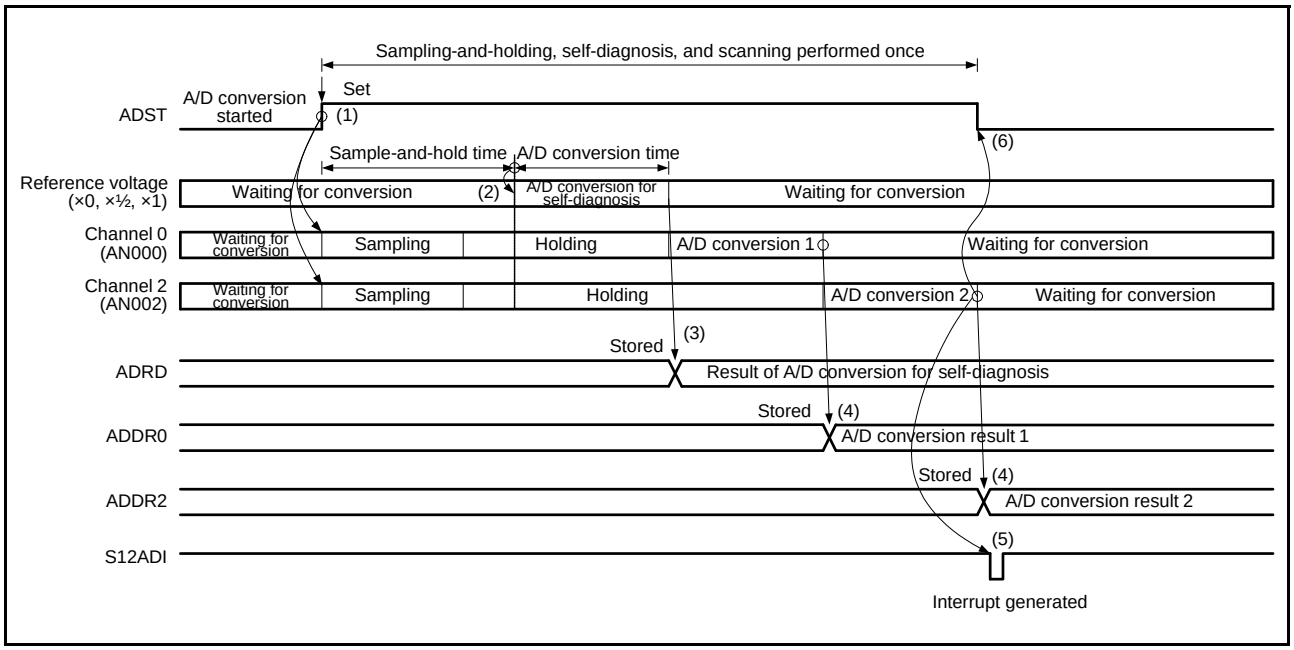


Figure 57.8 Example of Operation in Single Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used: AN000 and AN002 Selected + Self-Diagnosis; Continuous Sampling Disabled)

57.3.2.6 Channel Selection and Self-Diagnosis (With Channel-Dedicated Sample-and-Hold Circuits; Continuous Sampling Enabled)

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used while continuous sampling is enabled, sample-and-hold operations are performed first, and this is followed by A/D conversion of the reference voltage VREFH0 (unit 0) or AVCC1 (unit 1) (×0, ×1/2, or ×1) supplied to the 12-bit A/D converter as below. After that, A/D conversion is performed only once on the analog input of the selected channels.

(1) When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuits selected by the ADSHCR.SHANS[2:0] bits start continuous sampling.

(2) Analog input holding of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (MTU, GPT, TMR, TPU, ELC), or input of an asynchronous trigger. Set the ADCSR.ADST bit to 1 after at least 400 ns (when the permissible signal source impedance is 1 kΩ) has elapsed since the ADSHMSR.SHMD bit was set to 1.

(3) After the stabilization time of the sample-and-hold circuits has elapsed, A/D conversion for self-diagnosis is started.

(4) When A/D conversion for self-diagnosis is completed, the result of A/D conversion is stored in the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.

(5) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy), and the sample-and-hold circuit restarts continuous sampling.

(6) When A/D conversion of all the selected channels is completed, an S12ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled).

(7) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a waiting state. If this is followed by single scanning, set the continuous sampling time for the sample-and-hold circuits to at least 400 ns (when the permissible signal source impedance is 1 kΩ).

(8) When the ADSHMSR.SHMD bit is set to 0, the sample-and-hold circuits are stopped.

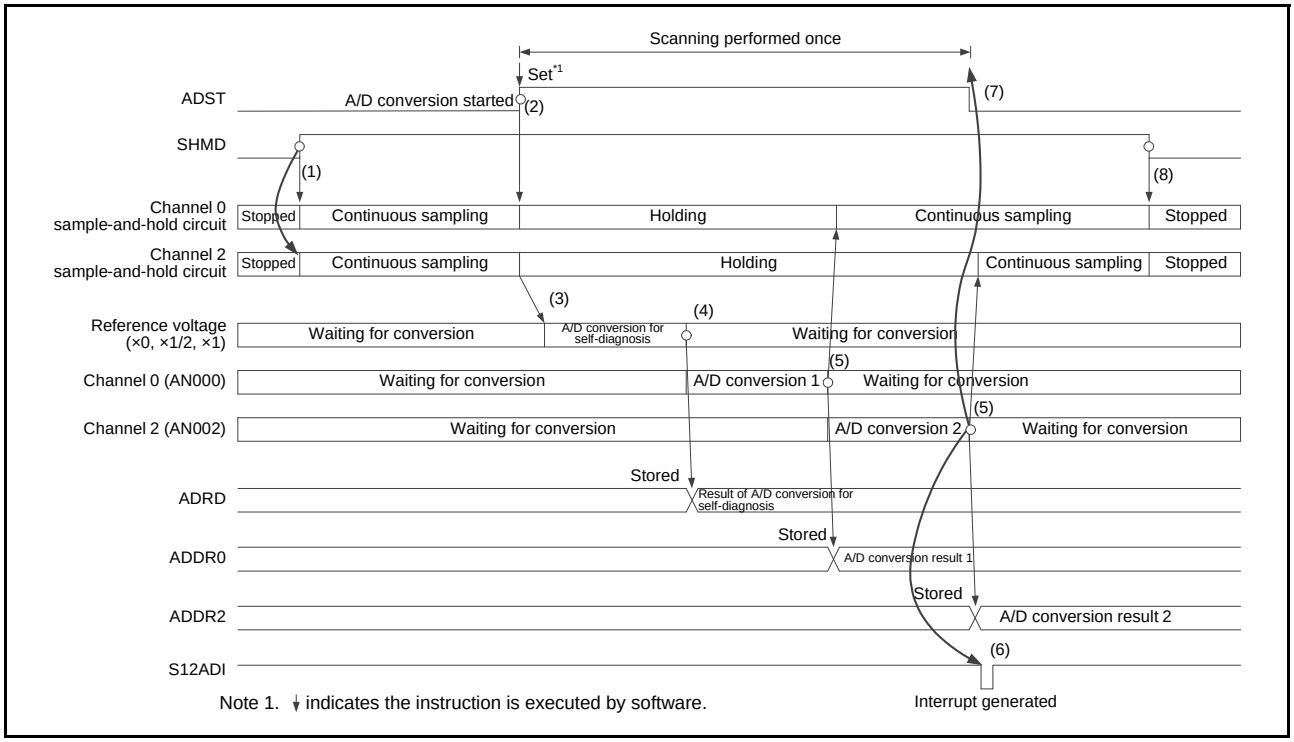


Figure 57.9 Example of Operation in Single Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used: AN000 to AN002 Selected + Self-Diagnosis; Continuous Sampling Enabled)

57.3.2.7 A/D Conversion of Temperature Sensor Output/Internal Reference Voltage

When the channels and temperature sensor output or internal reference voltage are selected at the same time, A/D conversion is first performed on the analog input of the selected channels, and then A/D conversion is performed once on the temperature sensor output or the internal reference voltage as below. When both temperature sensor output and internal reference voltage are selected, A/D conversion of the temperature sensor output and internal reference voltage is performed in that order.
With the channels deselected, selecting only the temperature sensor output or internal reference voltage is also possible.

(1) When software, synchronous trigger (MTU, ELC, GPT, TMR or TPU), or asynchronous trigger sets the ADCSR.ADST bit to 1 (starting A/D conversion), conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the lowest number n.
(2) On completion of A/D conversion on the channel, the result is stored in the corresponding A/D data register (ADDRy), and then A/D conversion of temperature sensor output starts.
(3) On completion of A/D conversion of temperature sensor output, the result is stored in the corresponding A/D temperature sensor data register (ADTRDR), and then A/D conversion of internal reference voltage starts.
(4) On completion of A/D conversion of internal reference voltage, the result is stored in the corresponding A/D internal reference voltage data register (ADOCDR), and if the ADCSR.ADIE bit is set to 1 (enabling S12ADI interrupt generation upon scan conversion completion), an S12ADI interrupt request is generated.
(5) The ADCSR.ADST bit remains 1 (starting A/D conversion) during A/D conversion, and is automatically cleared to 0 upon completion of A/D conversion. Then the 12-bit A/D converter enters a waiting state.

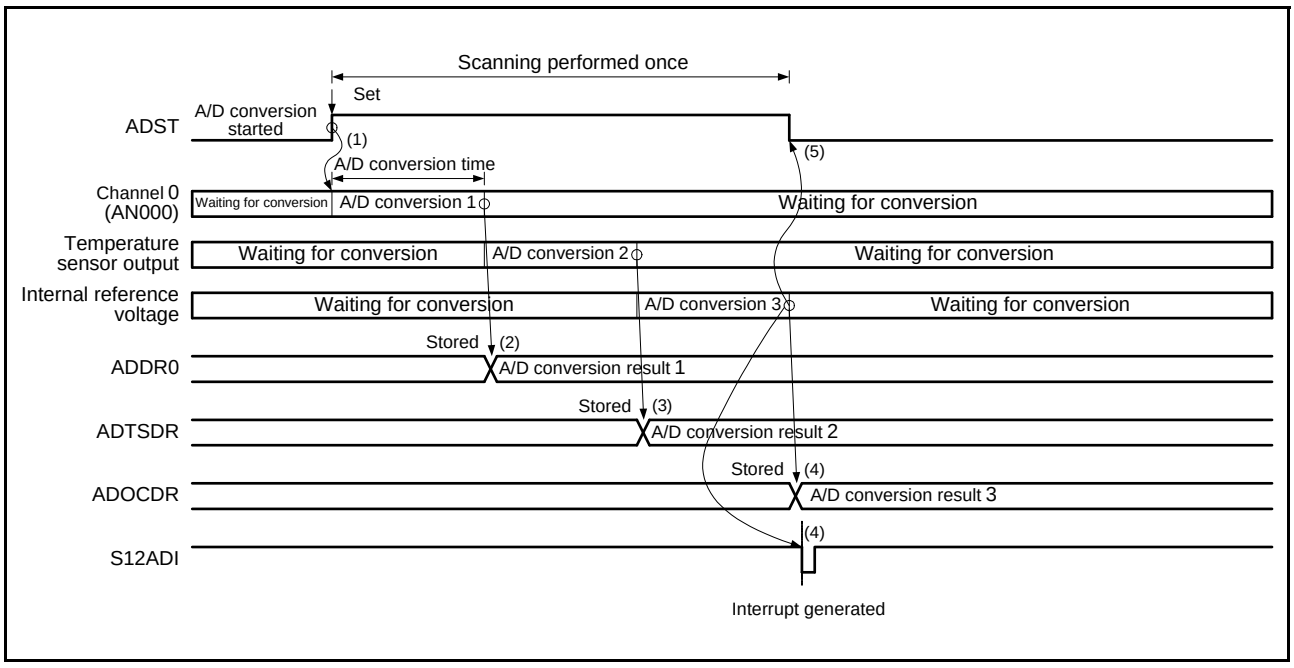


Figure 57.10 Example of Operation in Single Scan Mode (Basic Operation: AN000 and Temperature Sensor Output or Internal Reference Voltage Selected)

57.3.2.8 A/D Conversion in Double Trigger Mode

When double trigger mode is selected in single scan mode, two rounds of single scan operation started by a synchronous trigger (MTU, ELC, GPT, TMR, TPU) are performed as a sequence as shown below.
Self-diagnosis should be deselected, and the temperature sensor output A/D conversion select bits (ADEXICR.TSSA and ADEXICR.TSSB) and the internal reference voltage A/D conversion select bits (ADEXICR.OCSA and ADEXICR.OCSB) should be both set to 0.
Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the DBLANS[4:0] bits in ADCSR register and setting the DBLE bit in ADCSR register to 1. When the DBLE bit in ADCSR register is set to 1, channel selection using the ADANSA0 and ADANSA1 registers is invalid. In double trigger mode, a synchronous trigger (MTU, ELC, GPT, TMR, TPU) should be selected using the TRSA[5:0] bits in ADSTRGR register; the EXTRG bit and TRGE bit in ADCSR register should be set to 0 and 1, respectively. Software trigger should not be used.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a synchronous trigger input (MTU, ELC, GPT, TMR, TPU), A/D conversion is started on the single channel selected by the DBLANS[4:0] bits in ADCSR register.
(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) The ADCSR.ADST bit is automatically cleared to 0 and the 12-bit A/D converter enters a waiting state. Here, an S12ADI interrupt request is not generated irrespective of the ADIE (S12ADI interrupt upon scanning completion enabled) bit setting in ADCSR register.
(4) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by the second trigger input, A/D conversion is started on the single channel selected by the DBLANS[4:0] bits in ADCSR register.
(5) When A/D conversion is completed, the A/D conversion result is stored into the A/D data duplication register (ADDBLDR), which is exclusively used in double trigger mode.
(6) If the ADIE bit in ADCSR register is 1 (S12ADI interrupt upon scanning completion enabled), an S12ADI interrupt request is generated.
(7) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion is completed. Then the 12-bit A/D converter enters a waiting state.

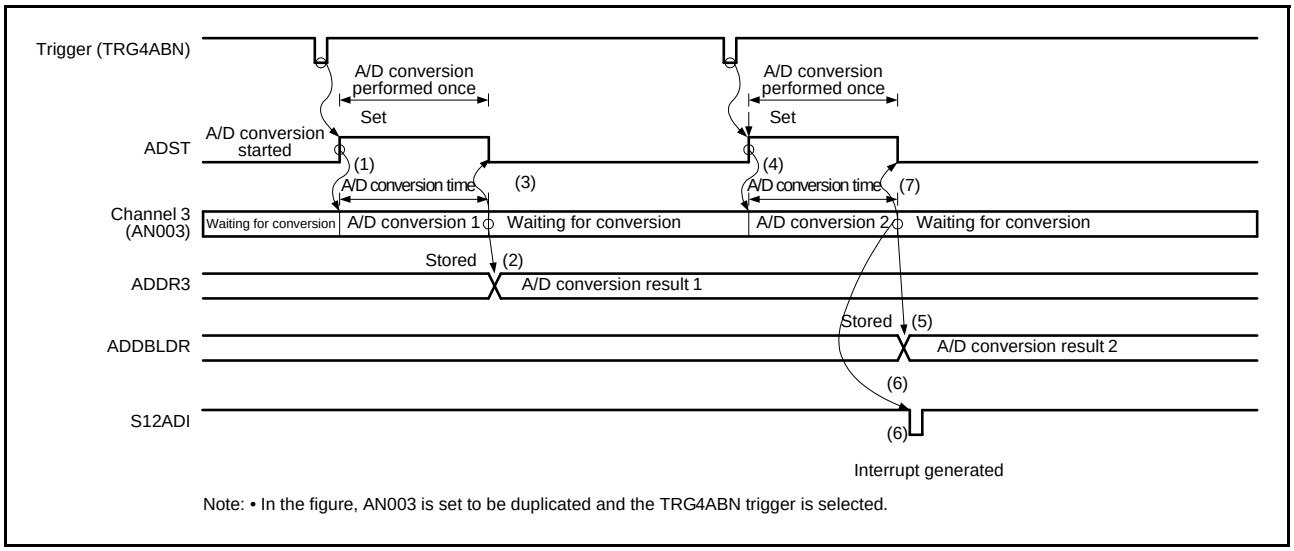


Figure 57.11 Example of Operation in Single Scan Mode (Double Trigger Mode Selected; AN003 Duplicated)

57.3.2.9 Extended Operations When Double Trigger Mode is Selected

When double trigger mode is selected in single scan mode, and a synchronous trigger TRGnAN or TRGnBN (n = 4, 7) is selected as the trigger for the start of A/D conversion (the ADSTRGR.TRSA[5:0] bits are set to 0Bh or 0Fh), or GTADTRAmN or GTADTRBmN (m = 0 to 3) is selected (the ADSTRGR.TRSA[5:0] bits are set to 19h, 1Ah, 1Bh, or 1Ch), the following operations are included with the operations described when double trigger mode is selected.
When A/D conversion is started by a synchronous trigger A (TRGnAN; n = 4, 7, or GTADTRAmN; m = 0 to 3), the results of A/D conversion are stored in A/D data duplication register A (ADDBLDRA). Furthermore, when A/D conversion is started by a synchronous trigger B (TRGnBN; n = 4, 7) or GTADTRBmN (m = 0 to 3), the results of A/D conversion are stored in A/D data duplication register B (ADDBLDRB). With the correspondence between synchronous trigger sources and registers for data storage determined in this way, the target register for storing the results of A/D conversion does not depend on the order of trigger input. Results of A/D conversion are also stored in the A/D data register (ADDRy) and the A/D data duplication register (ADDBLDR) at the same time, and this depends on the order of trigger input.
In extended double trigger mode, if two types of triggers have occurred simultaneously with TRGnAN or TRGnBN (n = 4, 7) or GTADTRAmN or GTADTRBmN (m = 0 to 3) selected, results are not sorted by the trigger sources and are stored in A/D data duplication register B (ADDBLDRB).
Note that if a new trigger source is input during A/D conversion caused by another trigger source, the former new trigger source is ignored and sorting is performed by the latter trigger source.

The extended operations in double trigger mode when the first trigger is TRG4AN with the synchronous trigger TRG4AN or TRG4BN selected as the trigger for the start of A/D conversion are performed as follows.

(1) A/D conversion for the single channel selected by the ADCSR.DBLANS[4:0] bits starts when the TRG4AN input sets the ADCSR.ADST bit to 1 (starting A/D conversion).
(2) The result is stored in A/D data-duplication register A (ADDBLDRA) and in the corresponding A/D data register (ADDRy) on completion of the A/D conversion for the channel.
(3) The ADCSR.ADST bit is automatically cleared and the 12-bit A/D converter enters the waiting state. An S12ADI interrupt is not output at this time, regardless of the setting of the ADCSR.ADIE bit (i.e. whether or not this is set to enable S12ADI interrupts in response to scan completion).
(4) When the TRG4BN input sets the ADCSR.ADST bit to 1 (starting A/D conversion), conversion for the single channel selected by the ADCSR.DBLANS[4:0] bits starts.
(5) The result is stored in A/D data-duplication register B (ADDBLDRB) and in the A/D data-duplication register (ADDBLDR) on completion of A/D conversion.
(6) An S12ADI interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled).
(7) The ADCSR.ADST bit retains the value 1 (starting A/D conversion) during A/D conversion and is automatically cleared on completion of conversion, after which the 12-bit A/D converter enters the waiting state.

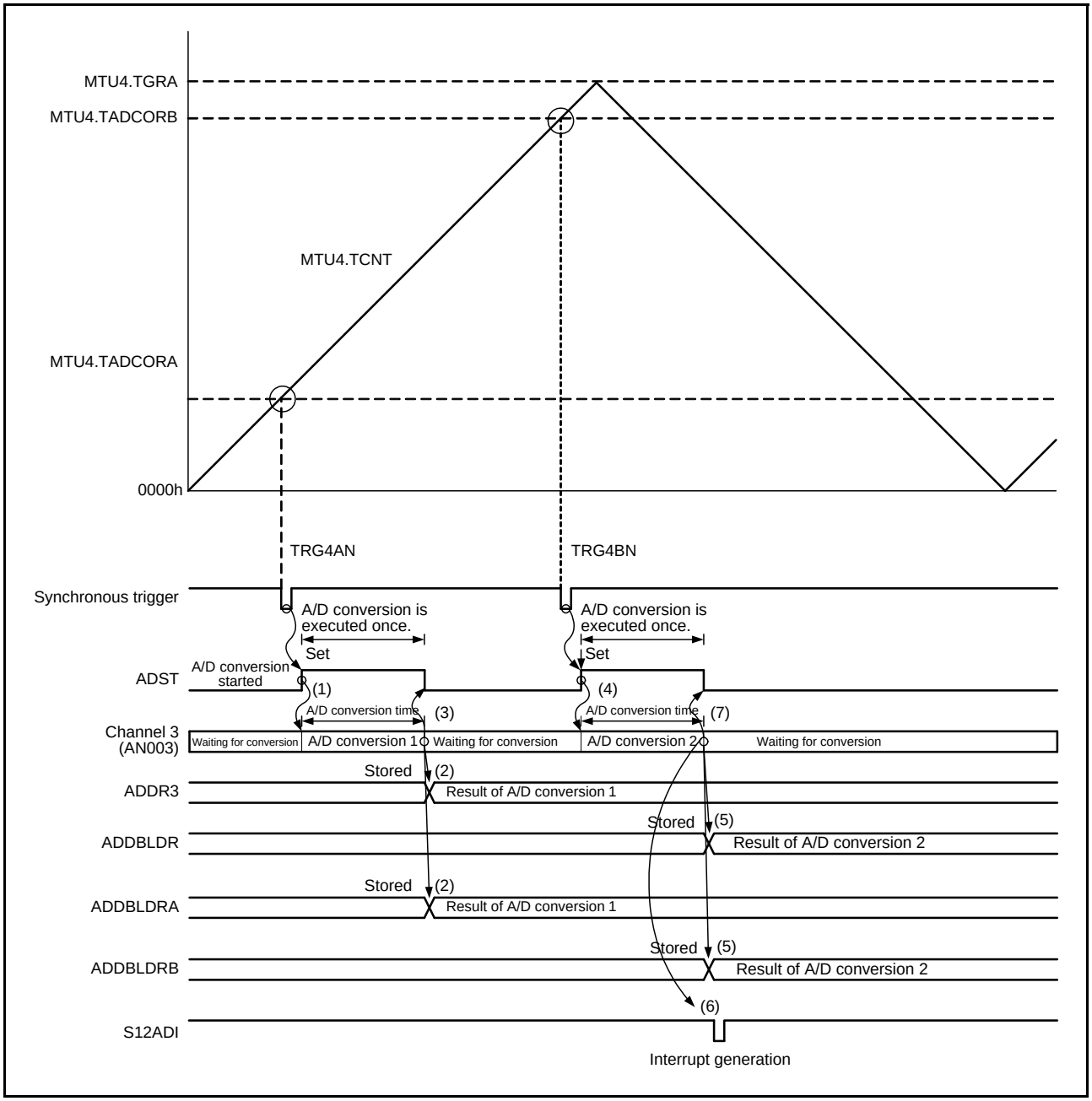


Figure 57.12 Example of Extended Operation in Double Trigger Mode (1)
(Duplication Selected for AN003, TRG4AN or TRG4BN Selected, First Trigger is TRG4AN)

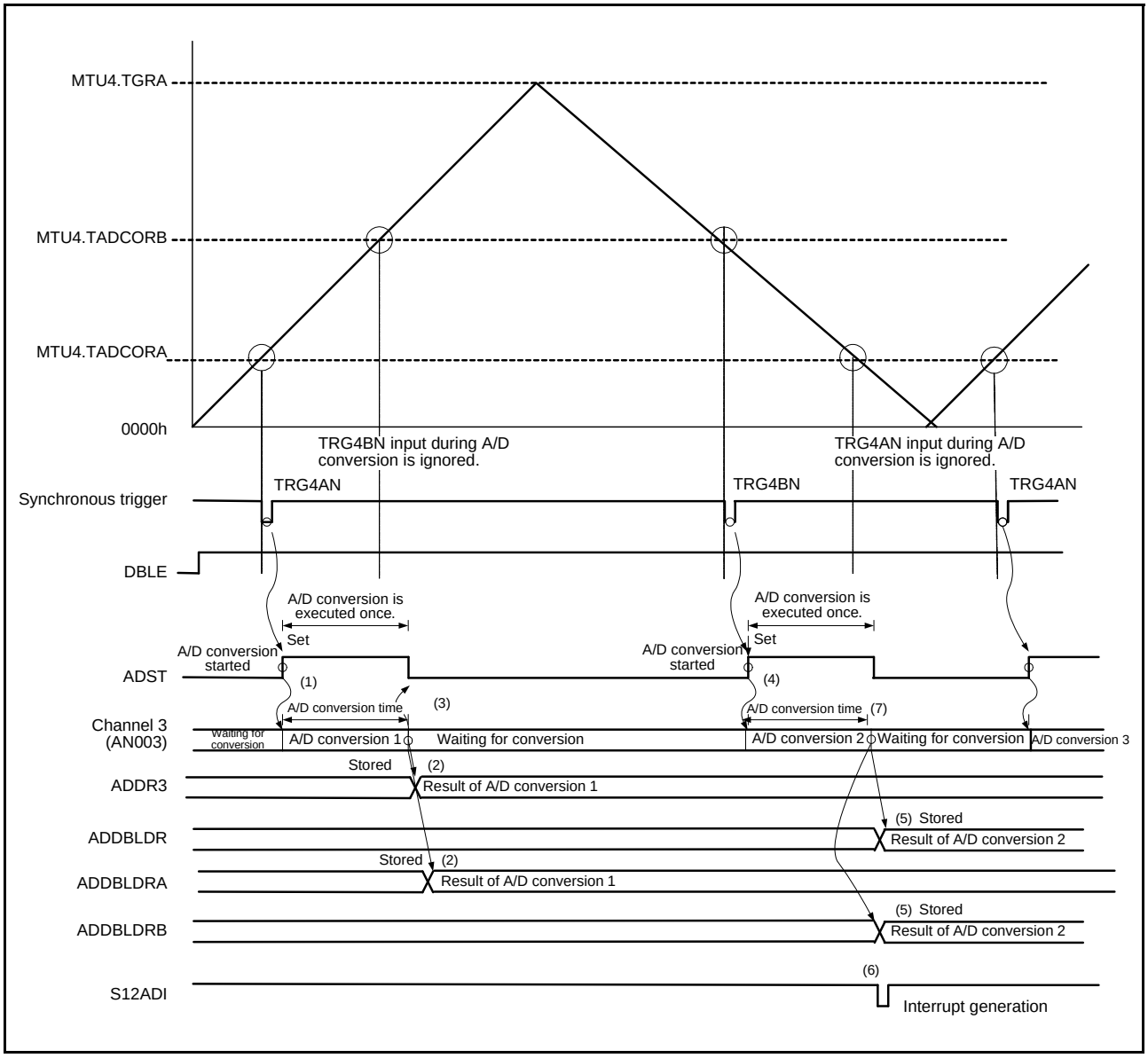


Figure 57.13 Example of Extended Operation in Double Trigger Mode (2)
(Duplication Selected for AN003, TRG4AN and TRG4BN Selected, First Trigger is TRG4AN)

The following describes the extended operations in double trigger mode in response to TRG4BN as the first trigger, again in the case where TRG4AN or TRG4BN is selected as the synchronous trigger for the start of A/D conversion.

(1) When the TRG4BN input sets the ADCSR.ADST bit to 1 (starting A/D conversion), conversion for the single channel selected by the ADCSR.DBLANS[4:0] bits starts.
(2) On completion of A/D conversion, the result is stored in A/D data duplication register B (ADDBLDRB) and in the corresponding A/D data register (ADDRy).
(3) The ADCSR.ADST bit is automatically cleared and the 12-bit A/D converter enters the waiting state. An S12ADI interrupt is not generated at this time, regardless of the setting of the ADCSR.ADIE bit (i.e. whether or not this is set to enable S12ADI interrupts in response to scan completion).
(4) When the TRG4AN input sets the ADCSR.ADST bit to 1 (starting A/D conversion), conversion for the single channel selected by the ADCSR.DBLANS[4:0] bits starts.
(5) On completion of A/D conversion, the result is stored in A/D data duplication register A (ADDBLDRA) and in the A/ D data duplication register (ADDBLDR).
(6) An S12ADI interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion is enabled).
(7) The ADCSR.ADST bit retains the value 1 (starting A/D conversion) during A/D conversion and is cleared on completion of conversion, after which the 12-bit A/D converter enters the waiting state.

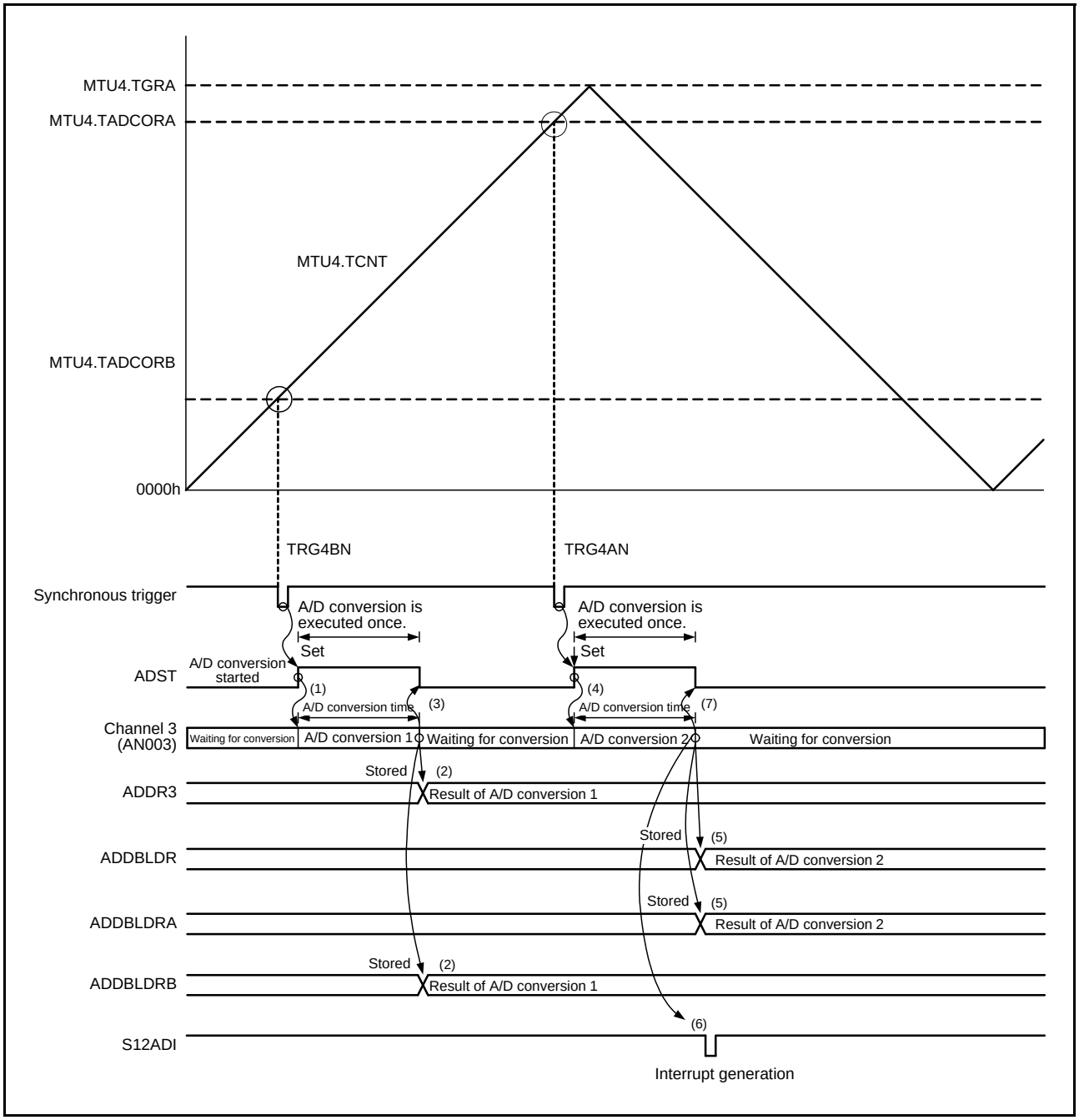


Figure 57.14 Example of Extended Operation in Double Trigger Mode (1)
(Duplication Selected for AN003, TRG4AN and TRG4BN Selected, First Trigger is TRG4BN)

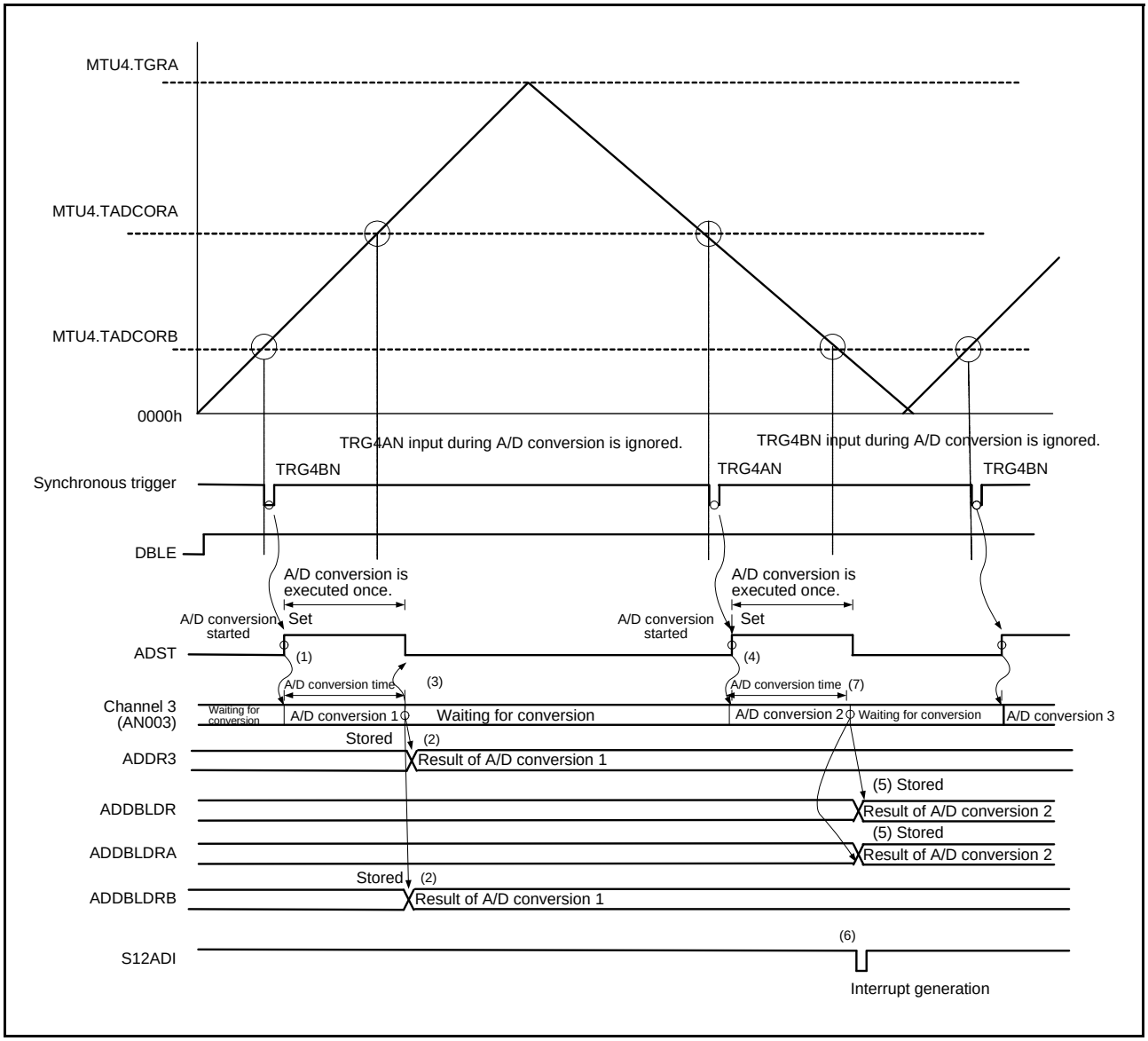


Figure 57.15 Example of Extended Operation in Double Trigger Mode (2)
(Duplication Selected for AN003, TRG4AN and TRG4BN Selected, First Trigger is TRG4BN)

The followings are the extended operations in double trigger mode performed when two types of trigger sources simultaneously occur with the synchronous trigger GTADTRA0N or GTADTRB0N selected as an A/D conversion start trigger.

(1) The first A/D conversion on the single channel selected by the ADCSR.DBLANS[4:0] bits starts when the first simultaneous input of the two trigger sources GTADTRA0N and GTADTRB0N sets the ADCSR.ADST bit to 1 (starting A/D conversion).
(2) The result is stored in A/D data duplication register B (ADDBLDRB) and in the corresponding A/D data register (ADDRy) on completion of the A/D conversion on the channel.
(3) The ADCSR.ADST bit is automatically cleared and the 12-bit A/D converter enters the waiting state. An S12ADI interrupt is not generated at this time, regardless of the setting of the ADCSR.ADIE bit (i.e. whether or not this is set to enable S12ADI interrupts in response to scan completion).
(4) The second A/D conversion on the single channel selected by the ADCSR.DBLANS[4:0] bits starts when the second simultaneous input of the two trigger sources GTADTRA0N and GTADTRB0N sets the ADCSR.ADST bit to 1 (starting A/D conversion).
(5) The result is stored in A/D data duplication register B (ADDBLDRB) and in A/D data duplication register (ADDBLDR) on completion of the A/D conversion.
(6) An S12ADI interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (enabling S12ADI interrupt).
(7) The ADCSR.ADST bit retains the value 1 (starting A/D conversion) during A/D conversion and is cleared on completion of conversion, after which the 12-bit A/D converter enters the waiting state.

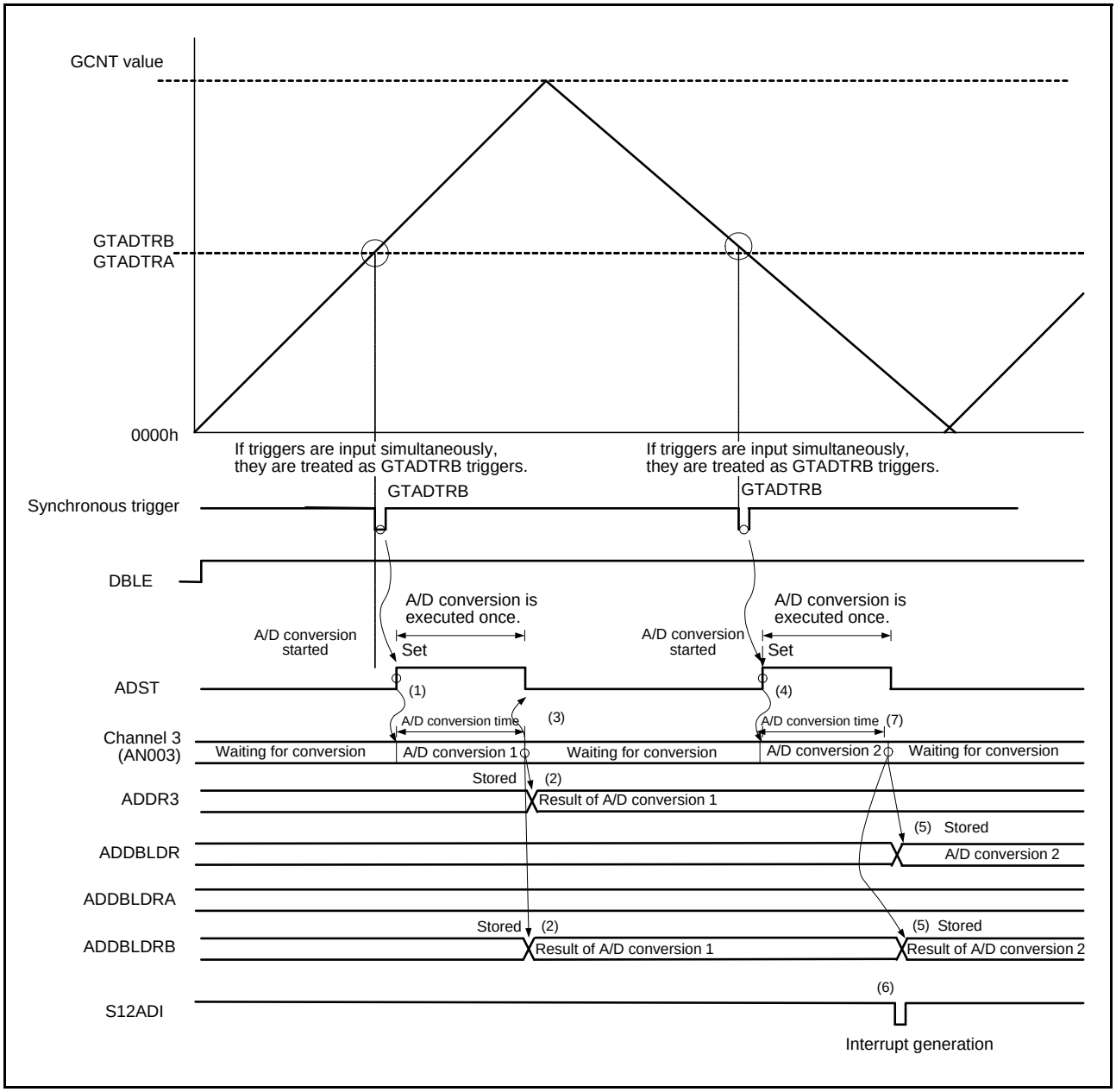


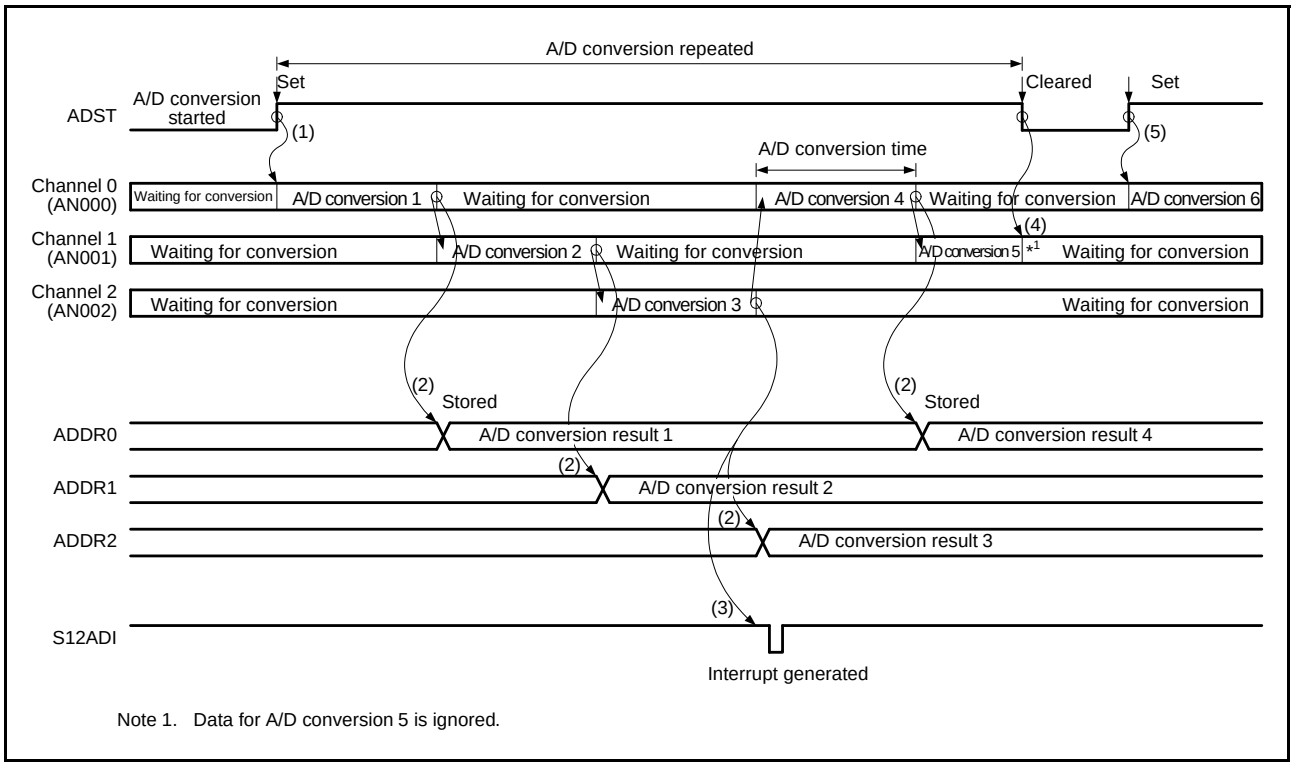
Figure 57.16 Example of Extended Operation in Double Trigger Mode (Duplication Selected for AN003, GTADTRA0N and GTADTRB0N Selected, Two Trigger Sources Simultaneously Occurred)

57.3.3 Continuous Scan Mode

57.3.3.1 Basic Operation (Without Channel-Dedicated Sample-and-Hold Circuits)

In basic operation of continuous scan mode, A/D conversion is performed repeatedly on the analog input of the specified channels as below.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (MTU, ELC, GPT, TMR, TPU), or an asynchronous trigger input, A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) When A/D conversion of all the selected channels is completed, an S12ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled).
The 12-bit A/D converter sequentially starts A/D conversion for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(4) The ADCSR.ADST bit is not automatically cleared and steps 2 and 3 are repeated as long as the bit remains 1 (A/ D conversion start). When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a waiting state.
(5) When the ADCSR.ADST bit is later set to 1 (A/D conversion start), A/D conversion is started again for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.



Note 1. Data for A/D conversion 5 is ignored.

Figure 57.17 Example of Operation in Continuous Scan Mode (Basic Operation: AN000 to AN002 Selected)

57.3.3.2 Basic Operation (With Channel-Dedicated Sample-and-Hold Circuits, Continuous Sampling Disabled)

When a channel-dedicated sample-and-hold circuit is used while continuous sampling is disabled, sample-and-hold operations are performed first, after which the analog inputs on all selected channels are A/D converted in accord with the description below. The channels for which the channel-dedicated sample-and-hold circuits are to be used can be selected by the ADSHCR.SHANS[2:0] bits.

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (MTU, ELC, GPT, TMR, TPU), or input of an asynchronous trigger.
(2) After sample-and-hold operation, A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, an S12ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion is enabled). At the same time, analog input sampling is started for all the channels for which the channel-dedicated sample-and-hold circuits are to be used.
(5) The ADCSR.ADST bit is not automatically cleared and steps 2 to 4 are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a waiting state.
(6) When the ADCSR.ADST bit is then set to 1 (A/D conversion start), analog input sampling is started again for all the channels for which the channel-dedicated sample-and-hold circuits are to be used.

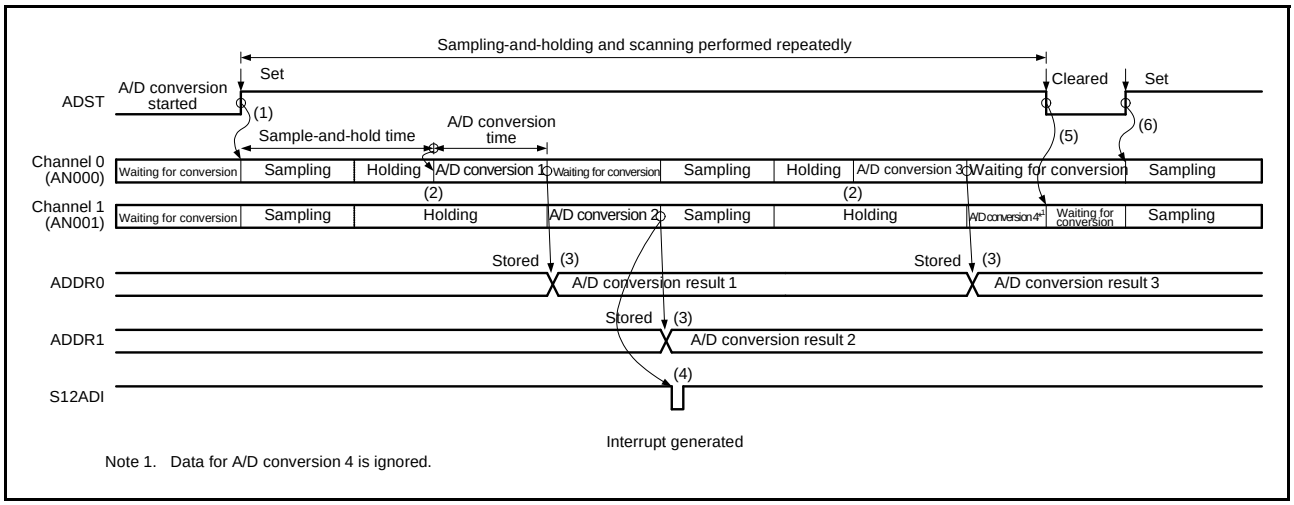


Figure 57.18 Example of Operation in Continuous Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used: AN000 and AN001 Selected; Continuous Sampling Disabled)

57.3.3.3 Basic Operation (With Channel-Dedicated Sample-and-Hold Circuits, Continuous Sampling Enabled)

When a channel-dedicated sample-and-hold circuit is used while continuous sampling is enabled, sample-and-hold operations are performed first, after which the analog inputs on all selected channels are A/D converted in accord with the description below. The channels for which the channel-dedicated sample-and-hold circuits are to be used can be selected by the ADSHCR.SHANS[2:0] bits.

(1) When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuits selected by the ADSHCR.SHANS[2:0] bits start continuous sampling.
(2) Analog input holding of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (MTU, GPT, TMR, TPU, ELC), or input of an asynchronous trigger. Set the ADCSR.ADST bit to 1 after at least 400 ns (when the permissible signal source impedance is 1 kΩ) has elapsed since the ADSHMSR.SHMD bit was set to 1.
(3) After the stabilization time of the sample-and-hold circuits has elapsed, A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(4) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy), and the sample-and-hold circuit restarts continuous sampling.
(5) When A/D conversion of all the selected channels is completed, an S12ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled). Also, analog input holding of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started.
(6) The ADCSR.ADST bit is not automatically cleared and steps 3 to 5 are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a waiting state.
(7) When the ADSHMSR.SHMD bit is set to 0, the sample-and-hold circuits are stopped.
(8) When the ADSHMSR.SHMD bit is then set to 1, the sample-and-hold circuits selected by the ADSHCR.SHANS[2:0] bits start continuous sampling.
(9) When the ADCSR.ADST bit is then set to 1 (A/D conversion start), analog input holding of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started.

Note: If continuous scanning is performed when only the channels with the sample-and-hold circuits are selected, time for continuous sampling cannot be secured in the second and subsequent continuous scanning. When continuous sampling by the channel-dedicated sample-and-hold circuits is enabled for continuous scanning, select one or more channels among AN003 to AN007 and set the continuous sampling time for the sample-and-hold circuits to at least 400 ns (when the permissible signal source impedance is 1 kΩ).

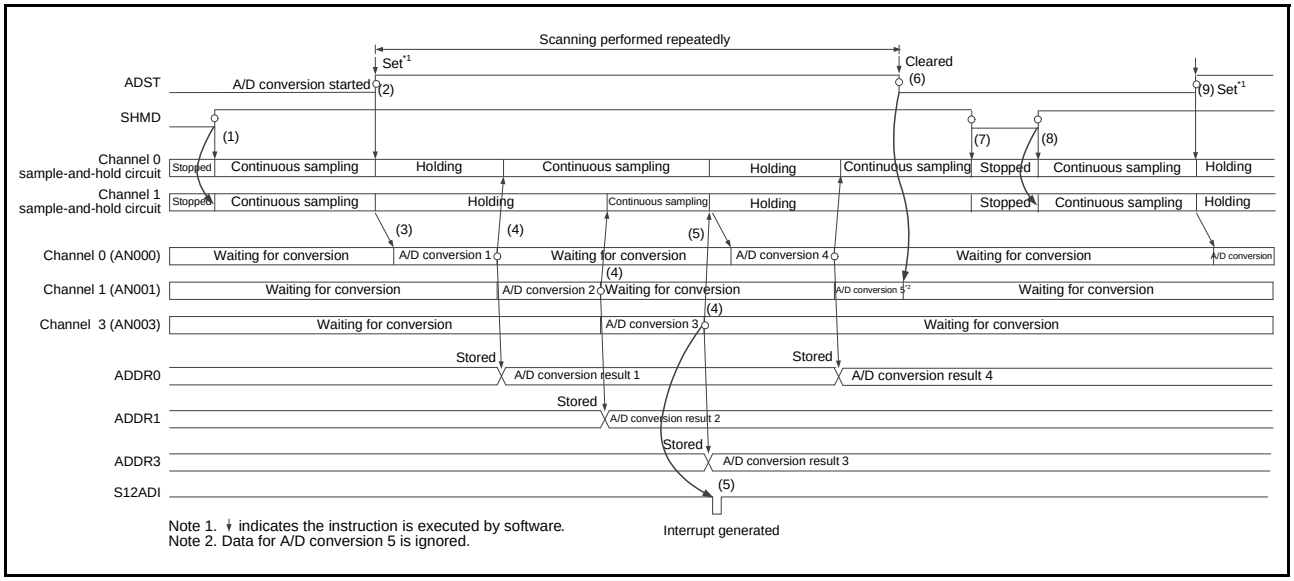


Figure 57.19 Example of Operation in Continuous Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used: AN000, AN001, and AN003 Selected; Continuous Sampling Enabled)

57.3.3.4 Channel Selection and Self-Diagnosis (Without Channel-Dedicated Sample-and-Hold Circuits)

When channels and self-diagnosis are selected at the same time, A/D conversion is first performed for the reference voltage VREFH0 (unit 0) or AVCC1 (unit 1) (×0, ×1/2, or ×1) supplied to the 12-bit A/D converter, after which the analog inputs on all selected channels are A/D converted in accord with the description below.

(1) A/D conversion for self-diagnosis is first started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (MTU, ELC, GPT, TMR, TPU), or input of an asynchronous trigger.
(2) When A/D conversion for self-diagnosis is completed, the result of A/D conversion is stored in the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, an S12ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion is enabled). At the same time, the 12-bit A/D converter starts A/D conversion for self-diagnosis and then starts A/D conversion on ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(5) The ADCSR.ADST bit is not automatically cleared and steps 2 to 4 are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a waiting state.
(6) When the ADCSR.ADST bit is later set to 1 (A/D conversion start), the A/D conversion for self-diagnosis is started again.

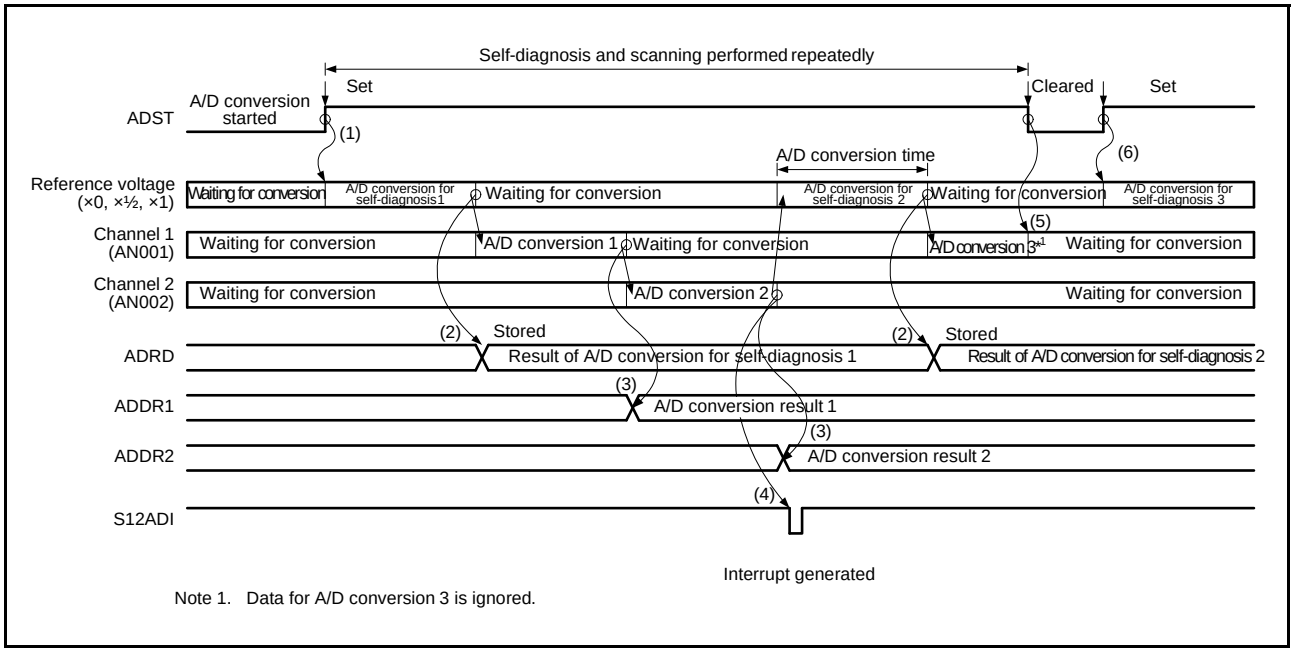


Figure 57.20 Example of Operation in Continuous Scan Mode (Basic Operation; AN001 and AN002 Selected + Self-Diagnosis)

57.3.3.5 Channel Selection and Self-Diagnosis (With Channel-Dedicated Sample-and-Hold Circuits, Continuous Sampling Disabled)

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used while continuous sampling is disabled, sample-and-hold operations are performed first, and this is followed by A/D conversion of the reference voltage VREFH0 (unit 0) or AVCC1 (unit 1) (×0, ×1/2, or ×1) supplied to the 12-bit A/D converter. After that, A/D conversion is performed on the analog input of the selected channels, which sequence is repeated as below.

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (MTU, ELC, GPT, TMR, TPU), or input of an asynchronous trigger.
(2) After sample-and-hold operation, A/D conversion for self-diagnosis is started.
(3) When A/D conversion for self-diagnosis is completed, the result of A/D conversion is stored in the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(4) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).
(5) When A/D conversion of all the selected channels is completed, an S12ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled). At the same time, analog input sampling is started for all the channels for which the channel-dedicated sample-and-hold circuits are to be used.
(6) The ADCSR.ADST bit is not automatically cleared and steps 2 to 5 are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a waiting state.
(7) When the ADCSR.ADST bit is then set to 1 (A/D conversion start), analog input sampling is started again for all the channels for which the channel-dedicated sample-and-hold circuits are be used.

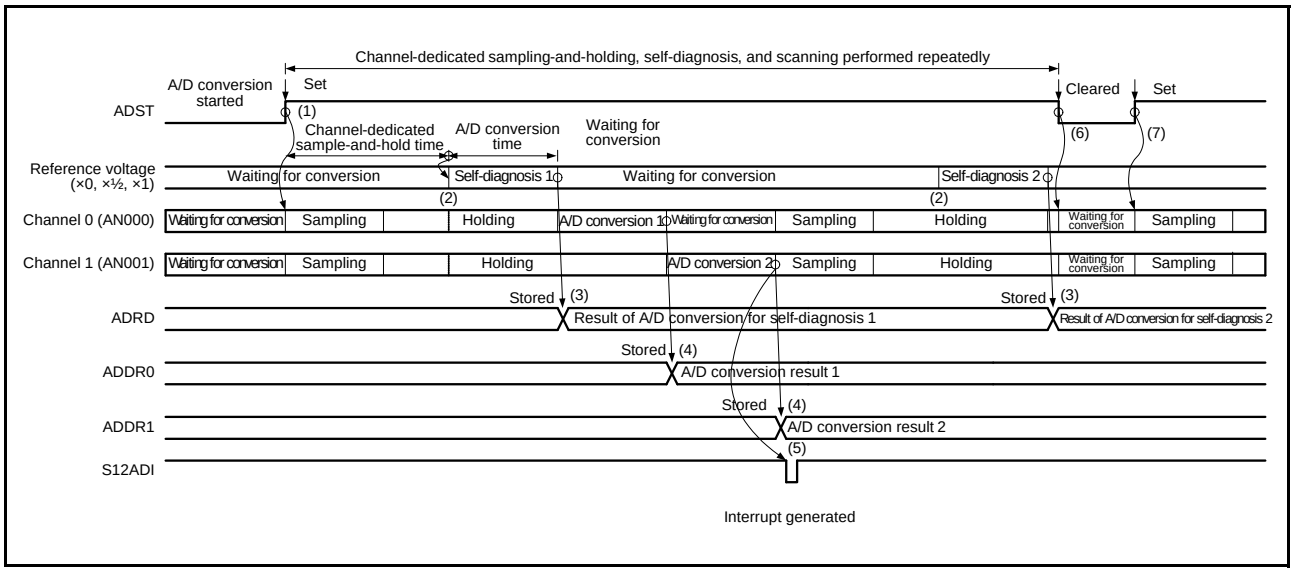


Figure 57.21 Example of Operation in Continuous Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used: AN000 and AN001 Selected + Self-Diagnosis; Continuous Sampling Disabled)

57.3.3.6 Channel Selection and Self-Diagnosis (With Channel-Dedicated Sample-and-Hold Circuits, Continuous Sampling Enabled)

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used while continuous sampling is enabled, sample-and-hold operation is first performed, and this is followed by A/D conversion of the reference voltage VREFH0 (unit 0) or AVCC1 (unit 1) (×0, ×1/2, or ×1) supplied to the 12-bit A/D converter. After that, A/D conversion is performed on the analog input of the selected channels, which sequence is repeated as below.

(1) When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuits selected by the ADSHCR.SHANS[2:0] bits start continuous sampling.
(2) Analog input holding of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (MTU, GPT, TMR, TPU, ELC), or input of an asynchronous trigger. Set the ADCSR.ADST bit to 1 after at least 400 ns (when the permissible signal source impedance is 1 kΩ) has elapsed since the ADSHMSR.SHMD bit was set to 1.
(3) After the stabilization time of the sample-and-hold circuits has elapsed, A/D conversion for self-diagnosis is started.
(4) When A/D conversion for self-diagnosis is completed, the result of A/D conversion is stored in the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
(5) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy), and the sample-and-hold circuit restarts continuous sampling.
(6) When A/D conversion of all the selected channels is completed, an S12ADI interrupt request is generated if the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion enabled). Also, analog input holding of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started.
(7) The ADCSR.ADST bit is not automatically cleared and steps 3 to 6 are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a waiting state.
(8) When the ADSHMSR.SHMD bit is set to 0, the sample-and-hold circuits are stopped.
(9) When the ADSHMSR.SHMD bit is then set to 1, the sample-and-hold circuits selected by the ADSHCR.SHANS[2:0] bits start continuous sampling.
(10) When the ADCSR.ADST bit is then set to 1 (A/D conversion start), analog input holding of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started.

Note: If continuous scanning is performed when only the channels with the sample-and-hold circuits are selected, time for continuous sampling cannot be secured in the second and subsequent continuous scanning. When continuous sampling by the channel-dedicated sample-and-hold circuits is enabled for continuous scanning, select one or more channels among AN003 to AN007 and set the continuous sampling time for the sample-and-hold circuits to at least 400 ns (when the permissible signal source impedance is 1 kΩ).

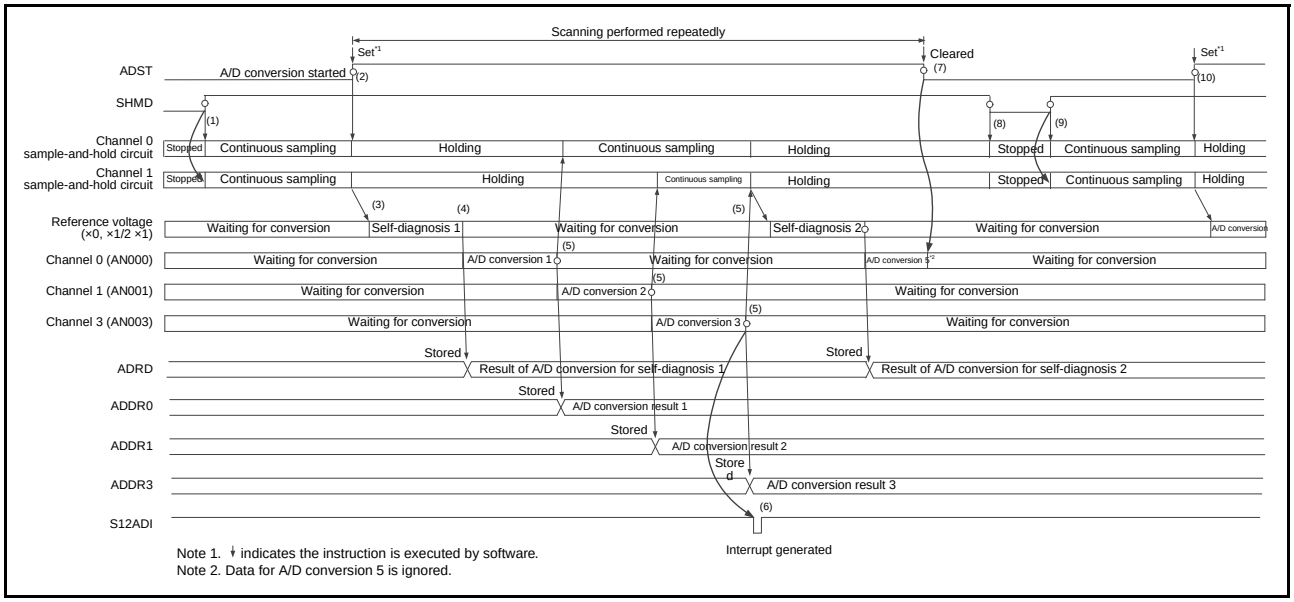


Figure 57.22 Example of Operation in Continuous Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used: AN000, AN001, and AN003 Selected + Self-Diagnosis; Continuous Sampling Enabled)

57.3.3.7 A/D Conversion of Temperature Sensor Output/Internal Reference Voltage

When the channels and temperature sensor output or internal reference voltage are selected at the same time, the analog inputs on all selected channels are A/D converted, after which the temperature sensor output and internal reference voltage are A/D converted in accord with the description below. When both temperature sensor output and internal reference voltage are selected, the temperature sensor output and internal reference voltage are A/D converted in that order.

With the channels deselected, selecting only the temperature sensor output or internal reference voltage is also possible.

(1) When software, synchronous trigger (MTU, ELC, GPT, TMR or TPU), or asynchronous trigger sets the ADCSR.ADST bit to 1 (starting A/D conversion), conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the lowest number n.

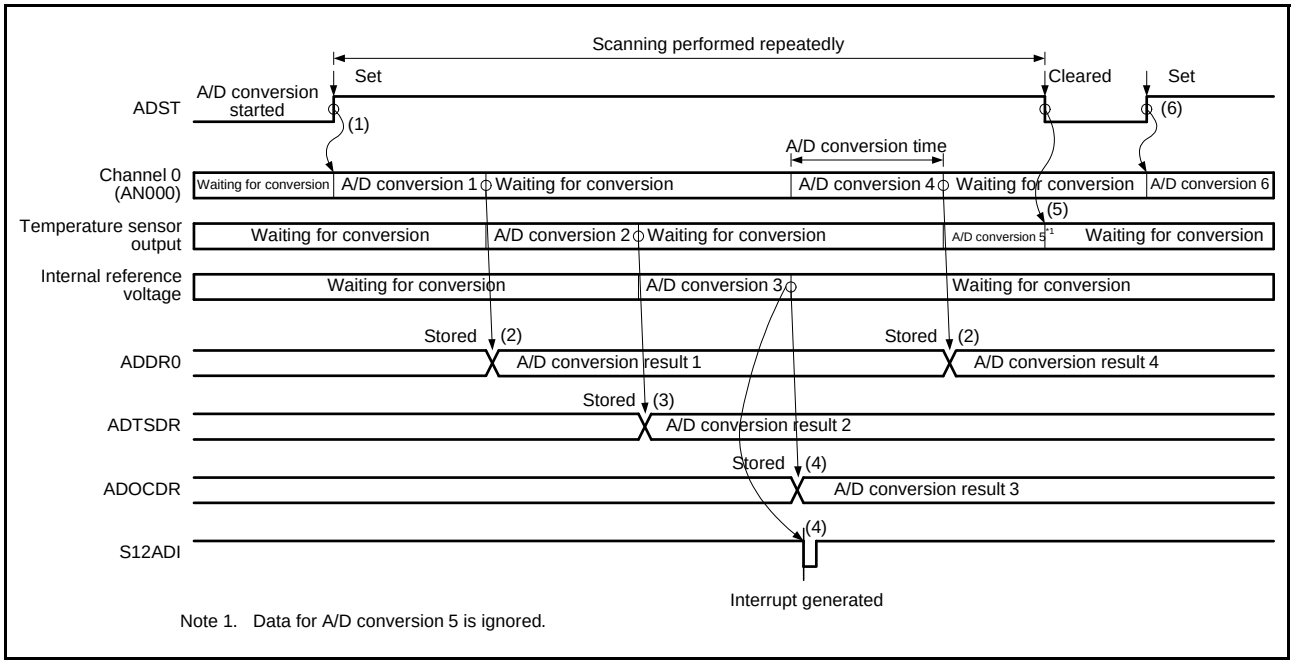
(2) On completion of A/D conversion on the channel, the result is stored in the corresponding A/D data register (ADDRy), and then A/D conversion of temperature sensor output starts.

(3) On completion of A/D conversion of temperature sensor output, the result is stored in the corresponding A/D temperature sensor data register (ADTRDR), and then A/D conversion of internal reference voltage starts.

(4) On completion of A/D conversion of internal reference voltage, the result is stored in the corresponding A/D internal reference voltage data register (ADOCDR), and if the ADCSR.ADIE bit is set to 1 (enabling S12ADI interrupt generation upon scan conversion completion), an S12ADI interrupt request is generated. Furthermore, the 12-bit A/D converter continuously starts A/D conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers in order from the channel with the lowest number n.

(5) The ADCSR.ADST bit is not cleared automatically, and steps 2 to 4 are repeated as long as this bit remains set to 1. When the ADCSR.ADST bit is set to 0 (stopping A/D conversion), A/D conversion stops and the 12-bit A/D converter enters a wait state.

(6) If the ADCSR.ADST bit is later set to 1 (starting A/D conversion), A/D conversion starts again for ANn channels selected by the ADANSA0 and ADANSA1 registers in order from the channel with the lowest number n.



Note 1. Data for A/D conversion 5 is ignored.

Figure 57.23 Example of Operation in Continuous Scan Mode (Basic Operation: AN000 and Temperature Sensor Output or Internal Reference Voltage Selected)

57.3.4 Group Scan Mode

57.3.4.1 Basic Operation

In basic operation of group scan mode, A/D conversion is performed once on the analog inputs of all the specified channels in group A and group B after scanning is started by a synchronous trigger (MTU, ELC, GPT, TMR, TPU) as below. Scan operation of each group is similar to the scan operation in single scan mode.
The synchronous triggers of group A and B can be selected using the TRSA[5:0] and TRSB[5:0] bits in ADSTRGR register, respectively. The different triggers should be used for group A and group B to prevent simultaneous A/D conversion of group A and group B. Software trigger should not be used.
The group A channels to be A/D-converted are selected using the ADANSA0 and ADANSA1 registers and the ADEXICR.TSSA and OCSA bits, while the group B channels to be A/D-converted are selected using the ADANSB0 and ADANSB1 registers and the ADEXICR.TSSB and OCSB bits. Group A and group B cannot use the same channels.
When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed for group A and group B.
The following describes operation in group scan mode using a trigger from the MTU. Specifically, the TRG4AN and TRG4BN triggers from the MTU are assumed to be used to start conversion of group A and group B, respectively.

(1) Scanning of group A is started by the TRG4AN trigger from the MTU.
(2) When group A scanning is completed, an S12ADI interrupt is output if the ADIE bit in ADCSR register is 1 (S12ADI interrupt upon scanning completion is enabled).
(3) Scanning of group B is started by the TRG4BN trigger from the MTU.
(4) When group B scanning is completed, a S12GBADI interrupt is output if the GBADIE bit in ADCSR is 1 (S12GBADI interrupt upon scanning completion is enabled).

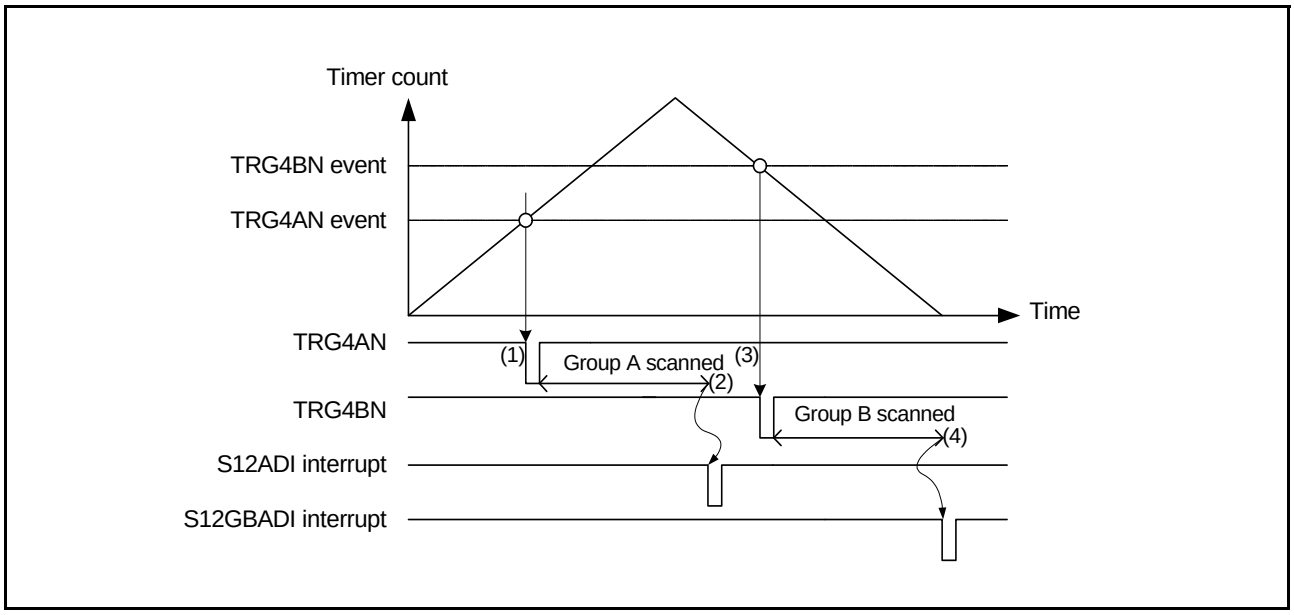


Figure 57.24 Example of Operation in Group Scan Mode (Basic Operation: Triggers from MTU Used)

57.3.4.2 A/D Conversion in Double Trigger Mode

When double trigger mode is selected in group scan mode, two rounds of single scan operation started by a synchronous trigger (MTU, ELC, GPT, TMR, TPU) are performed as a sequence for group A. For group B, single scan operation started by a synchronous trigger (MTU, ELC, GPT, TMR, TPU) is performed once.
In group scan mode, the synchronous triggers of group A and B can be selected using the TRSA[5:0] and TRSB[5:0] bits in ADSTRGR register, respectively. The different triggers should be used for group A and group B to prevent simultaneous A/D conversion of group A and group B. Software trigger, or asynchronous trigger (ADTRGn#) should not be used.
When a synchronous trigger TRGnAN or TRGnBN (n = 4, 7) is selected as the trigger for the start of A/D conversion (the ADSTRGR.TRSA[5:0] bits are set to 0Bh or 0Fh), or GTADTRAmN or GTADTRBmN (m = 0 to 3) is selected (the ADSTRGR.TRSA[5:0] bits are set to 19h, 1Ah, 1Bh, or 1Ch), operation proceeds in extended double trigger mode.
The group A and group B channels to be A/D-converted are selected using the DBLANS[4:0] bits in the ADCSR register and the ADANSB0 and ADANSB1 registers, respectively. The same channels cannot be selected for both groups.
When double trigger mode is selected in group scan mode, the temperature sensor output A/D conversion select bits (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bits (ADEXICR.OCSA) should both be set to 0 (deselected).
When double trigger mode is selected in group scan mode, self-diagnosis cannot be selected.
Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the DBLANS[4:0] bits in ADCSR and setting the DBLE bit in ADCSR to 1.
The following describes operation in group scan mode with double trigger mode using a trigger from the MTU. Specifically, the TRG4ABN and TRG0AN triggers from the MTU are assumed to be used to start conversion of group A and group B, respectively.

(1) Scanning of group B is started by the TRG0AN trigger from the MTU.
(2) When group B scanning is completed, a S12GBADI interrupt is output if the GBADIE bit in ADCSR is 1 (S12GBADI interrupt upon scanning completion is enabled).
(3) The first scanning of group A is started by the first TRG4ABN trigger from the MTU.
(4) When the first scanning of group A is completed, the conversion result is stored into the corresponding A/D data register (ADDRy); an S12ADI interrupt request is not generated irrespective of the ADIE bit setting in ADCSR.
(5) The second scanning of group A is started by the second TRG4ABN trigger from the MTU.
(6) When the second scanning of group A is completed, the conversion result is stored into the ADDBLDR register. An S12ADI interrupt is generated if the ADIE bit is 1 (S12ADI interrupt upon scanning completion is enabled).

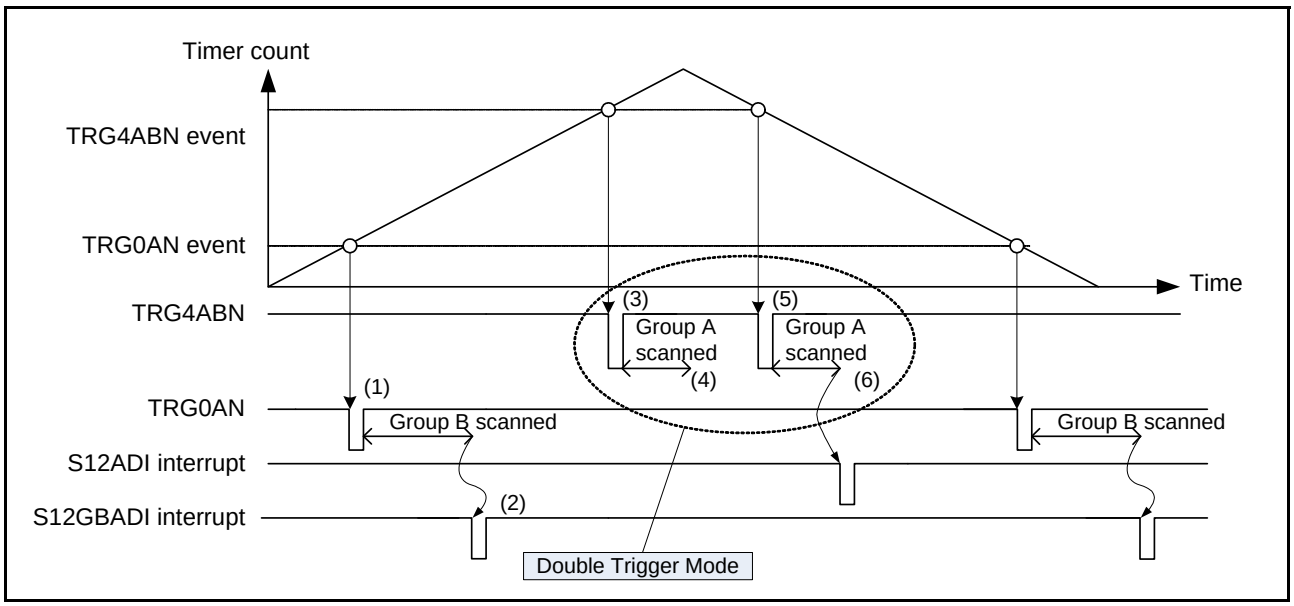


Figure 57.25 Example of Operation in Group Scan Mode with Double Trigger Mode (Basic Operation: Triggers from MTU Used)

57.3.4.3 Operation under Group-A Priority Control

Setting the ADGSPCR.PGS bit to 1 in group scan mode makes operation proceed under group-A priority control. When setting the ADGSPCR.PGS bit to 1, follow the procedure described in Figure 57.26. If the procedure is not followed, A/D conversion operation and stored data are not guaranteed.
In operation in basic group scan mode, input of the trigger for the other group during operation for A/D conversion in group A or group B is ignored. Under group-A priority control, if a group-A trigger is input during A/D conversion for group B, A/D conversion for group B is discontinued and A/D conversion for group A proceeds. If the setting of the ADGSPCR.GBRSCN bit is 0, the converter enters the waiting state on completion of the A/D conversion for group A. If the setting of the ADGSPCR.GBRSCN bit is 1, the converter automatically restarts scanning for group B from the head of the group after completion of the A/D conversion for group A. Table 57.9 summarizes operations in response to the input of a trigger during A/D conversion with the settings of the ADGSPCR.GBRSCN bit.

Scan operations in group A or group B are the same in single scan mode. Furthermore, single scanning continues to proceed if the ADGSPCR.GBRP bit is set to 1 during scanning operations for group B.

For the trigger settings in group scan mode, select a synchronous or asynchronous trigger for group A using the ADSTRGR.TRSA[5:0] bits and select a synchronous trigger different from that of group A for group B using the ADSTRGR.TRSB[5:0] bits. Set the ADSTRGR.TRSB[5:0] bits to 3Fh when setting the ADGSPCR.GBRP bit to 1. Furthermore, as targets for A/D conversion, select channels for group A using the ADANSA0 and ADANSA1 registers and the ADEXICR.TSSA and OCSA bits, and for group B, select channels different from those for group A using the ADANSB0 and ADANSB1 registers and the ADEXICR.TSSB and OCSB bits.

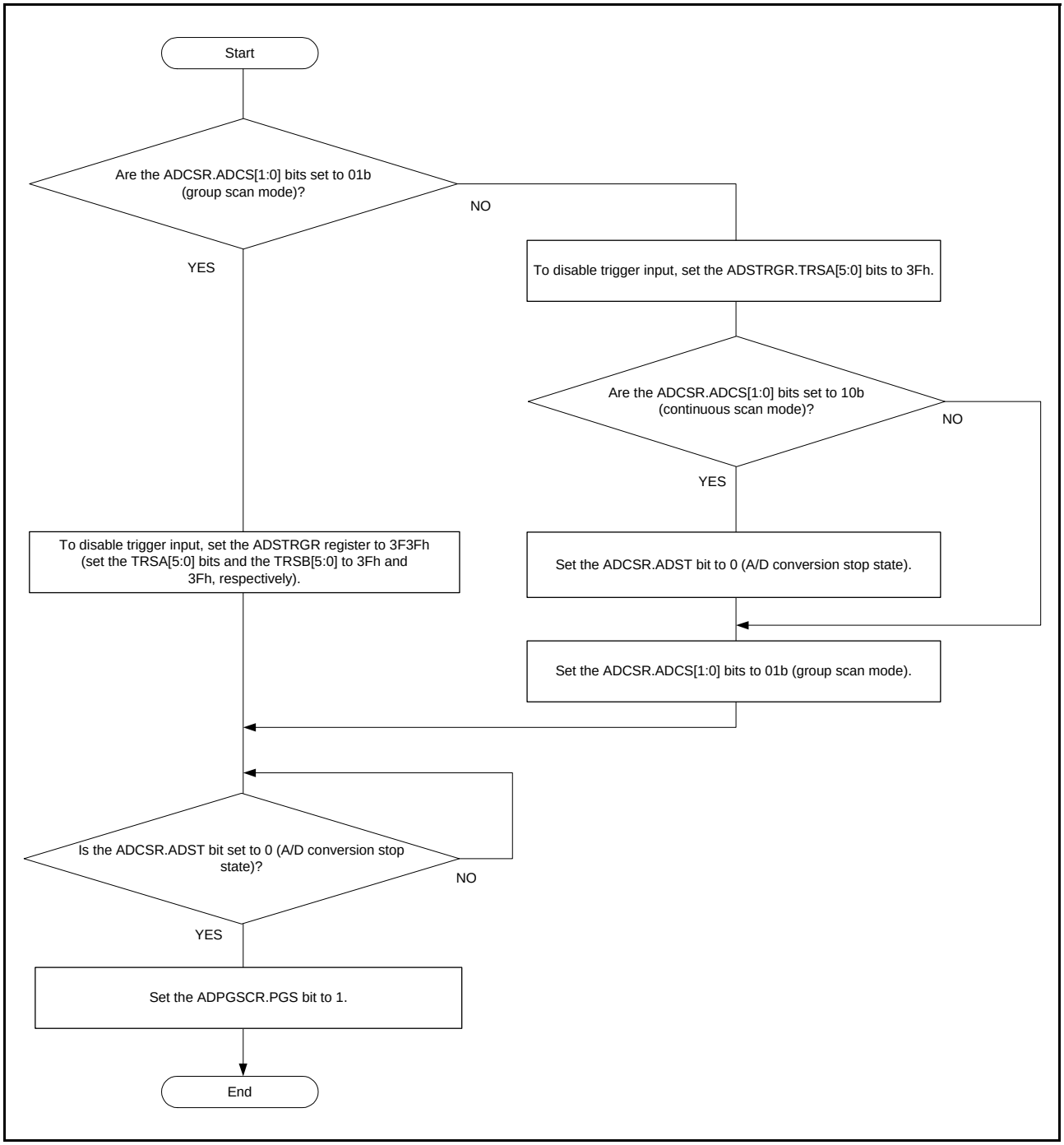


Figure 57.26 Flow of Setting the ADPGSCR.PGS Bit

Table 57.9 Control of A/D Conversion Operations According to the Settings of the ADGSPCR.GBRSCN Bit

A/D Conversion Operation	Trigger Input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group A is in progress	Input of trigger for group A	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group B	Trigger input is ineffective.	A/D conversion is performed on group B after A/D conversion on group A is completed.
When A/D conversion for group B is in progress	Input of trigger for group A	Conversion for group B that is in progress is discontinued and conversion for group A starts.	• Conversion in progress for group B is discontinued and conversion for group A starts. • Conversion for group B starts after conversion for group A is completed.
	Input of trigger for group B	Trigger input is ineffective.	Trigger input is ineffective.

The following describes the operations in group scan mode under group-A priority control (i.e. ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0) when channel 0 is selected for group A and channels 1 to 3 are selected for group B.

(1) When input of a trigger for group B sets the ADCSR.ADST bit to 1 (starting A/D conversion), conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the lowest number.
(2) On completion of A/D conversion, the result is stored in the corresponding A/D data register (ADDRy).
(3) The ADCSR.ADST bit is cleared on the input of a trigger for group A while operation for A/D conversion in group B is in progress, and the latter is discontinued. After that, the ADCSR.ADST bit is set to 1 (starting A/D conversion), and conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the lowest number.
(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(5) An S12ADI interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion is enabled).
(6) After the ADCSR.ADST bit is automatically cleared, again, the bit is automatically set to 1 (starting A/D conversion) and conversion for the ANn channels of group B selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the lowest number.
(7) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(8) An S12GBADI interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1 (S12GBADI interrupt upon group B scanning completion enabled).
(9) The ADCSR.ADST bit retains the value 1 (starting A/D conversion) during A/D conversion and is automatically cleared on completion of conversion, after which the A/D converter enters the waiting state.

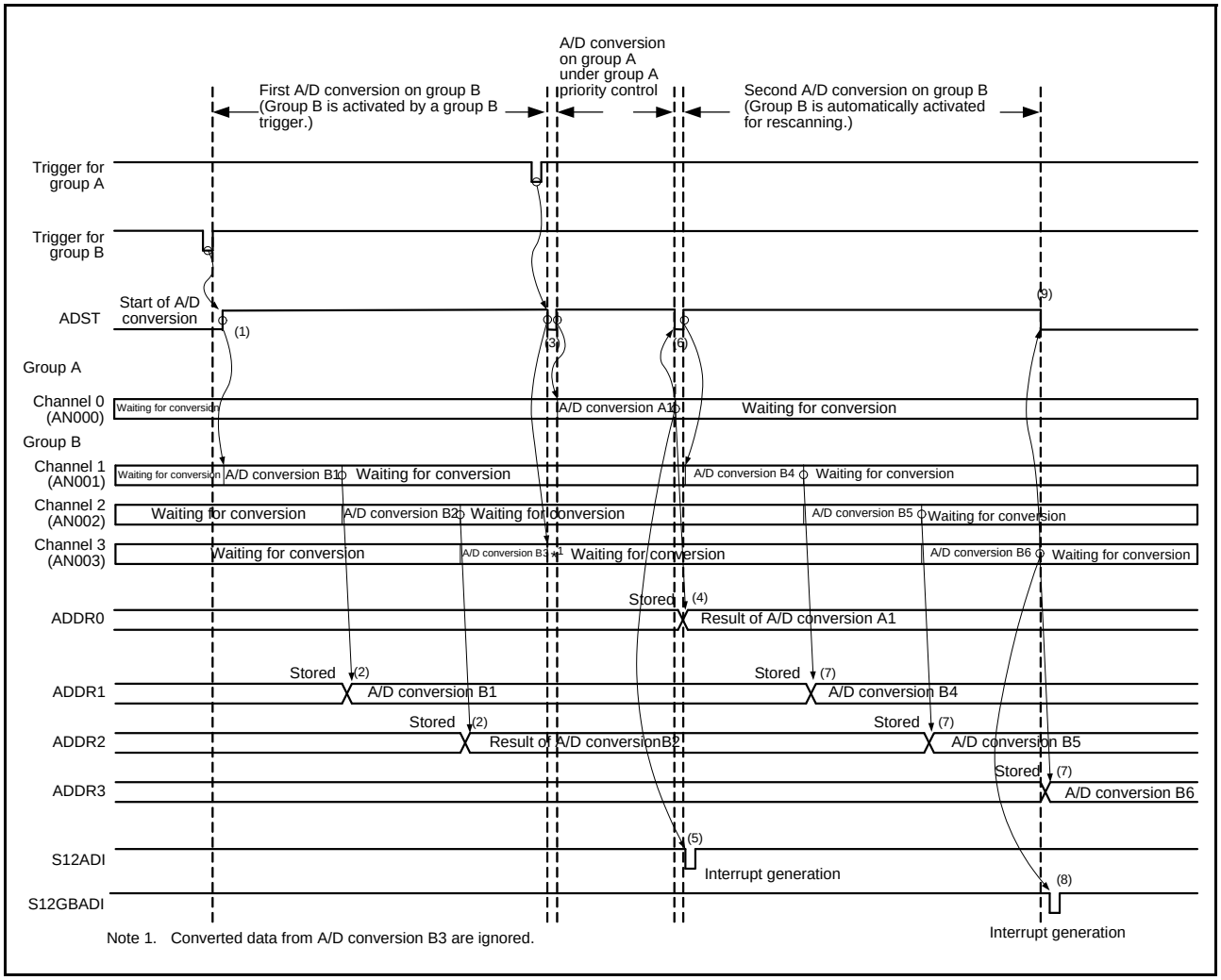


Figure 57.27 Example of Operations under Group-A Priority Control (1) (when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0)

The following is an example when a group A trigger is input again during rescanning operation on group B. In this example, channel 0 is selected for group A and channels 1 to 3 are selected for group B when operation on group A is given priority (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0).

(1) When a group B trigger input sets the ADCSR.ADST bit to 1 (starting A/D conversion), conversion for the ANn channels of group B selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the lowest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(3) The ADCSR.ADST bit is cleared to 0 (stopping A/D conversion) on the input of a trigger for group A while operation for A/D conversion in group B is in progress, and the latter is discontinued.
(4) After that, the ADCSR.ADST bit is set to 1 automatically and A/D conversion for the ANn group A channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the lowest number n.
(5) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(6) An S12ADI interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion is enabled).
(7) On completion of A/D conversion on the group A, rescanning operation on group B sets the ADCSR.ADST bit to 1 automatically if the setting of the ADGSPCR.GBRSCN bit is 1 (enabling rescanning operation). After that, A/D conversion for the ANn group B channels selected in the ADANSB0 and ADANSB1 registers starts again in order from the channel with the lowest number n.
(8) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(9) If a group A trigger is input during A/D conversion on group B for rescanning, the ADCSR.ADST bit is cleared to 0 (stopping A/D conversion) and the ongoing A/D conversion on group B is stopped.
(10) After that, the ADCSR.ADST bit is set to 1 automatically and A/D conversion for the ANn group A channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the lowest number n.
(11) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(12) An S12ADI interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion is enabled).
(13) On completion of A/D conversion on group A, rescanning operation on group B sets the ADCSR.ADST bit to 1 automatically if the setting of the ADGSPCR.GBRSCN bit is 1 (enabling rescanning operation). After that, A/D conversion for the ANn group B channels selected in the ADANSB0 and ADANSB1 registers starts again in order from the channel with the lowest number n.
(14) If a group A trigger is input during A/D conversion on group B for rescanning, steps 9 to 13 are repeated. If a group A trigger is not input, the ADCSR.ADST bit is cleared automatically on completion of A/D conversion on group B and the 12-bit A/D converter enters a wait state.

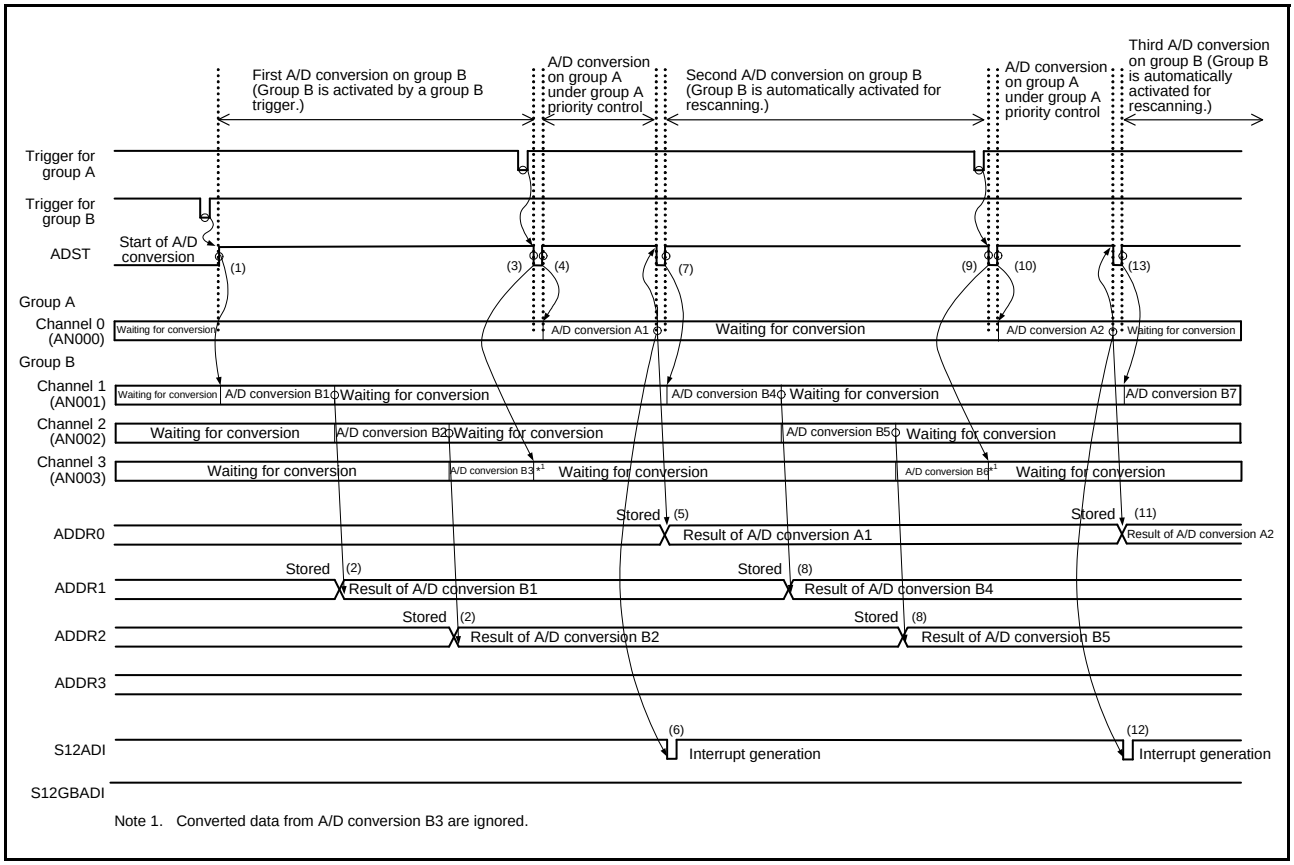


Figure 57.28 Example of Operations under Group-A Priority Control (2) (when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0)

The following is an example of a rescanning operation in which a group B trigger is input during A/D conversion on group A. In this example, channels 1 to 3 are selected for group A and channel 0 is selected for group B when operation on group A is given priority (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0).

(1) When input of a trigger for group A sets the ADCSR.ADST bit to 1 (starting A/D conversion), conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the lowest number.

(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).

(3) If a group B trigger is input during A/D conversion on group A, A/D conversion on group B can be performed after the A/D conversion on group A is completed. (However, if group A triggers are input continuously, the scan operation on group B is cancelled by group A and is not performed.)

(4) On completion of the A/D conversion on the group A, an S12ADI interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion is enabled).

(5) On completion of the A/D conversion on the group A, activation of group B for rescanning sets the ADCSR.ADST bit to 1 automatically.
After that, conversion for the ANn channels of group B selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the lowest number.

(6) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).

(7) On completion of the rescanning operation on the group B, an S12GBADI interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1 (S12GBADI interrupt upon scanning completion is enabled).

(8) The ADCSR.ADST bit retains the value 1 (starting A/D conversion) during A/D conversion and is automatically cleared on completion of conversion, after which the A/D converter enters a waiting state.

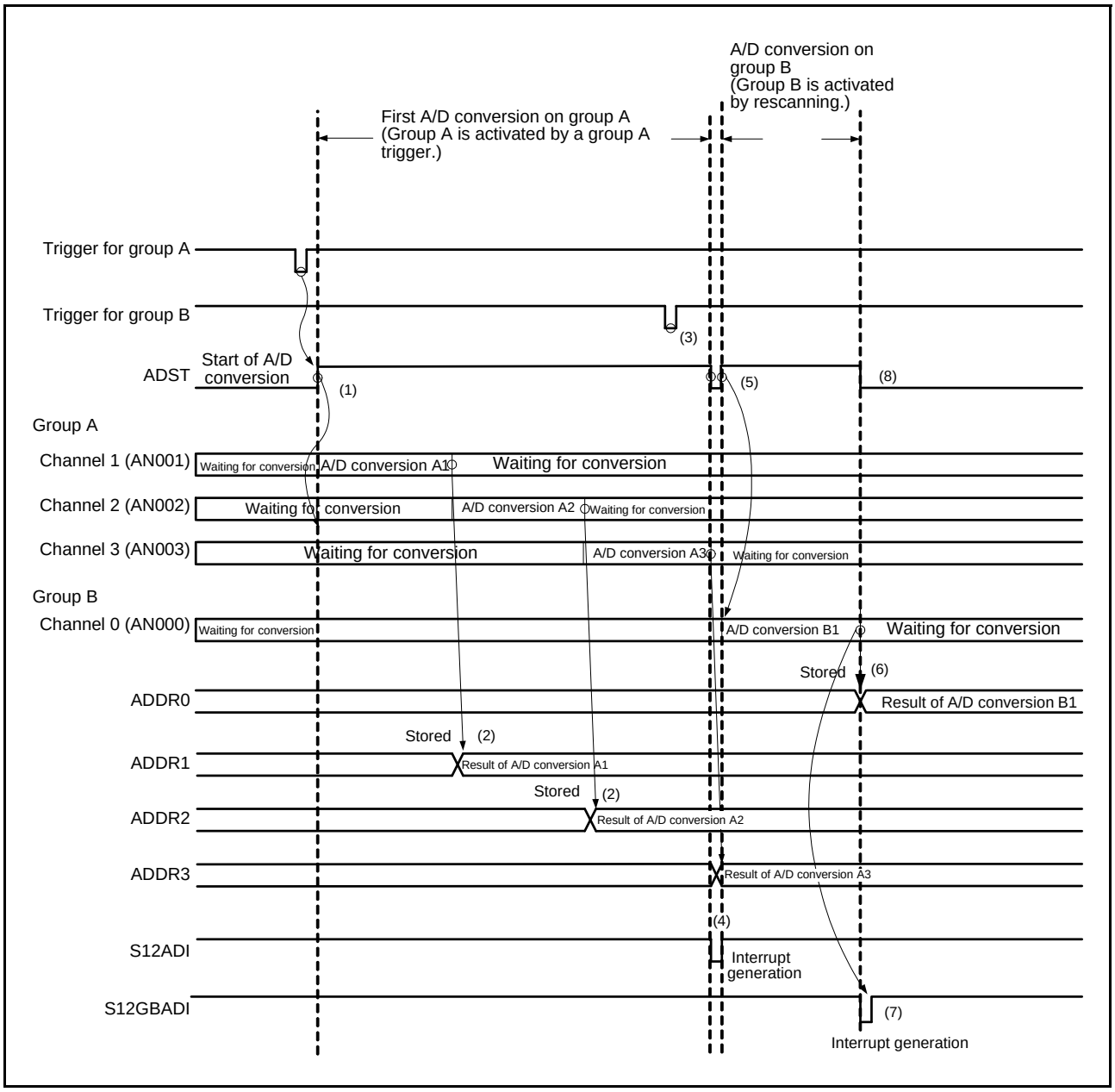
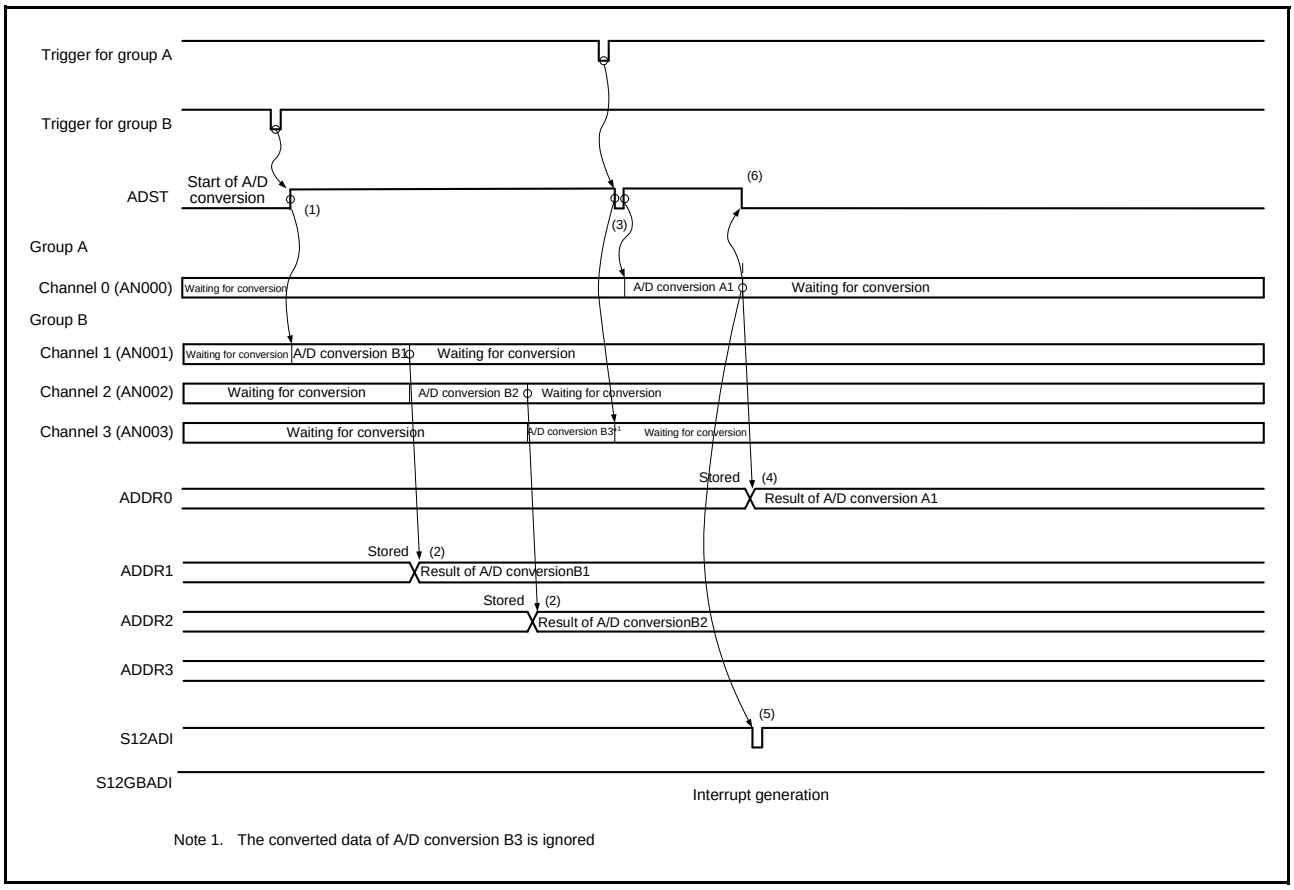


Figure 57.29 Example of Operations under Group-A Priority Control (3) (when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0)

The following is an example of operation under group-A priority control in which channel 0 is selected for group A and channels 1 to 3 are selected for group B (ADGSCR.GBRSCN = 0, ADGSCR.GBRP = 0).

(1) When input of a trigger for group B sets the ADCSR.ADST bit to 1 (starting A/D conversion), conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the lowest number.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(3) If a group A trigger is input during A/D conversion on group B, the ADCSR.ADST bit is cleared to 0 and the ongoing A/D conversion on group B is stopped. After that, the ADCSR.ADST bit is set to 1 (starting A/D conversion) and conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the lowest number.
(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(5) An S12ADI interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion is enabled).
(6) The ADCSR.ADST bit retains the value 1 (starting A/D conversion) during A/D conversion and is cleared on completion of conversion, after which the A/D converter enters the waiting state.

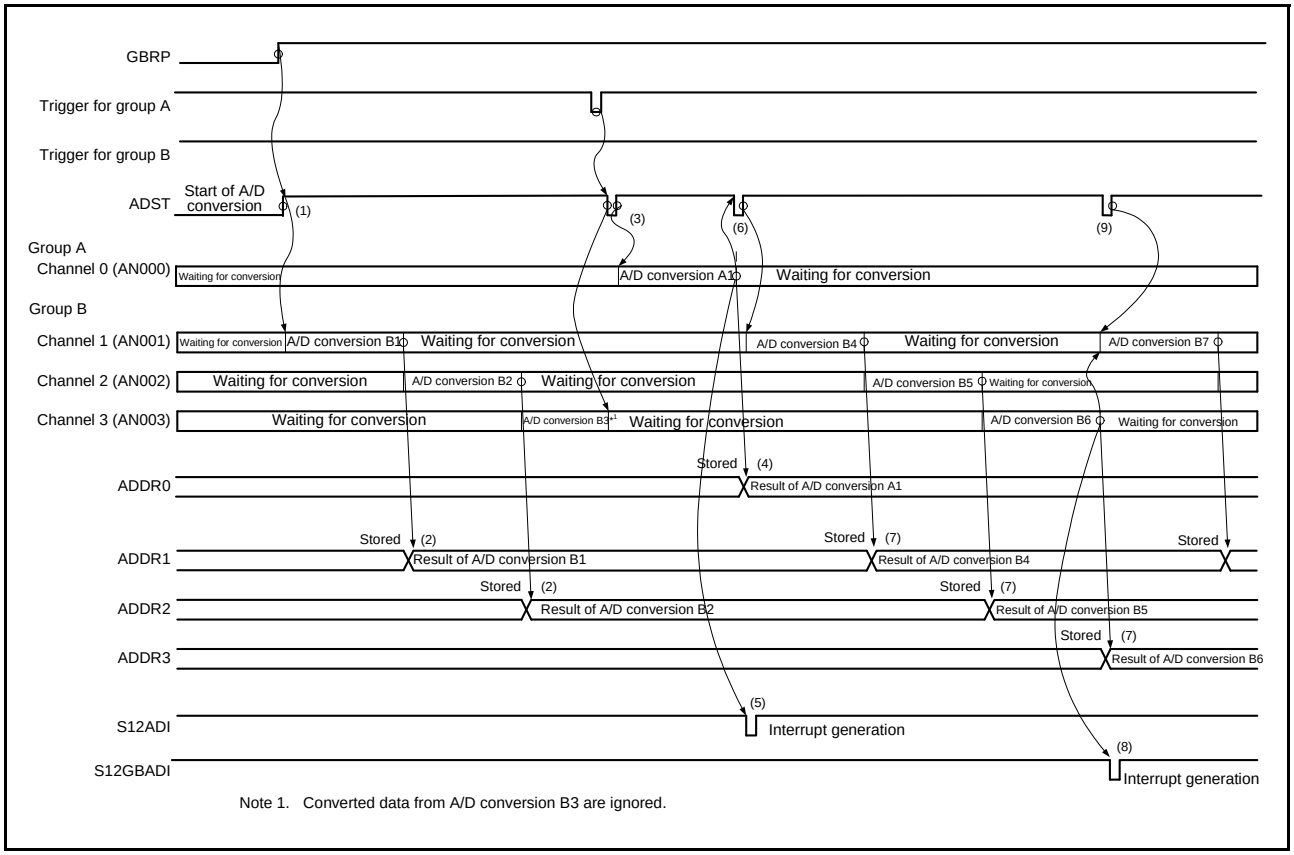


Note 1. The converted data of A/D conversion B3 is ignored

Figure 57.30 Example of Operation under Group-A Priority Control (4) (when ADGSPCR.GBRSCN = 0 and ADGSPCR.GBRP = 0)

The following is an example of operation under group A priority control in which channel 0 is selected for group A and channels 1 to 3 are selected for group B (ADGSCR.GBRP = 1).

(1) The ADCSR.ADST bit is set to 1 (starting A/D conversion) when ADGSPCR.GBRP is set to 1, and conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the lowest number.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(3) If a group A trigger is input during A/D conversion on group B, the ADCSR.ADST bit is cleared to 0 and the ongoing A/D conversion on group B is stopped. After that, the ADCSR.ADST bit is set to 1 (starting A/D conversion) and conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the lowest number.
(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(5) An S12ADI interrupt request is generated if the setting of the ADCSR.ADIE bit is 1 (S12ADI interrupt upon scanning completion is enabled).
(6) After the ADCSR.ADST bit is automatically cleared, again, the ADCSR.ADST bit is automatically set to 1 (starting A/D conversion) and conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the lowest number.
(7) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(8) An S12GBADI interrupt request is generated if the setting of the ADCSR. GBADIE bit is 1.
(9) After the ADCSR.ADST bit is automatically cleared, again, the bit is automatically set to 1 (starting A/D conversion) and conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the lowest number. Steps 6 to 9 are repeated as long as the ADGSPCR.GBRP bit remains 1.
Clearing of the ADCSR.ADST bit to 0 is prohibited while the ADGSPCR.GBRP bit is set to 1.
Follow the procedure for clear operation by software through the ADCSR.ADST bit, shown in Figure 57.41, if you wish to forcibly stop A/D conversion while ADGSPCR.GBRP = 1.



Note 1. Converted data from A/D conversion B3 are ignored.

Figure 57.31 Example of Operation under Group-A Priority Control (5) (when ADGSPCR.GBRP = 1)

57.3.5 Extended Analog Input

The extended analog input ANEX1 is used when an external operational amplifier is connected to the LSI to perform A/D conversion for the multiple analog values. When the extended analog input is selected, AN100 to AN107 can only be selected. Do not select AN108 to AN120, the temperature sensor output, or internal reference voltage. When the extended analog input is selected, self-diagnosis and disconnection detection assistance cannot be used.

57.3.5.1 Usage of ANEX1

To perform A/D conversion of the multiple analog values via the operational amplifier, the analog values are input using the analog input channels of the LSI (AN100 to AN107). Then the time-divided analog values are taken from the extended analog output (ANEX0), and the operational amplifier is connected between ANEX0 and ANEX1.
To select ANEX1, set the ADEXICR.EXSEL[1:0] bits to 01b. To enable ANEX0 output, set the ADEXICR.EXOEN bit to 1. Also select single scan mode or continuous scan mode. Do not select group scan mode. Figure 57.32 shows an example of the extended analog input circuit configuration. Figure 57.33 shows the operation when three channels (AN100, AN101, and AN102) and single scan mode are selected.

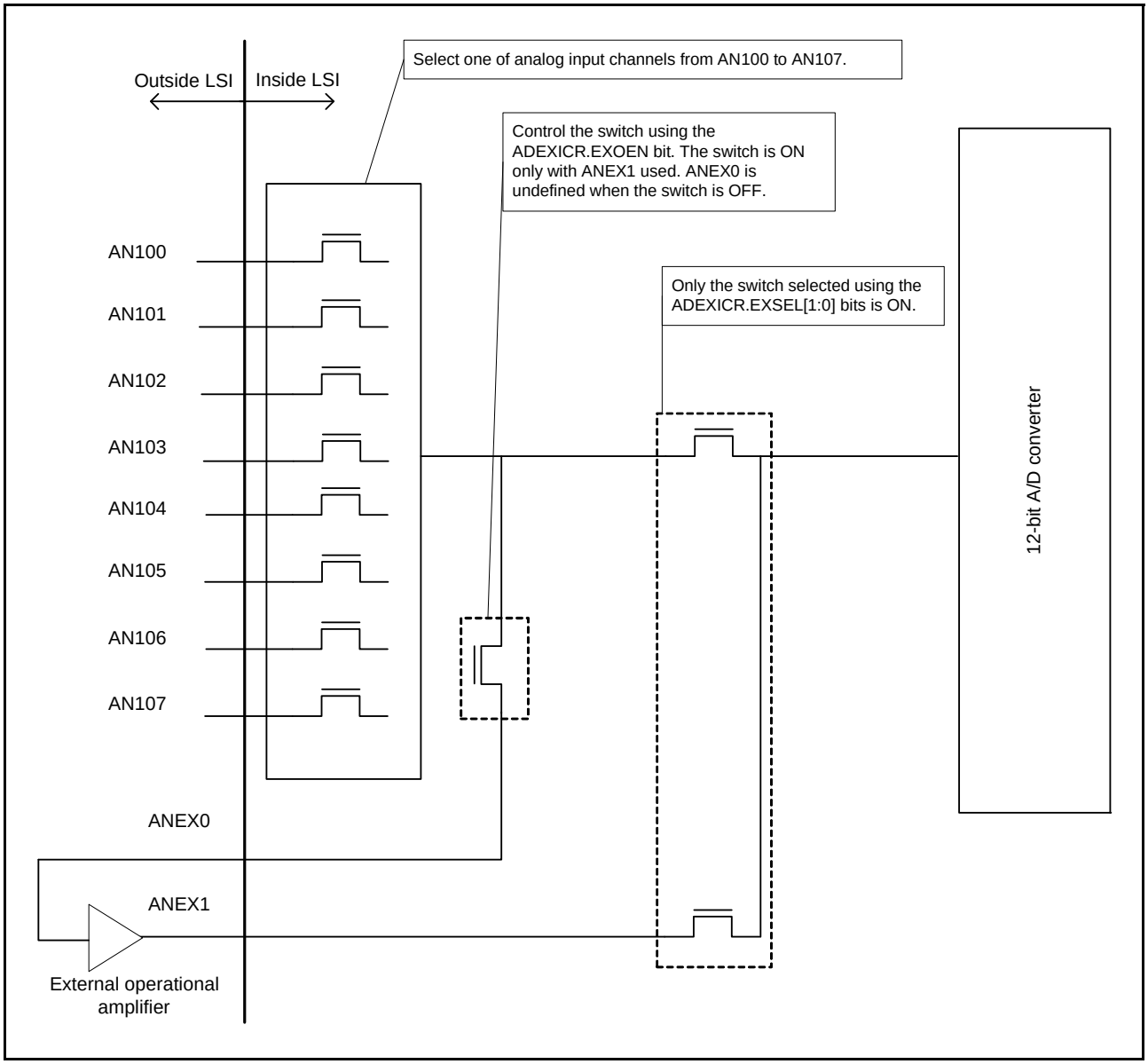


Figure 57.32 Configuration Example of Extended Analog Input Circuit

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (MTU, ELC, GPT, TMR, TPU), or an asynchronous trigger input, A/D conversion is performed for the selected channels, starting from the channel with the smallest number.

(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).

(3) When A/D conversion of all the selected channels is completed, an S12ADI interrupt request is generated if the ADIE bit in ADCSR register is 1 (S12ADI interrupt upon scanning completion is enabled).

(4) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a waiting state.

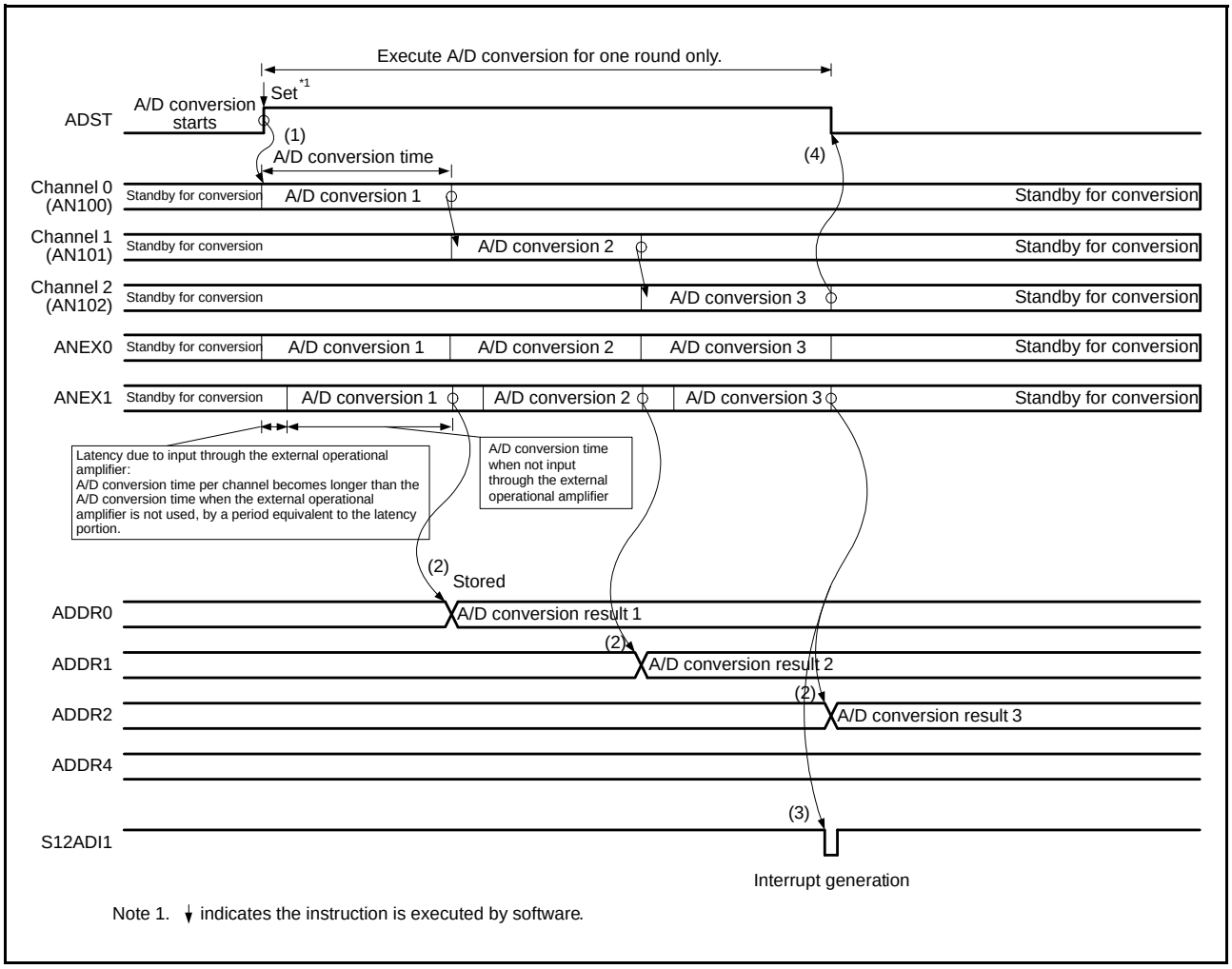


Figure 57.33 ANEX1 Input Operation Example (Single Scan Mode)

When the extended analog input is selected, the individual A/D conversion time becomes longer due to the latency of the operational amplifier, compared to the A/D conversion that is directly performed for the analog input channels.

57.3.6 Comparison

Comparison is of a reference value set in a register with the result of A/D conversion on selected channels. Self-diagnosis function and double-triggered mode are not available while comparison is in use.

Operation using comparison in combination with continuous scan mode is described below.

(1) A/D conversion is started in the order of the selected channels, the temperature sensor output, and the internal reference voltage when the ADCSR.ADST bit is set to 1 (to start A/D conversion) by software, or in response to a synchronous trigger (MTU, GPT, TMR, TPU, ELC) or asynchronous trigger.

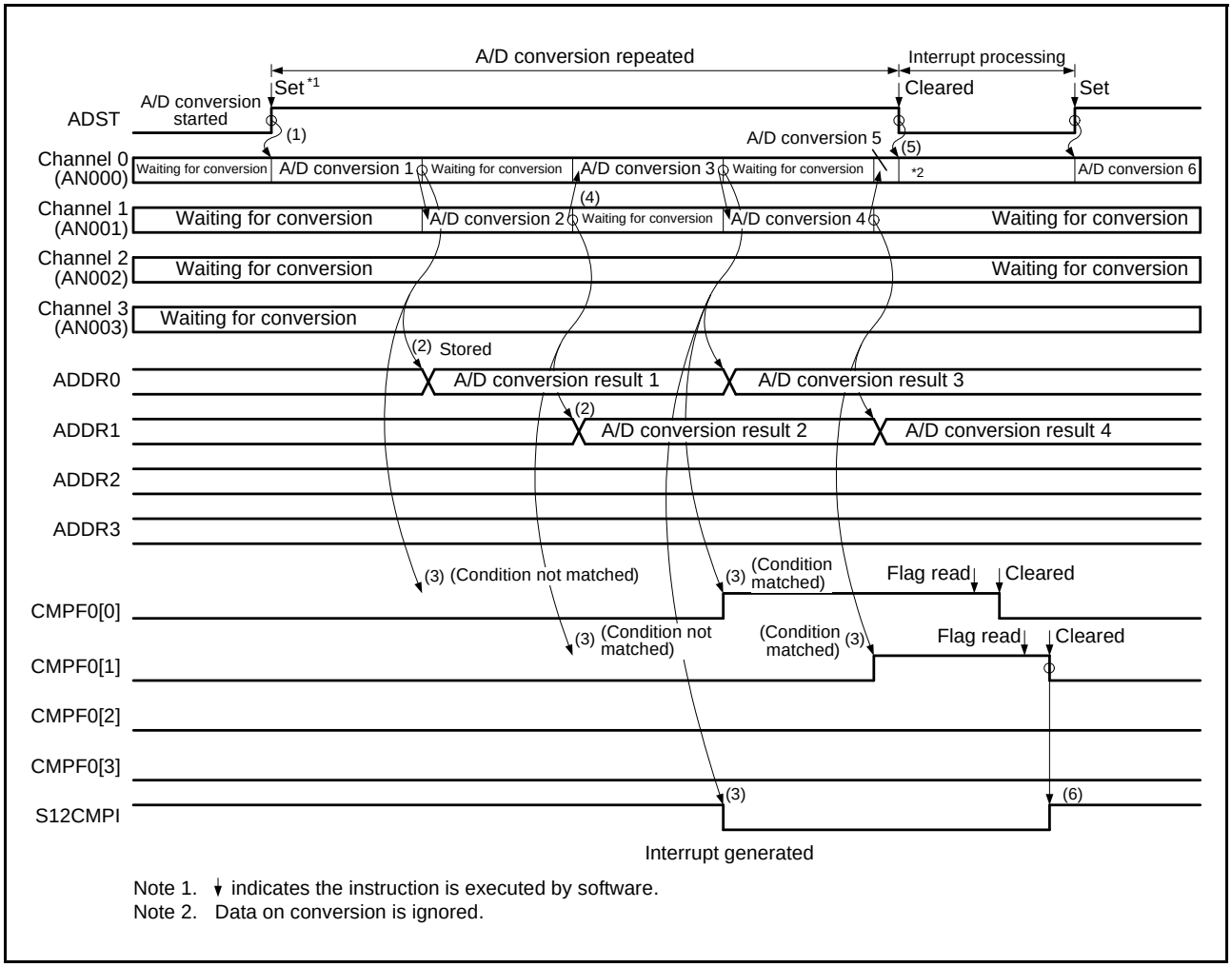
(2) When A/D conversion is completed, the result is stored in the corresponding A/D data register (ADDRy, ADTSDR, ADOCDR). If the register is selected for comparison by the settings of the ADCMPANSRy and ADCMPANSER registers, its value is then compared with that of the ADCMPDR0/1 registers.

(3) If the result of comparison meets the condition set in the ADCMPCR.WCMPE bit, the ADCMPLR0/1 register, and the ADCMPLER register, the bits ADCMPSR0.CMPF0n, ADCMPSR1.CMPF1n, ADCMPSER.CMPFTS, and ADCMPSER.CMPOC are set to 1. If the setting of the ADCMPCR.CMPIE bit is 1 at this time, an S12CMPI interrupt request is also generated.

(4) When A/D conversion is completed for all selected channels, A/D conversion is started again.

(5) If the setting of the ADCSR.ADST bit is 0 (A/D conversion stop) after the S12CMPI interrupt is accepted, processing proceeds for channels that have the compare flag.

(6) The S12CMPI interrupt signal is deasserted when all of the compare flags have been cleared. To start further comparison, start A/D conversion again.



Note 1. ↓ indicates the instruction is executed by software.
Note 2. Data on conversion is ignored.

Figure 57.34 Operation Example of Comparison (AN000 to AN003 Comparison Targets)

57.3.7 Analog Input Sampling and Scan Conversion Time

Scan conversion can be activated either by a software trigger; a synchronous trigger (MTU, ELC, GPT, TMR, TPU); or an asynchronous trigger (ADTRGn#). After the start-of-scanning-delay time (t_D) has elapsed, processing by the channel-dedicated sample-and-hold circuits, processing for disconnection detection assistance, and processing of conversion for self-diagnosis proceed, and this is followed by processing for A/D conversion.

Figure 57.35 shows the scan conversion timing in single scan mode, in which scan conversion is activated by a software trigger or a synchronous trigger (MTU, ELC, GPT, TMR, TPU). Figure 57.36 shows the scan conversion timing in single scan mode, in which scan conversion is activated by an asynchronous trigger ADTRGn#. The scan conversion time (t_{SCAN}) includes the start-of-scanning-delay time (t_D), channel-dedicated sample-and-hold circuit processing time (t_{SPLSH})*1, disconnection detection assistance processing time (t_{DIS})*2, self-diagnosis A/D conversion processing time (t_{DIAG})*3, A/D conversion processing time (t_{CONV}), channel-dedicated sample-and-hold circuit end time (t_{SHED})*4, and end-of-scanning-delay time (t_{ED}).

The A/D conversion processing time (t_{CONV}) consists of input sampling time (t_{SPL}) and time for conversion by successive approximation (t_{SAM}). The sampling time (t_{SPL}) is used to charge sample-and-hold circuits in the A/D converter. If there is not sufficient sampling time due to the high impedance of an analog input signal source, or if the A/D conversion clock (ADCLK) is slow, sampling time can be adjusted using the ADSSTR register.

The time for conversion by successive approximation (t_{SAM}) is at 13 ADCLK states with 12-bit accuracy selected, 11 ADCLK states with 10-bit accuracy selected, and 9 ADCLK states with 8-bit accuracy selected. Table 57.11 shows the scan conversion time.

The scan conversion time (t_{SCAN}) in single scan mode for which the number of selected channels is n can be determined as follows:

$$t_{SCAN} = t_D + t_{SPLSH} + (t_{DIS} \times n) + t_{DIAG} + (t_{CONV} \times n) + t_{ED}$$

The scan conversion time for the first cycle in continuous scan mode is t_{SCAN} for single scan minus t_{ED} plus t_{SHED}. The scan conversion time for the second and subsequent cycles in continuous scan mode is fixed to $t_{SPLSH} + (t_{DIS} \times n) + t_{DIAG} + (t_{CONV} \times n) + t_{SHED}$.

Note 1. When no channel-dedicated sample-and-hold circuits are used, $t_{SH} = 0$.
Note 2. When disconnection detection assistance is not selected, $t_{DIS} = 0$.
Note 3. When the self-diagnosis function is not used, $t_{DIAG} = 0$.
Note 4. When no channel-dedicated sample-and-hold circuits are used, $t_{SHED} = 0$. Here, continuous scan mode is assumed. In single scan mode and group scan mode, t_{SHED} is included in the end-of-scanning-delay time (t_{ED}).

Table 57.10 Example of Setting the ADSSTR Register

Use	Setting Range	Sampling Time*1
Standard (initial value)	0Bh	0.18 µs (For PCLKB = ADCLK = 60 MHz)
Use this range if there is not sufficient sampling time due to the high impedance of an analog input signal source.	0Ch to FFh	Example: FFh 4.3 µs (For PCLKB = ADCLK = 60 MHz)
Use this range if ADCLK is less than 60 MHz and the sampling time needs to be less than the initial value.	05h to 0Ah	Example: 0Ah 0.67 µs (For PCLKB = ADCLK = 15 MHz)

Note 1. The sampling time setting should satisfy the electrical characteristics. The sampling time is determined by the following formula.

$$\text{Sampling time (µs)} = \frac{\text{ADSSTR register setting}}{\text{ADCLK (MHz)}}$$

Table 57.11 Times for Conversion during Scanning (in Numbers of Cycles of the ADCLK and PCLKB)

Item			Symbol		Type/Conditions: Synchronous Trigger: MTU, GPT	Type/Conditions: Synchronous Trigger: TMR, TPU, ELC	Type/Conditions: Asynchronous Trigger	Type/Conditions: Software Trigger	Unit
Scan start processing time*1, *2	A/D conversion on group A under group A priority control.	Group B is to be stopped. (Group A is activated after group B is stopped due to an A/D conversion source of group A.)	t_D		4 PCLKB + 6 ADCLK	3 PCLKB + 6 ADCLK	5 PCLKB + 6 ADCLK	—	Cycle
		Group B is not to be stopped. (Activation by an A/D conversion source of group A.)			3 PCLKB + 4 ADCLK	2 PCLKB + 4 ADCLK	4 PCLKB + 4 ADCLK	—	
	A/D conversion when self-diagnosis is enabled	A/D conversion for self-diagnosis is to be started.			3 PCLKB + 6 ADCLK	2 PCLKB + 6 ADCLK	4 PCLKB + 6 ADCLK	6 ADCLK	
		Normal A/D conversion is to be started after completion of self-diagnosis conversion.			2 ADCLK	2 ADCLK	2 ADCLK	2 ADCLK	
		A/D conversion for self-diagnosis is to be started after completion of conversion for continuous scan on the last channel specified.			2 ADCLK	2 ADCLK	2 ADCLK	2 ADCLK	
	Other than above				3 PCLKB + 4 ADCLK	2 PCLKB + 4 ADCLK	4 PCLKB + 4 ADCLK	4 ADCLK	
Channel-dedicated sample-and-hold processing time*1	Sampling time		t_{SPLSH}	t_{SH}	The setting of ADSHCR.SSTSH[7:0] (initial value = 18h) × ADCLKK (continuous sampling disabled) 0 (continuous sampling enabled)				
	Wait time between sampling and A/D Conversion			t_W	12 ADCLK				
Disconnection detection assistance processing time			t_{DIS}		The setting of ADNDIS[3:0] (initial value = 00h) × ADCLK				
Self-diagnosis conversion processing time*1	Sampling time		t_{DIAG}	t_{SPL}	The setting of ADSSTR0 (initial value = 0Bh) × ADCLK				
	Time for conversion by successive approximation	12-bit conversion accuracy		t_{SAM}	15 ADCLK				
		10-bit conversion accuracy			13 ADCLK				
		8-bit conversion accuracy			11 ADCLK				
A/D conversion processing time*1	Sampling time		t_{CONV}	t_{SPL}	The setting of ADSSTRn (n = 0 to 7, L, T, O) (initial value = 0Bh) × ADCLK				
	Time for conversion by successive approximation	12-bit conversion accuracy		t_{SAM}	13 ADCLK				
		10-bit conversion accuracy			11 ADCLK				
		8-bit conversion accuracy			9 ADCLK				
Channel-dedicated sample-and-hold end processing time			t_{SHED}		2 ADCLK				
Scan end processing time*1			t_{ED}		1 PCLKB + 3 ADCLK				

Note 1. Refer to Figure 57.35 and Figure 57.36 for illustration of times t_D, t_{SPLSH}, t_{DIAG}, t_{CONV}, and t_{ED}.
Note 2. This is the maximum time required from software writing or trigger input to A/D conversion start.

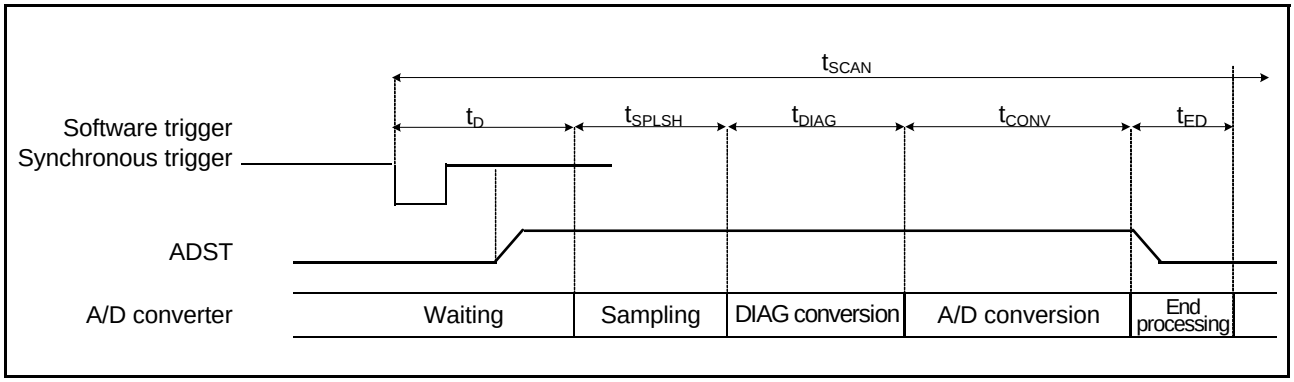


Figure 57.35 Scan Conversion Timing (Activated by Software or Synchronous Trigger Input (MTU, ELC, GPT, TMR, TPU))

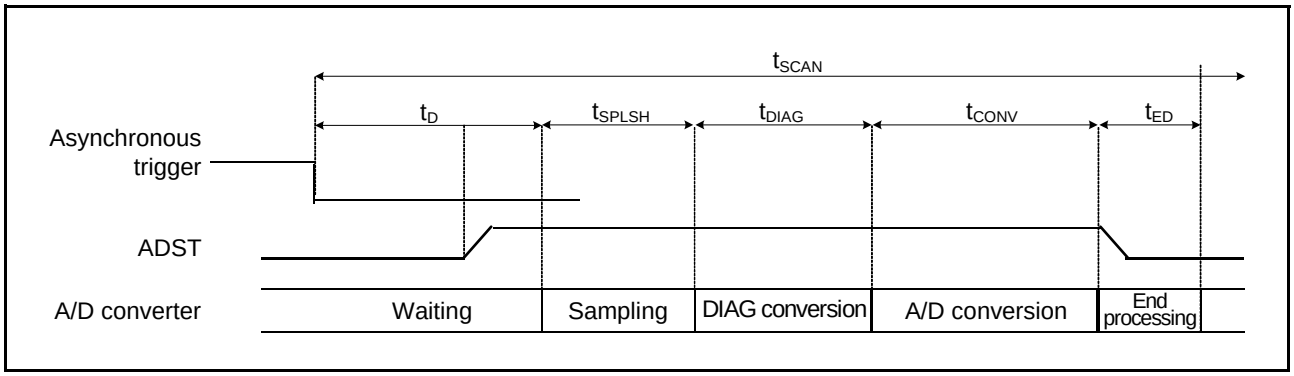


Figure 57.36 Scan Conversion Timing (Activated by Asynchronous Trigger Input (ADTRGn#))

57.3.8 Usage Example of A/D Data Register Automatic Clearing Function

A/D-converted value addition/average mode can be used when A/D conversion of the analog input of the selected channels, A/D conversion of the temperature sensor output, or A/D conversion of the internal reference voltage is selected.

Setting the ACE bit in ADCER register to 1 automatically clears the A/D data registers (ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR) to 0000h when the A/D data registers are read by the CPU, DTC, or DMACA.

This function enables detection of update failures of the A/D data registers (ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR). The following describes the examples in which the function to automatically clear the ADDRy register is enabled and disabled.

In a case where the ACE bit in ADCER register is 0 (automatic clearing is disabled), if the A/D conversion result (0222h) is not written to the ADDRy register for some reason, the old data (0111h) will be the ADDRy value. Furthermore, if this ADDRy value is read into a general register using an A/D scan end interrupt, the old data (0111h) can be saved in the general register. When checking whether there is an update failure, it is necessary to frequently save the old data in the RAM or a general register.

In a case where the ACE bit in ADCER register is 1 (automatic clearing is enabled), when ADDRy = 0111h is read by the CPU, DTC, or DMACA, ADDRy is automatically cleared to 0000h. After that, if the A/D conversion result 0222h cannot be transferred to ADDRy for some reason, the cleared data (0000h) remains as the ADDRy value. If this ADDRy value is read into a general register using an A/D scan end interrupt at this point, 0000h will be saved in the general register. Occurrence of an ADDRy update failure can be determined by simply checking that the read data value is 0000h.

57.3.9 A/D-Converted Value Addition/Average Mode

In A/D-converted value addition mode, the same channel is A/D-converted two to four consecutive times and the sum of the converted values is stored in the data register. In A/D-converted value average mode, the same channel is A/D-converted two to four consecutive times and the mean of the converted values is stored in the data register. The use of the average of these results can improve the accuracy of A/D conversion, depending on the types of noise components that are present. This function, however, cannot always guarantee an improvement in A/D conversion accuracy.
The A/D-converted value addition/average mode can be specified when A/D conversion of the channel select analog input, temperature sensor output, or internal reference voltage is selected.

57.3.10 Disconnection Detection Assist Function

This converter incorporates the function to fix the charge for sampling capacitance to the specified state (VREFH0 or VREFL0 for unit 0; AVCC1 or AVSS1 for unit 1) before start of A/D conversion. This function enables disconnection detection in wiring of analog inputs.

Figure 57.37 illustrates the A/D conversion operation when the disconnection detection assist function is used. Figure 57.38 shows an example of disconnection detection when precharge is selected. Figure 57.39 shows an example of disconnection detection when discharge is selected.

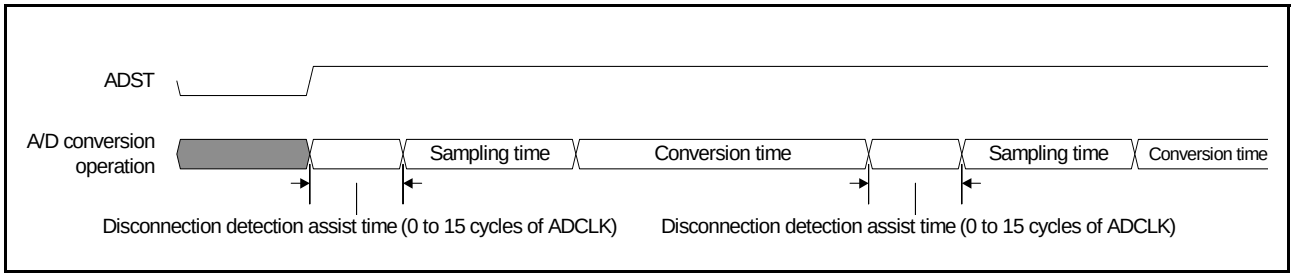
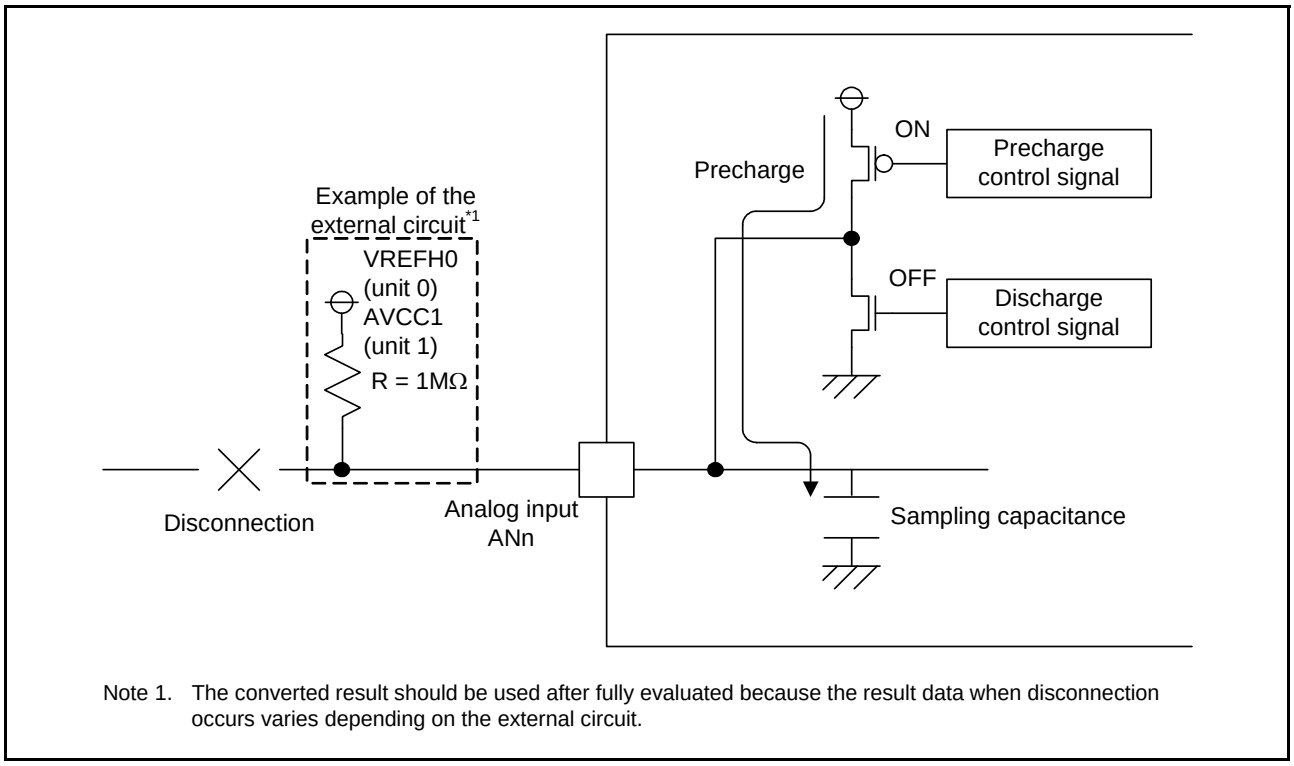
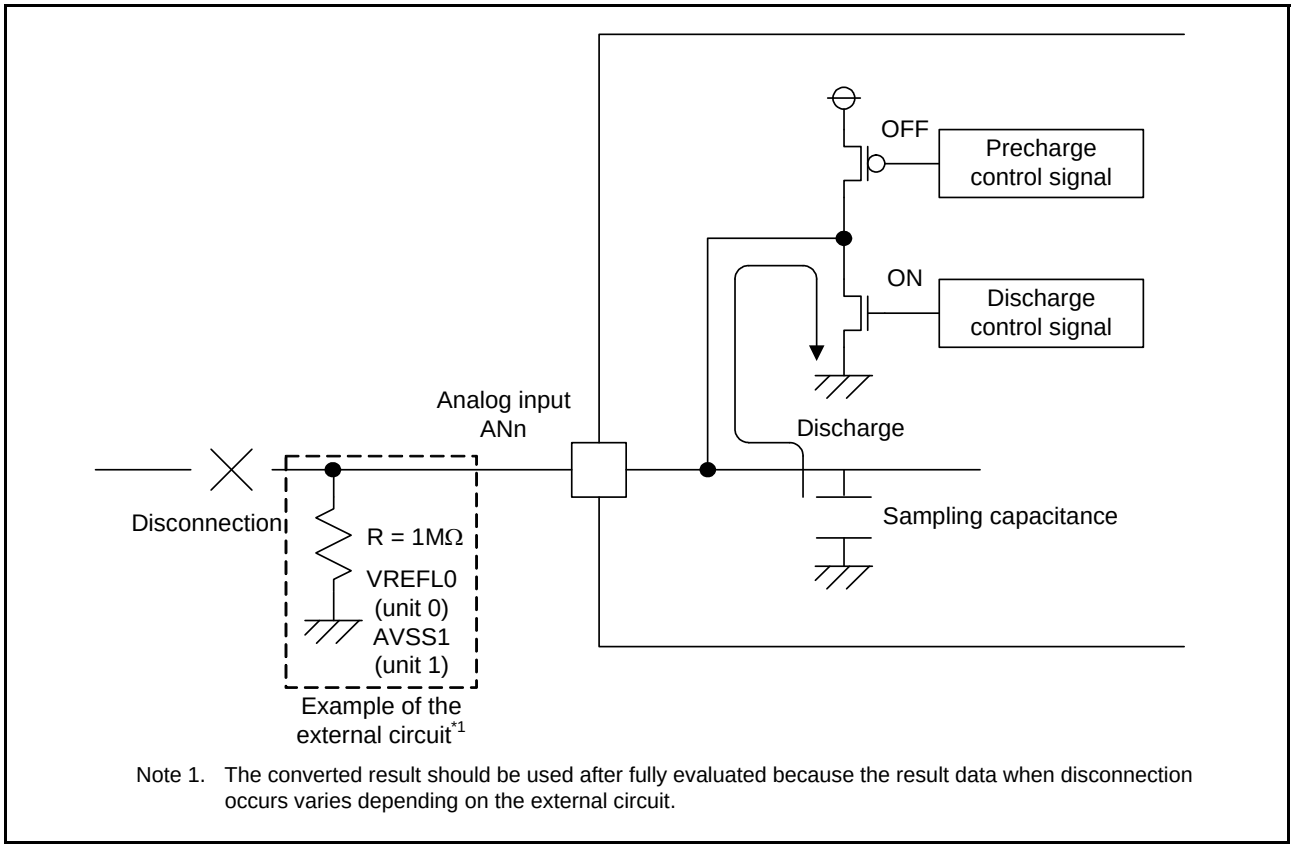


Figure 57.37 Operation of A/D Conversion when the Disconnection Detection Assist Function is Used



Note 1. The converted result should be used after fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 57.38 Example of Disconnection Detection when Precharge is Selected



Note 1. The converted result should be used after fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 57.39 Example of Disconnection Detection when Discharge is Selected

57.3.11 Starting A/D Conversion with Asynchronous Trigger

The A/D conversion can be started by the input of an asynchronous trigger. To start up the A/D converter by an asynchronous trigger, the A/D conversion start trigger select bits (ADSTRGR.TRSA[5:0]) should be set to 000000b and a high-level signal should be input to the asynchronous trigger (ADTRGn# pin). Both the ADCSR.TRGE and ADCSR.EXTRG bits then should be set to 1. Figure 57.40 shows a timing of the asynchronous trigger input.
For the time between setting the ADCSR.ADST bit to 1 and starting A/D conversion, refer to section 57.5.3, A/D Conversion Restarting Timing and Termination Timing.

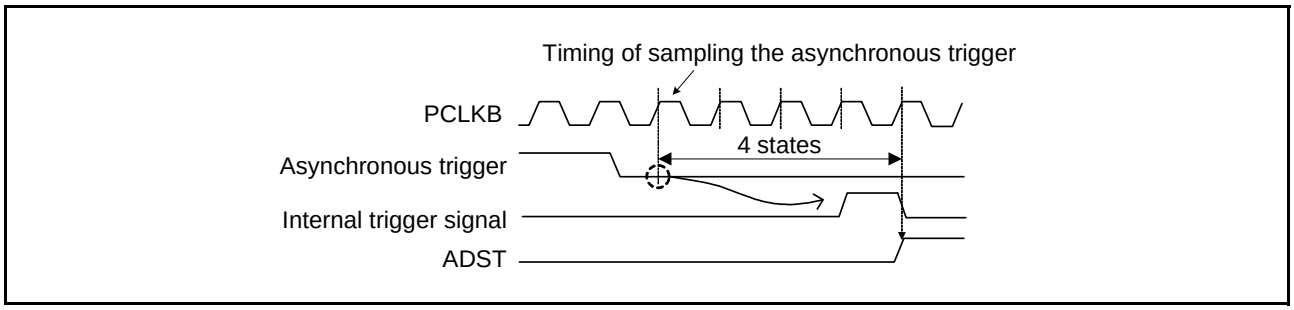


Figure 57.40 Asynchronous Trigger Input Timing

57.3.12 Starting A/D Conversion with Synchronous Trigger from Peripheral Module

The A/D conversion can be started by a synchronous trigger (MTU, ELC, GPT, TMR, TPU). To start the A/D conversion by a synchronous trigger, the ADCSR.TRGE bit should be set to 1, the ADCSR.EXTRG bit should be cleared to 0, and the relevant sources should be selected by the ADSTRGR.TRSA[5:0] and ADSTRGR.TRSB[5:0] bits.

57.4 Interrupt Sources and DTC/DMAC Transfer Requests

57.4.1 Interrupt Requests

The 12-bit A/D converter can send scan end interrupt requests S12ADI and S12GBADI to the CPU. The module also generates the S12CMPI interrupt for the CPU in response to matches with a condition for comparison.
Setting the ADCSR.ADIE bit to 1 and 0 enables and disables an S12ADI interrupt, respectively; similarly, setting the ADCSR.GBADIE bit to 1 and 0 enables and disables a S12GBADI interrupt, respectively. Setting the ADCMPCR.CMPIE bit to 1 and 0 enables and disables an S12CMPI interrupt, respectively.
In addition, the DTC or DMACA can be started up when an S12ADI or an S12GBADI interrupt is generated. Using an S12ADI or an S12GBADI interrupt to allow the DTC or DMACA to read the converted data enables continuous conversion without burden on software.
For details on DTC settings, see section 20, Data Transfer Controller (DTCa), and for details on DMACA settings, see section 18, DMA Controller (DMACAa).
The ADI and GBADI interrupts are output according to the settings of scan mode and double trigger mode as shown in Table 57.12.

Table 57.12 Relationship between Mode Setting and ADI Interrupt Output

Scan mode	Double trigger mode (DBLE)	Trigger	ADI interrupt (ADIE = 1)	GBADI interrupt (GBADIE = 1)
Single scan mode	DBLE = 0	Software trigger	Output on completion of each scan	Not output (group B scan is disabled)
		Synchronous trigger	Output on completion of each scan	Not output (group B scan is disabled)
		Asynchronous trigger	Output on completion of each scan	Not output (group B scan is disabled)
	DBLE = 1	Software trigger	Output on completion of each scan	Not output (group B scan is disabled)
		Synchronous trigger	Output on completion of each even-order scan	Not output (group B scan is disabled)
Continuous scan mode	Setting prohibited	Software trigger	Output on completion of each scan	Not output (group B scan is disabled)
		Synchronous trigger	Output on completion of each scan	Not output (group B scan is disabled)
		Asynchronous trigger	Output on completion of each scan	Not output (group B scan is disabled)
Group scan mode	DBLE = 0	Software trigger	Output on completion of each scan (performed for group A only)	Not output (group B scan is disabled)
		Synchronous trigger	Output on completion of each scan for group A	Output on completion of each scan for group B
	DBLE = 1	Software trigger	Output on completion of each scan (performed for group A only)	Not output (group B scan is disabled)
		Synchronous trigger	Output on completion of each even-order scan for group A	Output on completion of each scan for group B

57.4.2 Scan End Event Output to ELC

The ELC enables a link operation with the modules specified in advance using S12ADI interrupt request signal as an event signal. The S12GBADI and S12CMPI interrupt request signals cannot be used as an event signal. An event signal can be output regardless of the settings of the corresponding interrupt request enable bits. The 12-bit A/D converter outputs an A/D conversion end event.

57.5 Usage Notes

57.5.1 Notes on Reading Data Registers

The A/D data registers, A/D data duplication registers, A/D data duplication register A, A/D data duplication register B, A/D temperature sensor data register, A/D internal reference voltage register, and A/D self-diagnosis data register should be read in word units. If a register is read twice in byte units, that is, the upper byte and lower byte are separately read, the A/D-converted value having been read first may disagree with the A/D-converted value having been read for the second time. To prevent this, the data registers should never be read in byte units.

57.5.2 Notes on Stopping A/D Conversion

To stop A/D conversion when an asynchronous trigger or a synchronous trigger has been selected as the condition for starting A/D conversion, follow the procedure in Figure 57.41.

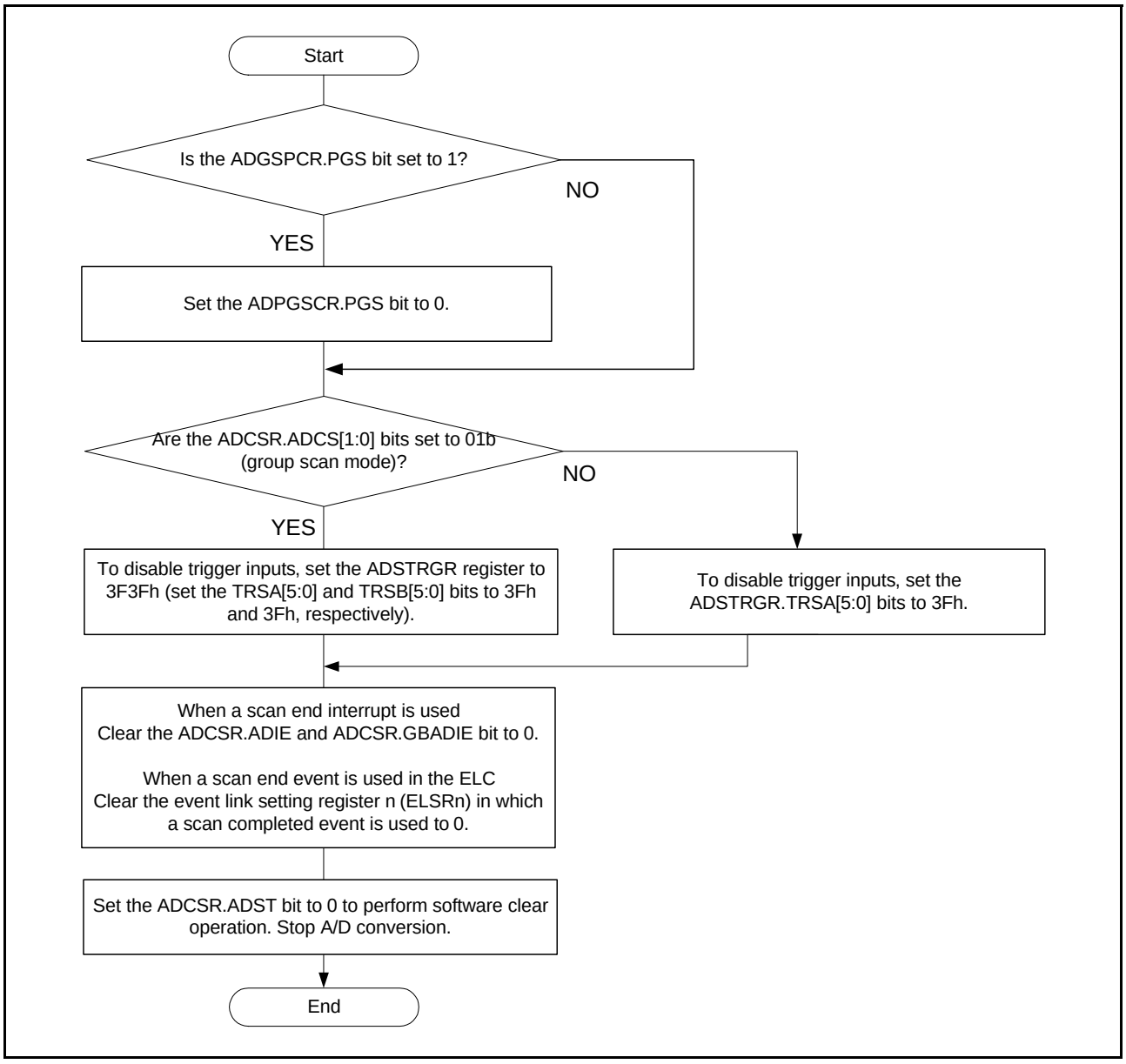


Figure 57.41 Procedures for Clear Operation by Software through the ADCSR.ADST Bit

57.5.3 A/D Conversion Restarting Timing and Termination Timing

It takes a maximum of six ADCLK cycles for the idle analog unit of the 12-bit A/D converter to be restarted by setting the ADST bit in ADCSR register to 1. It takes a maximum of two ADCLK cycles for the operating analog unit of the 12-bit A/D converter to be terminated by setting the ADST bit in ADCSR register to 0.

57.5.4 Notes on Scan End Interrupt Handling

When scanning the same analog input twice using any trigger, the first A/D-converted data is overwritten with the second A/D-converted data in the case that the CPU does not complete reading out the A/D-converted data by the time the A/D conversion of the first analog input for the second scan ends after the first scan end interrupt is generated.

57.5.5 Module Stop Function Setting

Operation of the 12-bit A/D converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the 12-bit A/D converter to be halted. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Consumption.

57.5.6 Notes on Entering Low Power Consumption States

Before entering the module-stop state or software standby mode, make sure to stop A/D conversion. Here, set the ADST bit in ADCSR register to 0, and secure certain period of time until the analog unit of the 12-bit A/D converter is stopped. Follow the procedure given below to secure this time.
Follow the procedure for clear operation by software through the ADCSR.ADST bit, shown in Figure 57.41. After that, wait for two clock cycles of ADCLK before entering the module-stop state or software standby mode.
To place the 12-bit A/D converter in standby mode, set the MSTPCRA.MSTPA17 bit (unit 0) or the MSTPCRA.MSTPA16 bit (unit 1) to 1.

57.5.7 Port Setting When Using the 12-bit A/D Converter Input

When using the 12-bit A/D converter (unit 0), do not use ports 07, 05, 03 and port 4 as output ports. We also recommend not using ports 02, 01, 00, port 9, port D, and port E as output ports. If ports 02, 01, 00, port 9, port D, and port E are used for output signals, perform A/D conversion several times, eliminate the maximum and minimum values, and obtain the average of the other results.
When using the 12-bit A/D converter (unit 1), we recommend not using ports 02, 01, 00, port 9, port D, and port E as output ports. If ports 02, 01, 00, port 9, port D, and port E are used for output signals, perform A/D conversion several times, eliminate the maximum and minimum values, and obtain the average of the other results.

57.5.8 Caution When Using an External Bus

A/D conversion at the same time as access to an external bus may produce poor results.
Perform conversion several times, eliminate the maximum and minimum values, and obtain the average of the other results.

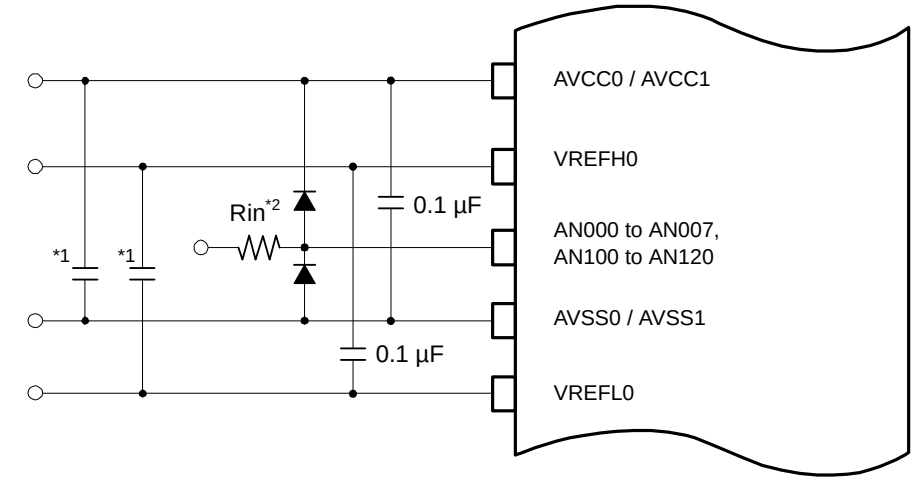
57.5.9 Error in Absolute Accuracy When Disconnection Detection Assistance is in Use

Using disconnection detection assistance leads to an error in absolute accuracy of the A/D converter. This is because an error voltage is input to the analog input pins due to the resistive voltage division between the pull-up or pull-down resistor (Rp) and the resistance of the signal source (Rs). This error in absolute accuracy is calculated from the following formula. Only use disconnection detection assistance after thorough evaluation.

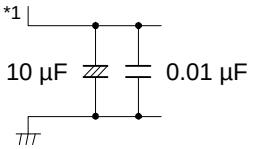
Maximum error in absolute accuracy (LSB) = 4095 × Rs / Rp

57.5.10 Notes on Noise Prevention

To prevent the analog input pins (AN000 to AN007, AN100 to AN120) from being destroyed by abnormal voltage such as excessive surge, a capacitor should be inserted between AVCC0 and AVSS0 and between VREFH0 and VREFL0 and between AVCC1 and AVSS1, and a protection circuit should be connected to protect the analog input pins (AN000 to AN007, AN100 to AN120) as shown Figure 57.42.



Note 1. The values shown here are reference values.



Note 2. Rin: Signal source impedance

Figure 57.42 Sample Protection Circuit for Analog Inputs

57.5.11 Notes on Operating Modes and Status Bits

- When resetting the status of the self-diagnosis voltage, set the ADCER.DIAGLD bit to 1 and set the ADCER.DIAGVAL[1:0] bits to select the voltage value.
- In double trigger mode, when the ADCSR.DBLE bit is modified from 0 to 1, the A/D converter starts from the first scan operation.
- Continuous sampling (the set value of the ADSHMSR.SHMD bit is 1) should be initialized by setting the ADSHMSR.SHMD bit to 0.
  To use continuous sampling again after the function is initialized, wait for at least one ADCLK cycle before setting the ADSHMSR.SHMD bit to 1.

58. 12-Bit D/A Converter (R12DA)

In this section, “PCLK” is used to refer to PCLKB.

58.1 Overview

This MCU includes two channels of 12-bit D/A converter with an output buffer amplifier.
Table 58.1 lists the specifications of the 12-bit D/A converter and Figure 58.1 shows a block diagram of the 12-bit D/A converter.

Table 58.1 Specifications of 12-Bit D/A Converter

Item	Specifications
Resolution	12 bits
Output channels	Two channels
Countermeasure against mutual interference between analog modules	Measure against interference between D/A and A/D conversion D/A converted data update timing is controlled by the 12-bit A/D converter synchronous D/A conversion enable input signal from the 12-bit A/D converter (unit 1). Therefore, the degradation of A/D conversion accuracy due to interference is reduced by controlling the timing in which the 12-bit D/A converter inrush current occurs, with the enable signal.
Low power consumption function	Module stop state can be set.
Event link function (input)	DA0 conversion can be started when an event signal is input.
Output buffer amplifier control function	Buffered output (gain = 1) or unbuffered output can be selected.

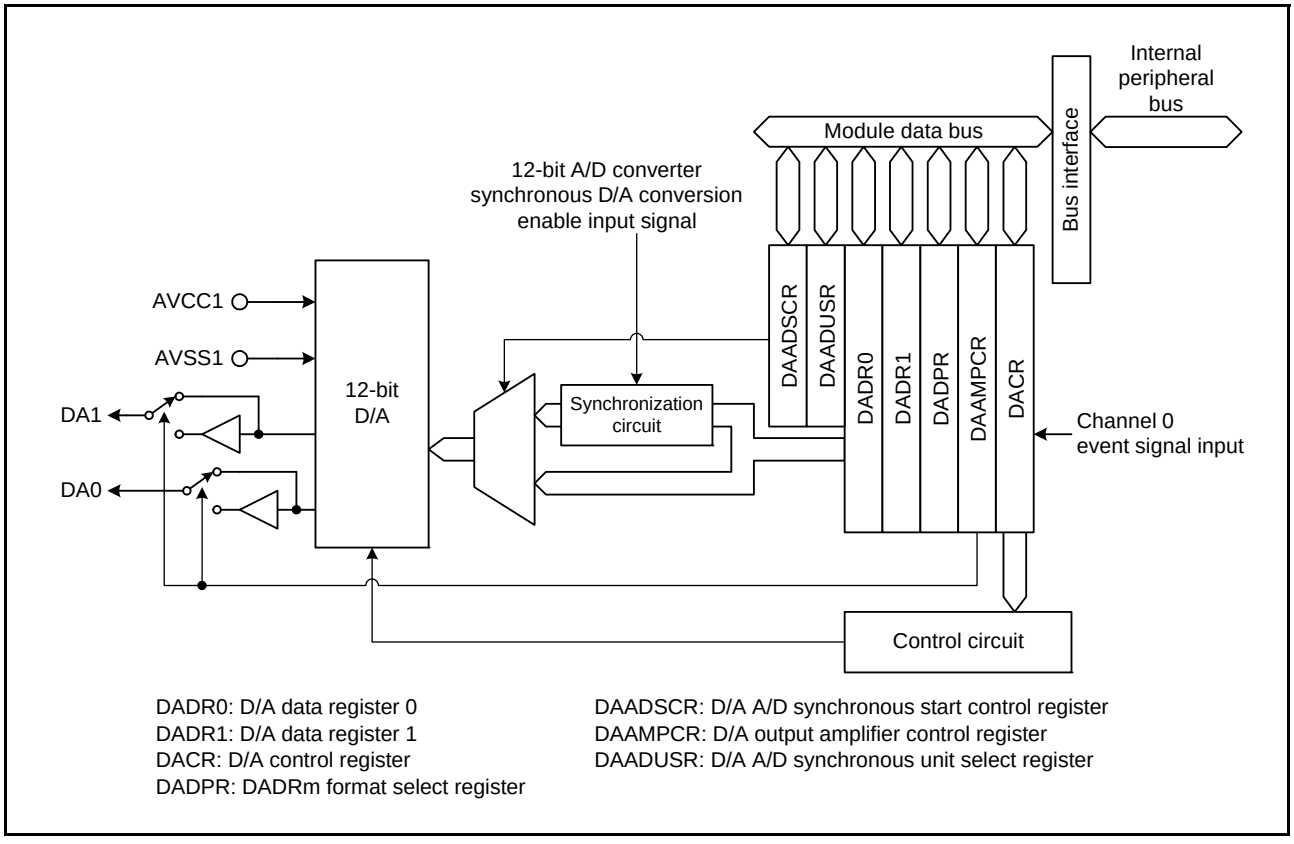


Figure 58.1 Block Diagram of 12-Bit D/A Converter

Table 58.2 lists the pin configuration of the 12-bit D/A converter.

Table 58.2 Pin Configuration of 12-Bit D/A Converter

Pin Name	I/O	Function
AVCC1	Input	Reference voltage input pin for the 12-bit A/D converter (unit 1) and the 12-bit D/A converter: This pin is also used as an analog power supply pin for the modules and temperature sensor. Connect to VCC when the 12-bit A/D converter (unit 1), the 12-bit D/A converter, and the temperature sensor are not used.
AVSS1	Input	Reference voltage input pin for the 12-bit A/D converter (unit 1) and the 12-bit D/A converter: This pin is also used as an analog ground pin for the modules and temperature sensor. Set to the same potential as that of the VSS pin.
DA0	Output	Channel 0 analog output pin
DA1	Output	Channel 1 analog output pin

58.2 Register Descriptions

58.2.1 D/A Data Register m (DADRm) (m = 0, 1)

Address(es): DA.DADR0 0008 8040h, DA.DADR1 0008 8042h

• DADPR.DPSEL bit = 0 (data is flush with the right end of the register)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

• DADPR.DPSEL bit = 1 (data is flush with the left end of the register)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
													—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DADRm register is a 16-bit readable/writable register, which stores data to which D/A conversion is to be performed. Whenever an analog output is enabled, the values in DADRm are converted and output to the analog output pins.

12-bit data can be relocated by setting the DADPR.DPSEL bit.

Bits "—" are read as 0. The write value should be 0.

When the output buffer amplifier is used, see section 58.6.5, Initial Setting Procedure when the Output Buffer Amplifier is Used.

58.2.2 D/A Control Register (DACR)

Address(es): DA.DACR 0008 8044h

	b7	b6	b5	b4	b3	b2	b1	b0
	DAOE1	DAOE0	DAE	—	—	—	—	—
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b5	DAE	D/A Enable*1	0: D/A conversion of channels 0 and 1 is controlled individually. 1: D/A conversion of channels 0 and 1 is enabled collectively.	R/W
b6	DAOE0	D/A Output Enable 0	0: Analog output of channel 0 (DA0) is disabled. 1: D/A conversion of channel 0 is enabled. Analog output of channel 0 (DA0) is enabled.	R/W
b7	DAOE1	D/A Output Enable 1	0: Analog output of channel 1 (DA1) is disabled. 1: D/A conversion of channel 1 is enabled. Analog output of channel 1 (DA1) is enabled.	R/W

Note 1. This bit controls D/A conversion in combination with the DAOEm bit (m = 0, 1). The DAOEm bit controls output of the results of conversion. For details, see Table 58.3.

Table 58.3 Controls of D/A Conversion

b5	b7	b6	
DAE	DAOE1	DAOE0	Description
0	0	0	D/A conversion and analog output pins (DA0, DA1) are disabled.*1
		1	D/A conversion of channel 0 is enabled. D/A conversion of channel 1 is disabled. Analog output of channel 0 (DA0) is enabled. Analog output of channel 1 (DA1) is disabled.*1
	1	0	D/A conversion of channel 0 is disabled. D/A conversion of channel 1 is enabled. Analog output of channel 0 (DA0) is disabled.*1 Analog output of channel 1 (DA1) is enabled.
		1	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0, DA1) is enabled.
1	x	x	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0, DA1) is enabled collectively.

x: Don't care

Note 1. When analog output is disabled, the analog output signal is placed in the Hi-Z state.

This register should be set when the DAADSCR.DAADST bit is 1 (measure against interference between D/A and A/D conversion is enabled) while the 12-bit A/D converter (unit 1) is halted (the ADCSR.ADST bit is 0). At that time, the software trigger should be selected for the 12-bit A/D converter (unit 1) trigger to securely stop the 12-bit A/D converter (unit 1).

DAE Bit (D/A Enable)

The DAE bit controls D/A conversion, amplifier operation, and analog output in combination with the DAOEm bit (m = 0, 1) and the DAAMPCR.DAAMPm bit. See Table 58.4 for details.

When the countermeasure against an interference between D/A and A/D conversions is enabled (DAADSCR.DAADST = 1), set the DAE bit while the ADCSR.ADST bit in the 12-bit A/D converter (unit 1) is 0. At that time, the software trigger should be selected for the 12-bit A/D converter (unit 1) trigger to securely stop the 12-bit A/D converter (unit 1).

DAOEm Bit (D/A Output Enable m) (m = 0, 1)

The DAOEm bit controls D/A conversion, amplifier operation, and analog output in combination with the DAE bit and DAAMPCR.DAAMPm bit. See Table 58.4 for details.

When both the DAOEm bit and DAE bit are 0, D/A conversion of channel m is not done and no conversion result is output.

When the countermeasure against an interference between D/A and A/D conversions is enabled (DAADSCR.DAADST = 1), set the DAOEi bit while the ADCSR.ADST bit in the 12-bit A/D converter (unit 1) is 0. At that time, the software trigger should be selected for the 12-bit A/D converter (unit 1) trigger to securely stop the 12-bit A/D converter (unit 1).

The event link function can be used to set the DAOE0 bit to 1. The DAOE0 bit becomes 1 when the event specified by setting the ELSR16 register of the ELC occurs, and output of the D/A conversion results starts.

Table 58.4 D/A Conversion and Analog Output Control (m = 0, 1)

DACR		DAAMPCR	D/A Conversion Operation of Channel m	Amplifier Operation of Channel m	Analog Output of Channel m
DAE	DAOEm	DAAMPm			
0	0	0	Disabled	Disabled	Hi-Z
		1	Disabled	Disabled	Hi-Z
	1	0	Enabled	Disabled	Unbuffered output
		1	Enabled	Enabled	Buffered output
1	0	0	Enabled	Disabled	Unbuffered output
		1	Enabled	Enabled	Buffered output
	1	0	Enabled	Disabled	Unbuffered output
		1	Enabled	Enabled	Buffered output

58.2.3 DADRm Format Select Register (DADPR) (m = 0, 1)

Address(es): DA.DADPR 0008 8045h

	b7	b6	b5	b4	b3	b2	b1	b0
	DPSEL	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DPSEL	DADRm Format Select	0: Data is flush with the right end of the D/A data register. 1: Data is flush with the left end of the D/A data register.	R/W

58.2.4 D/A A/D Synchronous Start Control Register (DAADSCR)

Address(es): DA.DAADSCR 0008 8046h

	b7	b6	b5	b4	b3	b2	b1	b0
	DAADST	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DAADST	D/A A/D Synchronous Conversion	0: 12-bit D/A converter operation does not synchronize with 12-bit A/D converter (unit 1) operation. (measure against interference between D/A and A/D conversion is disabled) 1: 12-bit D/A converter operation synchronizes with 12-bit A/D converter (unit 1) operation. (measure against interference between D/A and A/D conversion is enabled)	R/W

As a measure against interference between D/A and A/D conversion, the DAADSCR register selects whether or not the timing for starting 12-bit D/A conversion is synchronized with the 12-bit A/D converter synchronous D/A conversion enable input signal from the 12-bit A/D converter (unit 1).
This register should be set while the 12-bit A/D converter (unit 1) is halted (while the ADCSR.ADST bit is 0 after selecting software trigger as the 12-bit A/D converter (unit 1) trigger).
Unit 1 should be selected as the target unit of the 12-bit A/D converter before setting the DAADST bit to 1.
Set the DAADUSR.AMADSEL1 bit to 1 to select unit 1.

DAADST Bit (D/A A/D Synchronous Conversion)

Setting the DAADST bit to 0 allows the DADRm register value (m = 0, 1) to be converted into analog data at any time.
Setting the DAADST bit to 1 allows synchronous D/A conversion with the 12-bit A/D converter synchronous D/A conversion enable input signal from the 12-bit A/D converter (unit 1). Therefore, even if the DADRm register value is modified, D/A conversion does not start until the 12-bit A/D converter (unit 1) completes A/D conversion.
Set this bit while the ADCSR.ADST bit is set to 0. At this time, the software trigger should be selected for the 12-bit A/D converter trigger to securely stop the 12-bit A/D converter (unit 1). Set the DAADUSR.AMADSEL1 bit to 1 before setting the DAADST bit to 1.
The event link function cannot be used when the DAADST bit is set to 1. Stop the event link function by setting the ELSR16 register of the ELC. The setting of the DAADST bit is common to channels 0 and 1 of the 12-bit D/A converter.

58.2.5 D/A Output Amplifier Control Register (DAAMPCR)

Address(es): DA.DAAMPCR 0008 8048h

	b7	b6	b5	b4	b3	b2	b1	b0
	DAAMP1	DAAMP0	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	DAAMP0	Amplifier Control 0	0: Output buffer amplifier of channel 0 is not used. 1: Output buffer amplifier of channel 0 is used.	R/W
b7	DAAMP1	Amplifier Control 1	0: Output buffer amplifier of channel 1 is not used. 1: Output buffer amplifier of channel 1 is used.	R/W

The DAAMPCR register selects D/A converter output with or without using the output buffer amplifier.

DAAMP0 Bit (Amplifier Control 0)

When the DAAMP0 bit is 0, the conversion result for the D/A converter channel 0 is output without using the output buffer amplifier. When the DAAMP0 bit is 1, the conversion result for the D/A converter channel 0 is output via the output buffer amplifier.

When both the DAE and DAOE0 bits are 0, the output buffer amplifier is disabled regardless of the setting of the DAAMP0 bit. See Table 58.4 for details.

DAAMP1 Bit (Amplifier Control 1)

When the DAAMP1 bit is 0, the conversion result for the D/A converter channel 1 is output without using the output buffer amplifier. When the DAAMP1 bit is 1, the conversion result for the D/A converter channel 1 is output via the output buffer amplifier.

When both the DAE and DAOE1 bits are 0, the output buffer amplifier is disabled regardless of the setting of the DAAMP1 bit. See Table 58.4 for details.

58.2.6 D/A A/D Synchronous Unit Select Register (DAADUSR)

Address(es): DA.DAADUSR 0008 C5C0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	AMADSEL1	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	AMADSEL1	A/D Unit 1 Select	0: Unit 1 is not selected. 1: Unit 1 is selected.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The DAADUSR register is used to select the target unit of the 12-bit A/D converter for D/A and A/D synchronous conversions. When setting the DAADSCR.DAADST bit to 1 for synchronous conversions, the target unit should be selected by this register in advance. Set the AMADSEL1 bit to 1 and select unit 1 as the target synchronous unit. The DAADUSR register should be set when the ADCSR.ADST bit of the 12-bit A/D converter is set to 0 and the DAADSCR.DAADST bit is set to 0.

58.3 Operation

The 12-bit D/A converter includes D/A conversion circuits for two channels, each of which can operate independently. When the DACR.DAOEm bit (m = 0, 1) is set to 1, D/A converter is enabled and the conversion result is output. An operation example of D/A conversion on channel 0 is shown below. Figure 58.2 shows the timing of this operation.

(1) Set the data for D/A conversion in the DADPR.DPSEL bit and the DADR0 register.

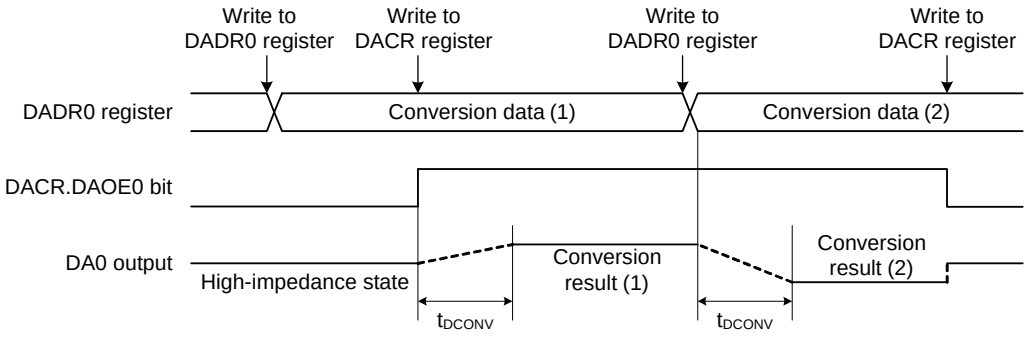
(2) Set the DACR.DAOE0 bit to 1 to start D/A conversion. The conversion result is output from the analog output pin DA0 after the conversion time tDCONV has elapsed. The conversion result continues to be output until the DADR0 register is written to again or the DAOE0 bit is cleared to 0. The output value (reference) is expressed by the following formula:

$$\frac{\text{Value of DADRm register}}{4096} \times \text{AVCC1}$$

When the output buffer amplifier is used, the output voltage does not reach AVSS1 or AVCC1. Refer to section 64, Electrical Characteristics for the output voltage range.

(3) If the DADR0 register is written to again, the conversion is started. The conversion result is output after the conversion time tDCONV has elapsed.
When the DAADSCR.DAADST bit is 1 (measure against interference between D/A and A/D conversion is enabled), it takes a maximum of one A/D conversion time for D/A conversion to start.

(4) If the DAOE0 bit is set to 0, analog output is disabled.



t_{DCONV}: D/A conversion time. Refer to the electric characteristics for the value.

Figure 58.2 Example of 12-Bit D/A Converter Operation

58.3.1 Measure against Interference between D/A and A/D Conversion

When D/A conversion starts, an inrush current occurs to the 12-bit D/A converter. Since the same analog power supply is shared by the 12-bit D/A converter and 12-bit A/D converter (unit 1), the inrush current may interfere with the proper operation of the 12-bit A/D converter (unit 1).
With the DAADSCR.DAADST bit being 1, even if the DADRm register data (m = 0, 1) is modified during 12-bit A/D converter (unit 1) operation, D/A conversion does not start immediately but starts synchronously with A/D conversion completion. It takes a maximum of one A/D conversion time for the DADRm register data update to be reflected as the D/A conversion circuit input. Before reflection, the DADRm register value does not correspond to the analog output value.
When this function is enabled, it is impossible to check by any software means whether the DADRm register value has been D/A converted or not.
Even with the DAADSCR.DAADST bit being 1, when the DADRm register data is modified while the 12-bit A/D converter (unit 1) is halted, D/A conversion starts in one PCLK cycle.
Figure 58.3 shows an example of channel 0 D/A conversion, in which the 12-bit D/A converter operates synchronously with the 12-bit A/D converter (unit 1).

(1) Confirm that the 12-bit A/D converter (unit 1) is halted. Set the DAADUSR.AMADSEL1 bit to 1.
(2) Confirm that the 12-bit A/D converter (unit 1) is halted. Set the DAADSCR.DAADST bit to 1.
(3) Confirm that the 12-bit A/D converter (unit 1) is halted. Set the DACR.DAOE0 bit to 1.
(4) Set the DADR0 register.

- If the 12-bit A/D conversion is halted (ADCSR.ADST bit = 0) when the DADR0 register is modified, D/A conversion starts in one PCLK cycle.
- If the 12-bit A/D conversion is in progress (ADCSR.ADST bit = 1) when the DADR0 register is modified, D/A conversion starts upon A/D conversion completion. If the DADR0 register is modified twice during A/D conversion, the first update may not be converted.

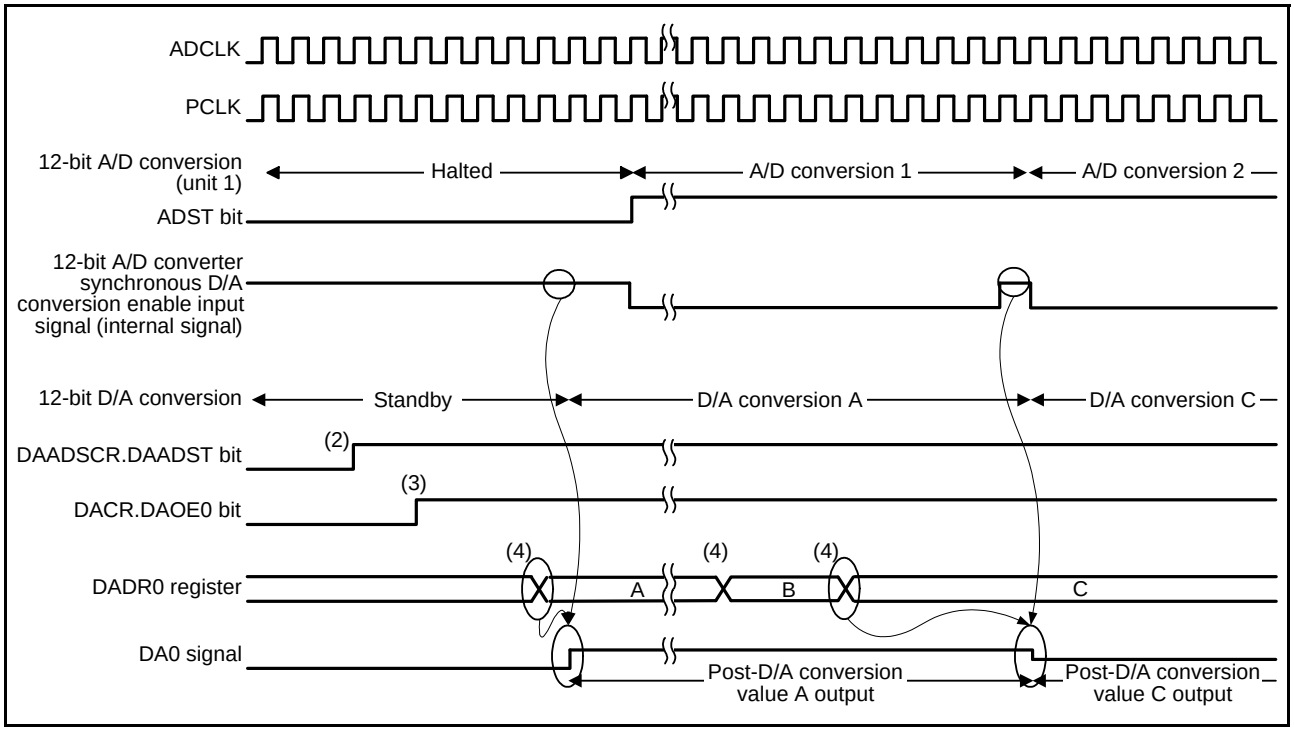


Figure 58.3 Example of Conversion When the 12-Bit D/A Converter is Synchronized with the 12-Bit A/D Converter (Unit 1)

58.4 Event Link Operation Setting Procedure

The event link operation procedure is described below.

(1) Set the DADPR.DPSEL bit and set the data for D/A conversion in the DADR0 register.
(2) Set the bit value of the ELSR16 setting event signal to link the ELSR16 register of the ELC.
(3) Set the ELCR.ELCON bit to 1. This procedure enables event link operation for all modules with the event link function selected.
(4) Set the event output source module to activate the event link. After the event is output from the module, the DACR.DAOE0 bit becomes 1, and D/A conversion on channel 0 starts.
(5) Set the ELSR16.ELS[7:0] bits to 0000 0000b to stop event link operation of 12-bit D/A converter channel 0. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

58.5 Usage Notes on Event Link Operation

(1) When the event link function is used, do not use the output buffer amplifier.
(2) When the event link function is used, set the DACR.DAE bit to 0.
(3) When the event specified by the ELSR16 register is generated while the write cycle is performed to the DACR.DAOE0 bit, the write cycle is stopped, and the setting to 1 by the generated event takes precedence.
(4) Use of the event link function is prohibited when the DAADSCR.DAADST bit is set to 1 as the countermeasure against an interfere between D/A and A/D conversions.

58.6 Usage Notes

58.6.1 Module Stop Function Setting

Operation of the 12-bit D/A converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the 12-bit D/A converter to be stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

58.6.2 Operation of the D/A Converter in Module Stop State

When the MCU enters the module stop state with D/A conversion enabled, the D/A converter outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current has to be reduced in the module stop state, disable D/A conversion by setting the DACR.DAOE1, DAOE0, and DAE bits to 0.

58.6.3 Operation of the D/A Converter in Software Standby Mode

When the MCU enters software standby mode with D/A conversion enabled, the D/A converter outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current has to be reduced in software standby mode, disable D/A conversion by setting the DACR.DAOE1, DAOE0, and DAE bits to 0.

58.6.4 Note on Entering Deep Software Standby Mode

When the MCU enters deep software standby mode with D/A conversion enabled, the outputs of the D/A converter are placed in a high impedance state.

58.6.5 Initial Setting Procedure when the Output Buffer Amplifier is Used

When using the output buffer amplifier, enable the amplifier output in the following procedure. An example for channel 0 is described below.

(1) Write 0000h to the DADR0 register.
(2) Set the DAAMPCR.DAAMP0 bit to 1.
(3) Set the DACR.DAE or DACR.DAOE0 bit to 1. The output buffer amplifier starts the operation.
(4) Wait for at least 3 μs and then write a value to be converted in the DADR0 register.

While the output buffer amplifier is operating, setting the DACR.DAE and DACR.DAOE0 bits to 0 disables the output buffer amplifier. Repeat the procedure from (1) to (4) to use the output buffer amplifier again.

58.6.6 Note on Usage When Measure against Interference between D/A and A/D Conversion is Enabled

When the DAADSCR.DAADST bit is 1 (measure against interference between D/A and A/D conversion is enabled), do not place the 12-bit A/D converter (unit 1) in the module stop state. It may halt D/A conversion in addition to A/D conversion.

59. Temperature Sensor (TEMPS)

59.1 Overview

This MCU includes a temperature sensor. The temperature sensor outputs a voltage which varies with the temperature. The 12-bit A/D converter unit 1 can convert the voltage from the sensor into a digital value. The temperature around the MCU can be obtained by converting the value into the temperature.

Table 59.1 lists the specifications of the temperature sensor, and Figure 59.1 shows a block diagram of the temperature sensor.

Table 59.1 Specifications of Temperature Sensor

Item	Description
Temperature sensor voltage output	Temperature sensor outputs a voltage to the 12-bit A/D converter unit 1.
Low-power consumption function	The module-stop state is selectable.
Temperature Sensor Calibration Data	Reference data measured for each chip at factory shipment is stored.

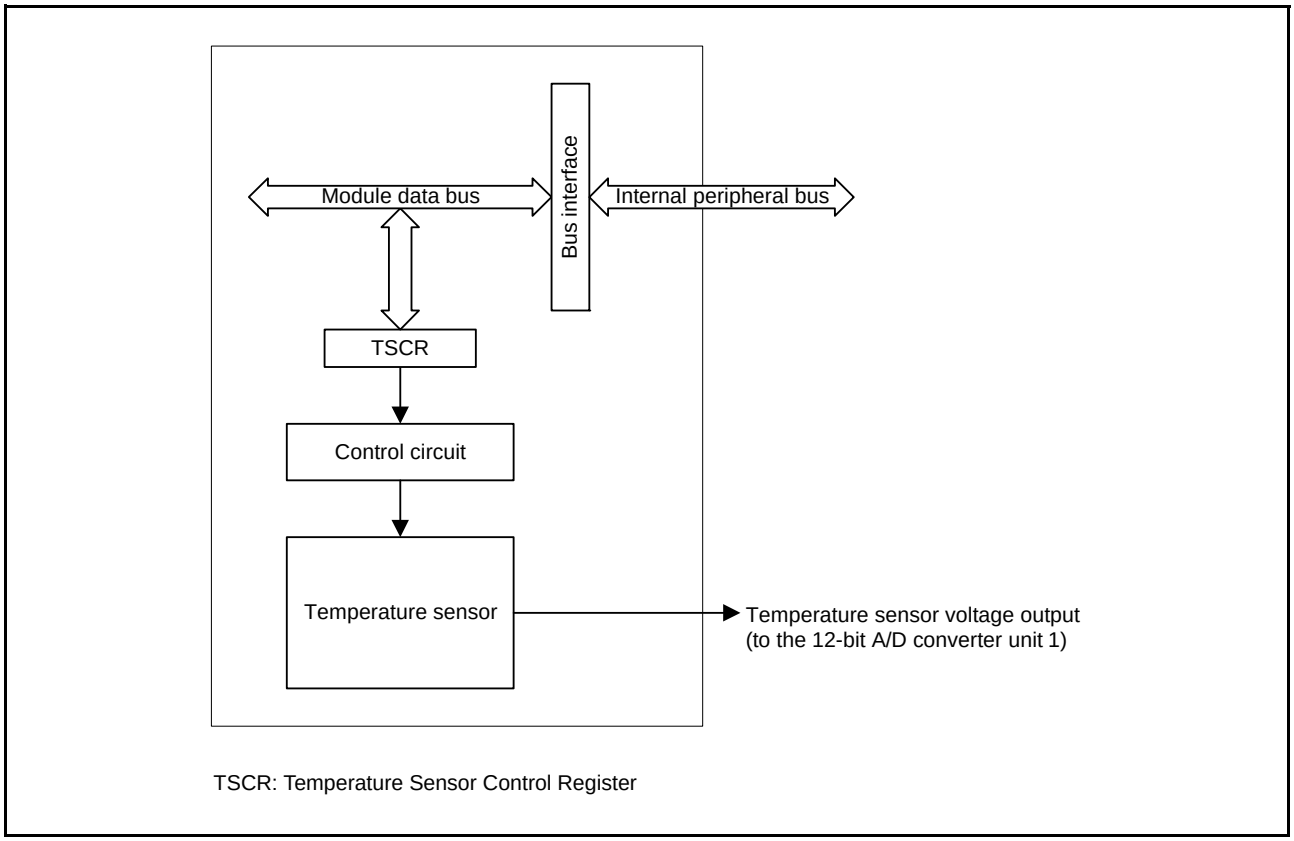


Figure 59.1 Block Diagram of Temperature Sensor

59.2 Register Descriptions

59.2.1 Temperature Sensor Control Register (TSCR)

Address(es): 0008 C500h

	b7	b6	b5	b4	b3	b2	b1	b0
	TSEN	—	—	TSOE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	TSOE	Temperature Sensor Output Enable	0: Disables output from the temperature sensor to the 12-bit A/D converter unit 1. 1: Enables output from the temperature sensor to the 12-bit A/D converter unit 1.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TSEN	Temperature Sensor Enable	0: Stops the temperature sensor. 1: Starts the temperature sensor.	R/W

The settings of TSCR register have the timing restrictions shown in Figure 59.4.

59.2.2 Temperature Sensor Calibration Data Register (TSCDR)

Address(es): 007F B17Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—												
Value after reset:	0	0	0	0	Unique value for each chip											

The TSCDR register stores temperature sensor calibration data measured for each chip at factory shipment. Temperature sensor calibration data is a digital value obtained using the 12-bit A/D converter unit 1 to convert the voltage output by the temperature sensor under the condition Ta = Tj = 128°C and AVCC1 = 3.3 V.
The TSCDR register is a 32-bit read-only register and should be read in 32-bit units.

59.3 Using the Temperature Sensor

The temperature sensor outputs a voltage which varies with the temperature.
This voltage is converted to a digital value by the 12-bit A/D converter unit 1. The temperature around the MCU can be obtained by converting the value into the temperature.

59.3.1 Preparation for Using the Temperature Sensor

Perform a calibration of the temperature sensor as shown below. The voltage output by the temperature sensor is proportional to temperature, which can be calculated according to the following formula.

Formula for the temperature characteristic:
T = (Vs – V1)/Slope + T1

T: Measured temperature (°C)
Vs: Voltage output by the temperature sensor when the temperature is measured (V)
T1: Sample temperature measurement at first point (°C)
V1: Voltage output by the temperature sensor when T1 is measured (V)
T2: Sample temperature measurement at second point (°C)
V2: Voltage output by the temperature sensor when T2 is measured (V)
Slope: Temperature slope of the temperature sensor (V/°C); Slope = (V2 – V1)/(T2 – T1)

Characteristics of the temperature sensor vary from MCU to MCU. Therefore, a two-point calibration (the following experimental measurement at two different temperatures) is recommended.
Use the 12-bit A/D converter unit 1 to measure the voltage V1 output by the temperature sensor at temperature T1. Again, using the 12-bit A/D converter unit 1, measure the voltage V2 output by the temperature sensor at a different temperature T2. Obtain the temperature slope (Slope = (V2 – V1)/(T2 – T1)) from these results.
Subsequently, obtain temperatures by substituting the slope into the formula for the temperature characteristic (T = (Vs – V1)/Slope + T1).

If you are using the temperature slope given in Table 64.49 of section 64, Electrical Characteristics, use the 12-bit A/D converter unit 1 to measure the voltage V1 output by the temperature sensor at temperature T1, and then calculate the temperature characteristic by using the formula below.
However, this calibration gives less accurate temperatures than two-point calibration.

T = (Vs – V1)/Slope + T1

T: Measured temperature (°C)
Vs: Voltage output by the temperature sensor when the temperature is measured (V)
T1: Sample temperature measurement at first point (°C)
V1: Voltage output by the temperature sensor when T1 is measured (V)
Slope: Temperature slope given in Table 64.49 ÷ 1000 (V/°C)

In this MCU, the TSCDR register stores the temperature value (CAL_{128}) of the temperature sensor measured under the condition Ta = Tj = 128°C and AVCC1 = 3.3 V. By using this value as the sample measurement result at the first point, preparation before using the temperature sensor can be omitted.
If V1 is calculated from CAL_{128},

$V1 = 3.3 \times CAL_{128}/4096$ [V]
Using this, the measured temperature can be calculated according to the formula below.

T = (Vs - V1)/Slope + 128 [°C]

T: Measured temperature (°C)
Vs: Voltage output by the temperature sensor when the temperature is measured (V)
V1: Voltage output by the temperature sensor when Ta = Tj = 128°C and AVCC1 = 3.3 V (V)
Slope: Temperature slope given in Table 64.49 ÷ 1000 (V/°C)

Error in the measured temperature (variation range is 3σ) is shown in Figure 59.2.

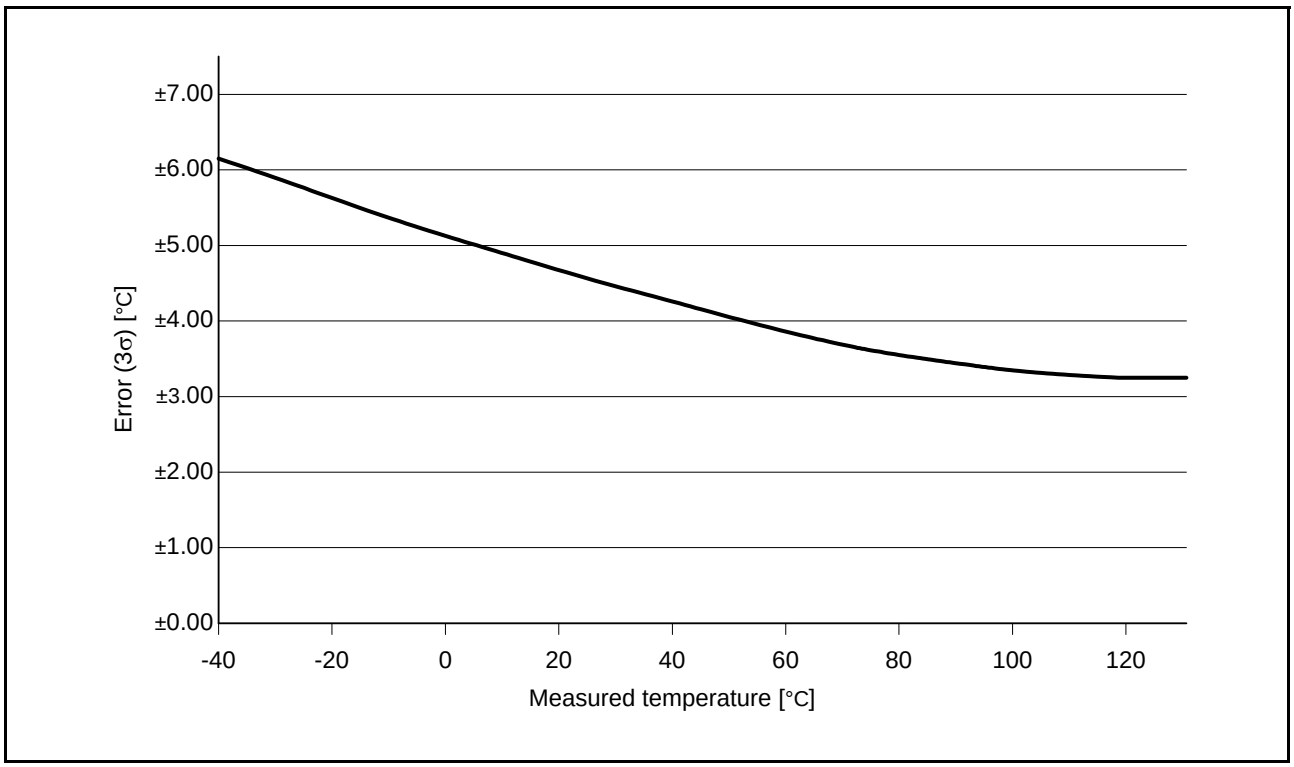


Figure 59.2 Error in the Measured Temperature

59.3.2 Setting of 12-Bit A/D Converter unit 1

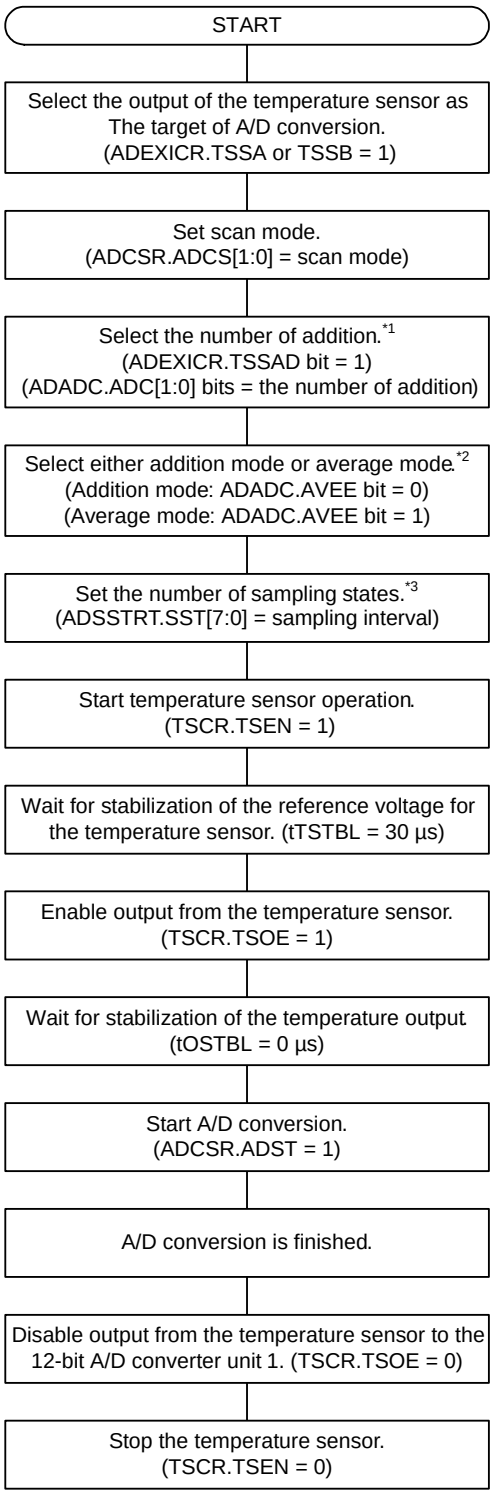
For A/D conversion of temperature sensor output voltages, 12-bit A/D converter unit 1 registers should be set as follows.

- Selecting the Temperature Sensor Voltage as an A/D Conversion Target
  Select A/D conversion of the voltage from the temperature sensor by setting the temperature sensor output A/D conversion select bit in the A/D conversion extended input control register (ADEXICR.TSSA or TSSB) to 1.
- Setting Scan Mode
  Select scan mode by setting the scan mode select bits in the A/D control register (ADCSR.ADCS[1:0]).
- Setting Addition/Average Mode
  For A/D conversion of the temperature sensor output, additional or average mode is selectable. To use either additional or average mode, set the temperature sensor output A/D converted value addition mode select bit in the A/D conversion extended input control register (ADEXICR.TSSAD) to 1, and the addition count select bits in the A/D converted value addition count select register (ADADC.ADC[1:0]) to the desired number of addition. Furthermore, clear the AVEE bit in ADADC to 0 to select addition mode; set the AVEE bit in ADADC to 1 to select average mode. In average mode, however, the ADC[1:0] bits in ADADC should not be set to 10b.
- Setting the Number of Sampling States of the 12-bit A/D converter unit 1
  The number of states for sampling of the output of the temperature sensor for A/D conversion is selectable. The initial setting is for 11 states. To change the number of states for sampling from 11 states, set the sampling time setting bits in A/D sampling state register T (ADSSTRT.SST[7:0]), when the ADST bit in ADCSR is 0.

Setting the A/D conversion start bit in the A/D control register (ADCSR.ADST) to 1 starts A/D conversion, and the result is stored in the A/D temperature sensor data register (ADTSDR). If you will be using A/D conversion of the output from the temperature sensor, do so in accord with section 59.3.3, Procedure for Using the Temperature Sensor.

59.3.3 Procedure for Using the Temperature Sensor

Figure 59.3 shows the procedure for using the temperature sensor.



Note 1. Unless addition/average mode is set, this setting is not required. To select average mode, do not set the ADC[1:0] bits to 10b.
Note 2. Unless addition/average mode is set, this setting is not required.
Note 3. The initial value for the number of states over which sampling proceeds is 11.
If the number of states is not to be changed, this setting is not required.

Figure 59.3 Procedure for Using the Temperature Sensor

59.3.4 Timing of A/D Conversion of Temperature Sensor Output

Figure 59.4 shows the timing from the start of temperature-sensor operation until the completion of A/D conversion when only the output from the temperature sensor is to be A/D converted and conversion is in single-scan mode. The times shown in the figure are described in Table 59.2.

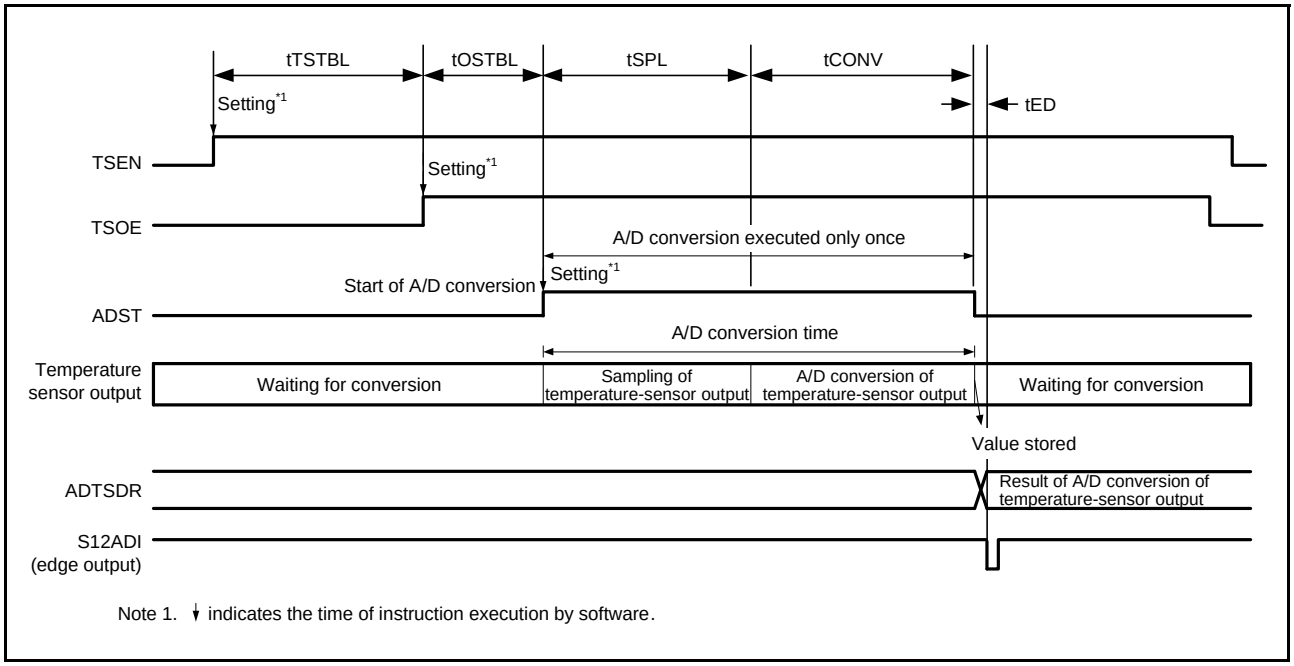


Figure 59.4 Timing from the Start of Temperature-Sensor Operation until Completion of A/D Conversion

Table 59.2 Time until Completion of A/D Conversion after the Start of Temperature-Sensor Operation

Item	Symbol	Time
Temperature-sensor reference-voltage stabilization wait time	tTSTBL	30 µs (min)
Temperature-sensor output stabilization wait time	tOSTBL	0 µs (min)
12-bit A/D converter unit 1 input sampling time	tSPL	ADSSTRT setting × ADCLK cycles
A/D conversion time	tCONV	Refer to Table 57.11, Times for Conversion during Scanning (in Numbers of Cycles of the ADCLK and PCLKB) in section 57.3.7, Analog Input Sampling and Scan Conversion Time.
Scan conversion end delay time	tED	Refer to Table 57.11, Times for Conversion during Scanning (in Numbers of Cycles of the ADCLK and PCLKB) in section 57.3.7, Analog Input Sampling and Scan Conversion Time.

59.4 Usage Note

59.4.1 Module-Stop Function Setting

The corresponding bit in module stop control register B (MSTPCRB) can be used to enable and disable the temperature sensor. The initial setting is for the temperature sensor to be stopped. The register becomes accessible on release from the module-stop state. For details, see section 11, Low Power Consumption.

60. Data Operation Circuit (DOC)

60.1 Overview

The data operation circuit (DOC) is used to compare, add, and subtract 16-bit data.
Table 60.1 lists the data operation circuit specifications and Figure 60.1 shows a block diagram of the data operation circuit.
16-bit data is compared and an interrupt can be generated when a selected condition applies.

Table 60.1 DOC Specifications

Item	Description
Data operation function	16-bit data comparison, addition, and subtraction
Lower power consumption function	Module stop state can be set.
Interrupts	An interrupt occurs at the following timings: • The compared values either match or mismatch • The result of data addition is greater than FFFFh • The result of data subtraction is less than 0000h
Event link function (output)	An interrupt occurs at the following timings: • The compared values either match or mismatch • The result of data addition is greater than FFFFh • The result of data subtraction is less than 0000h

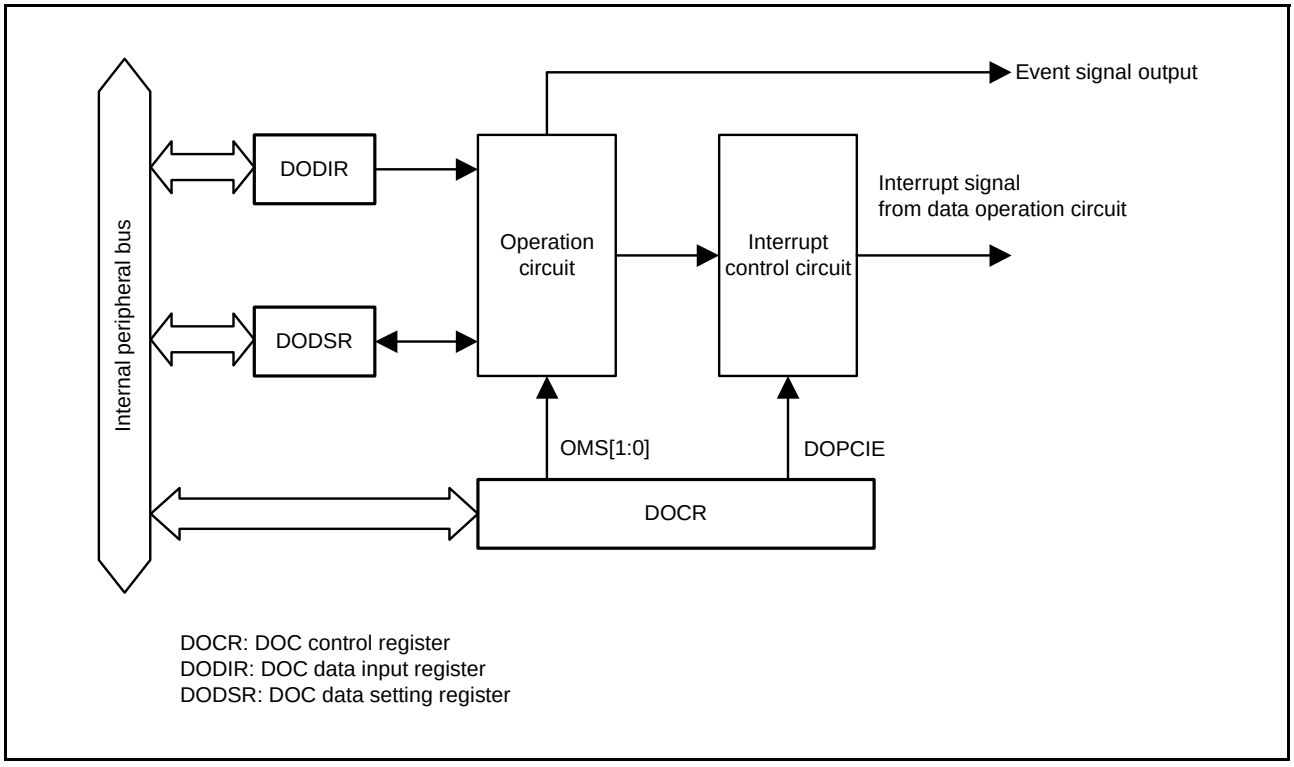


Figure 60.1 DOC Block Diagram

60.2 Register Descriptions

60.2.1 DOC Control Register (DOCR)

Address(es): 0008 B080h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DOPCFCL	DOPCF	DOPCIE	—	DCSEL	OMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OMS[1:0]	Operating Mode Select	b1 b0 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited	R/W
b2	DCSEL*1	Detection Condition Select	Result of data comparison 0: Data mismatch is detected. 1: Data match is detected.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	DOPCIE	Data Operation Circuit Interrupt Enable	0: Disables interrupts from the data operation circuit. 1: Enables interrupts from the data operation circuit.	R/W
b5	DOPCF	Data Operation Circuit Flag	Indicates the result of an operation.	R
b6	DOPCFCL	DOPCF Clear	0: Maintains the DOPCF flag state. 1: Clears the DOPCF flag.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Valid only when data comparison mode is selected.

OMS[1:0] Bits (Operating Mode Select)

These bits select the operating mode of the data operation circuit.

DCSEL Bit (Detection Condition Select)

This bit is valid only when data comparison mode is selected.

This bit selects the condition for detection in data comparison mode.

DOPCIE Bit (Data Operation Circuit Interrupt Enable)

Setting this bit to 1 enables interrupts from the data operation circuit.

DOPCF Flag (Data Operation Circuit Flag)

[Setting conditions]

- The condition selected by the DCSEL bit is met
- A result of data addition is greater than FFFFh
- A result of data subtraction is less than 0000h

[Clearing condition]

- Writing 1 to the DOPCFCL bit

DOPCFCL Bit (DOPCF Clear)

Setting this bit to 1 clears the DOPCF flag.

This bit is read as 0.

60.2.2 DOC Data Input Register (DODIR)

Address(es): 0008 B082h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DODIR is a 16-bit readable/writable register in which 16-bit data for use in the operations are stored.

60.2.3 DOC Data Setting Register (DODSR)

Address(es): 0008 B084h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DODSR is a 16-bit readable/writable register. This register stores 16-bit data for use as a reference in data comparison mode. This register also stores the results of operations in data addition and data subtraction modes.

60.3 Operation

60.3.1 Data Comparison Mode

Figure 60.2 shows an example of the steps involved in data comparison mode operation by the data operation circuit. The following is an example of operation when DCSEL is set to 0 (data mismatch is detected as a result of data comparison).

(1) Writing 00b to the DOCR.OMS[1:0] bits selects data comparison mode.
(2) The 16-bit reference data is set in DODSR.
(3) 16-bit data for comparison is written to DODIR.
(4) Writing of 16-bit data continues until all data for comparison have been written to DODIR.
(5) If a value written to DODIR does not match that in DODSR*1, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

Note 1. When DOCR.DCSEL = 0

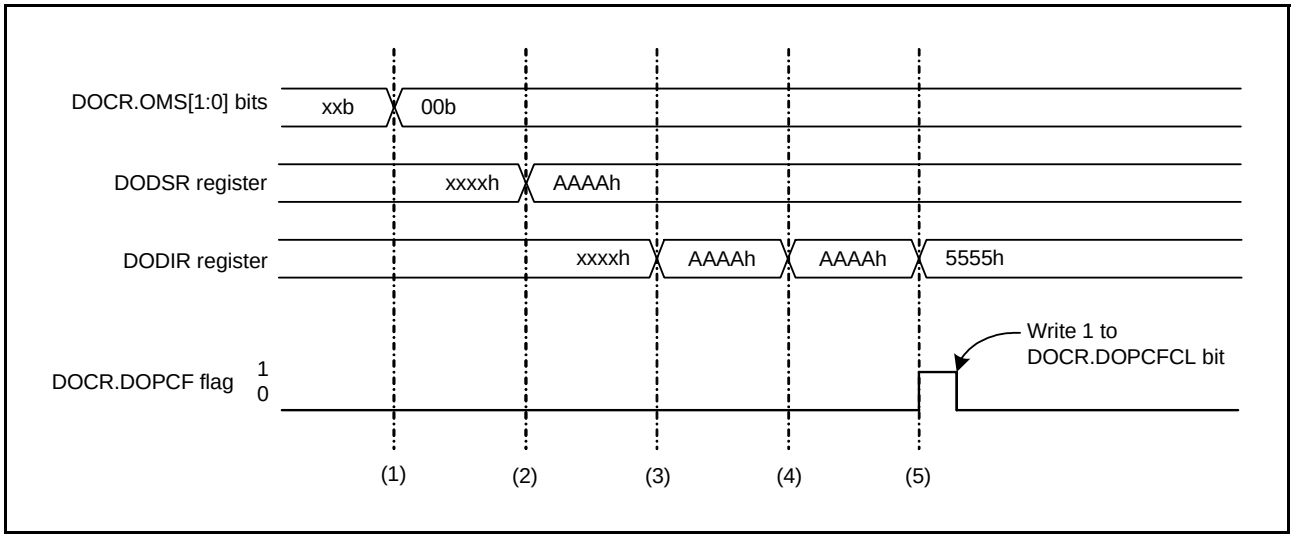


Figure 60.2 Example of Operation in Data Comparison Mode

60.3.2 Data Addition Mode

Figure 60.3 shows an example of the steps involved in data addition mode operation by the data operation circuit.

(1) Writing 01b to the DOCR.OMS[1:0] bits selects data addition mode.
(2) 16-bit data is set in the DODSR register as the initial value.
(3) 16-bit data to be added is written to DODIR. The result of the operation is stored in DODSR.
(4) Writing of 16-bit data continues until all data for addition have been written to DODIR.
(5) If the result of an operation is greater than FFFFh, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

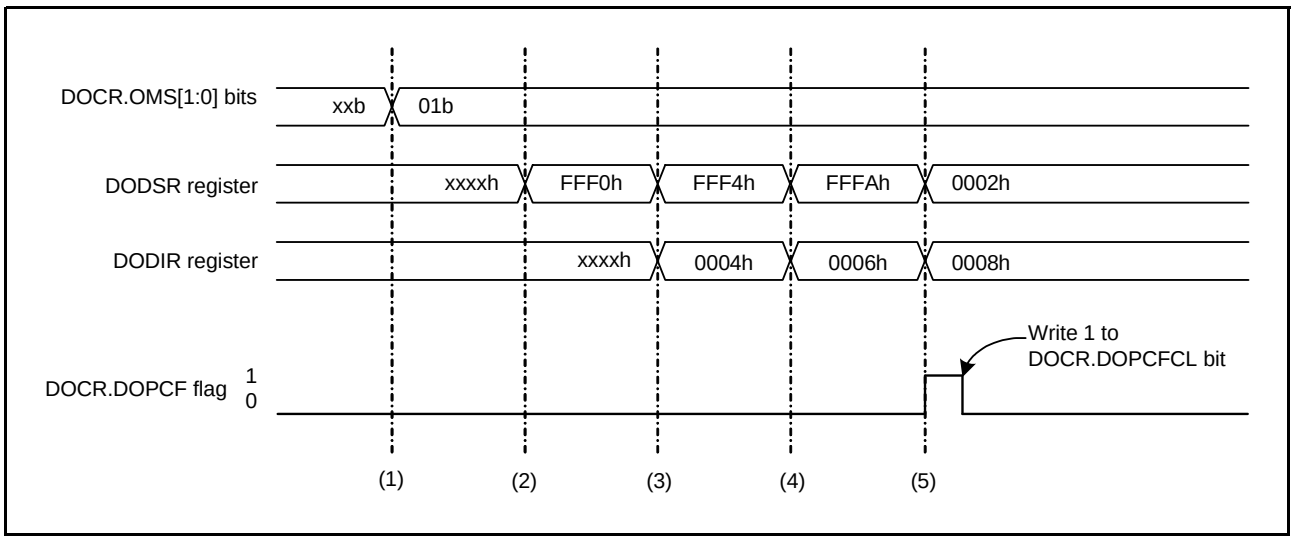


Figure 60.3 Example of Operation in Data Addition Mode

60.3.3 Data Subtraction Mode

Figure 60.4 shows an example of the steps involved in data subtraction mode operation by the data operation circuit.

(1) Writing 10b to the DOCR.OMS[1:0] bits selects data subtraction mode.
(2) 16-bit data is set in the DODSR register as the initial value.
(3) 16-bit data to be subtracted is written to DODIR. The result of the operation is stored in DODSR.
(4) Writing of 16-bit data continues until all data for subtraction have been written to DODIR.
(5) If the result of an operation is less than 0000h, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

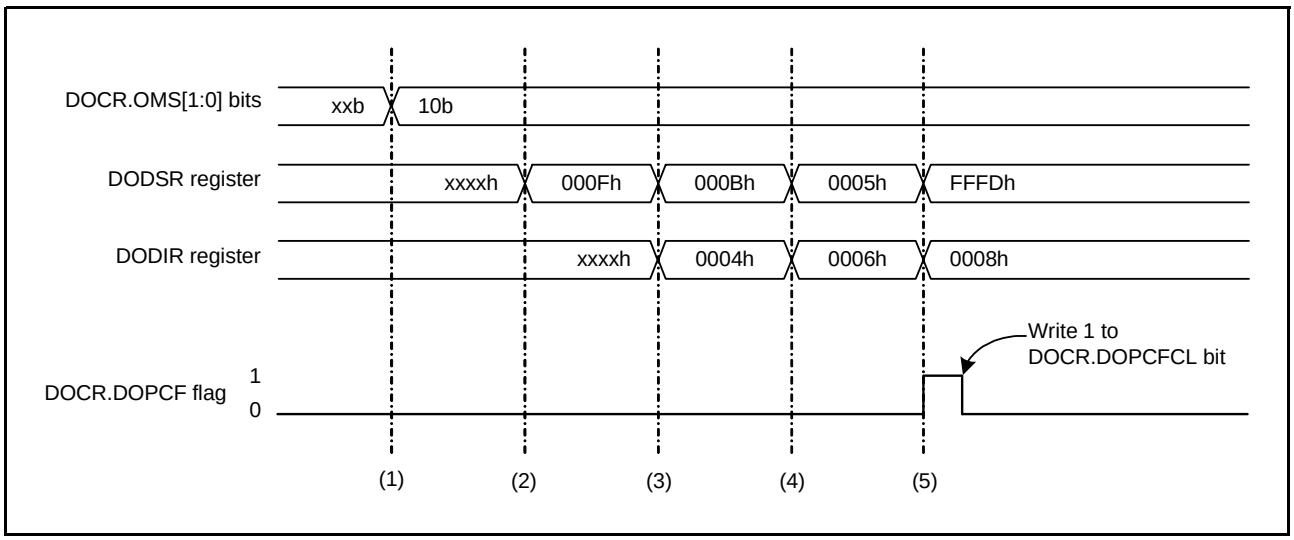


Figure 60.4 Example of Operation in Data Subtraction Mode

60.4 Interrupt Requests

The data operation circuit generates the data operation circuit interrupt as an interrupt request. When an interrupt source is generated, the data operation circuit flag corresponding to the interrupt is set to 1. Table 60.2 describes the interrupt request.

Table 60.2 Interrupt Request from Data Operation Circuit

Interrupt Request	Data Operation Circuit Flag	Interrupt Generation Timing
Data operation circuit interrupt	DOPCF	• The compared values either match or mismatch • The result of data addition is greater than FFFFh • The result of data subtraction is less than 0000h

60.5 Event Link Output

The DOC outputs event signals for the event link controller (ELC) under the following conditions, and these can be used to initiate operations by other modules selected in advance.

- The compared values either match or mismatch
- The result of data addition is greater than FFFFh
- The result of data subtraction is less than 0000h

60.5.1 Interrupt Handling and Event Linking

The DOC has a bit to enable or disable interrupts. An interrupt request signal is output for the CPU when an interrupt source is generated while the corresponding enable bit is enabled.
In contrast, an event link output signal is sent to other modules as an event signal via the ELC when an interrupt source is generated, regardless of the setting of the corresponding interrupt enable bit.

60.6 Usage Note

60.6.1 Module Stop Function Setting

Operation of the data operation circuit can be disabled or enabled using module stop control register B (MSTPCRB). The initial setting is for the data operation circuit to be stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

61. RAM

This MCU has an on-chip high-speed static RAM (without ECC error correction) and on-chip high-speed static RAM (with ECC error correction).

61.1 Overview

Table 61.1 lists the specifications of the RAM.

Table 61.1 Specifications of RAM

Item	Without ECC Error Correction	With ECC Error Correction (ECCRAM)
RAM capacity	512 Kbytes (RAM: 512 Kbytes)	32 Kbytes
RAM address	RAM0: 0000 0000h to 0007 FFFFh	ECCRAM: 00FF 8000h to 00FF FFFFh
Memory bus	Memory bus 1	Memory bus 3 (ECCRAM)
Access	• Single-cycle access is possible for both reading and writing.*1 • Enabling or disabling of the RAM is selectable.*2	• Enabling or disabling of the ECCRAM is selectable.*3 • The ECC function is disabled: Access is done in two cycles for both reading and writing.*1 • The ECC function is enabled (when no error has occurred): Access is done in two cycles for both reading and writing.*1 • The ECC function is enabled (when an error has occurred): Access is done in three cycles for both reading and writing.*1
Data retention function	Not available in deep software standby mode	Not available in deep software standby mode
Low power consumption function	The module-stop state is selectable.	The module-stop state is selectable.
Error checking	• Detection of 1-bit errors • A non-maskable interrupt or interrupt is generated in response to an error.	• ECC Error Correction Correction of 1-bit errors and detection of 2-bit errors • A non-maskable interrupt or interrupt is generated in response to an error.

Note 1. When accessing across the 8-byte boundary, the number of cycles is doubled.
Note 2. Selectable by the RAME bit in SYSCR1. For details on SYSCR1, see section 3.2.4, System Control Register 1 (SYSCR1).
Note 3. Enabling and disabling of the ECCRAM can be selected through the setting of the SYSCR1.ECCRAME bit. For details on SYSCR1, see section 3.2.4, System Control Register 1 (SYSCR1).

61.2 Register Descriptions

61.2.1 ECCRAM Operating Mode Control Register (ECCRAMMODE)

Address(es): 0008 12C0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	RAMMOD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	RAMMOD[1:0]	RAM Operating Mode Select	b1 b0 0 0: ECCs are disabled. 0 1: Setting prohibited. 1 0: ECCs are enabled without error checking. 1 1: ECCs are enabled with error checking.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ECCRAMMODE register is write-protected by the ECCRAM protection register (ECCRAMPRCR). Before writing to the ECCRAMMODE register, set the corresponding bit in the ECCRAMPRCR register to 1 to enable writing to it. Be sure to read this register immediately after writing to it, and then compare the read and written values. If this operation is not performed, access to the RAM is not guaranteed.
Also, do write to this register while accessing to the RAM.

61.2.2 ECCRAM 2-Bit Error Status Register (ECCRAM2STS)

Address(es): 0008 12C1h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ECC2ERR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ECC2ERR	2-Bit ECC Error Status Flag	0: A 2-bit ECC error has not occurred. 1: A 2-bit ECC error has occurred.	R/(W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to clear the flag.

When the ECCRAMMODE.RAMMOD[1:0] bits are set to 11b, the ECC2ERR flag is set to 1 if a 2-bit error is found. The RAM error interrupt is also generated at this time. Writing of 0 to the ECC2ERR flag leads to clearing of the RAM error interrupt having the 2-bit ECC error as its source.

61.2.3 ECCRAM 1-Bit Error Information Update Enable Register (ECCRAM1STSEN)

Address(es): 0008 12C2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ECC1STSEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ECC1STSEN	1-Bit ECC Error Information Update Enable	0: Disables updating of the 1-bit ECC error information. 1: Enables updating of the 1-bit ECC error information.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ECCRAM1STSEN register enables or disables updating of the ECCRAM 1-bit error status register (ECCRAM1STS) in response to a 1-bit error in the ECCRAM.
The ECCRAMPRCR register can protect this register against writing. Accordingly, change the effective bit in the ECCRAMPRCR register to the enabled setting before attempting to write to the ECCRAM1STSEN register.
Always be sure to read this register immediately after writing to it, and then compare the read and written values. Do not attempt access to the RAM during this process. Also, do not write to this register while accessing to the RAM.

61.2.4 ECCRAM 1-Bit Error Status Register (ECCRAM1STS)

Address(es): 0008 12C3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ECC1ERR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ECC1ERR	1-Bit ECC Error Status	0: A 1-bit ECC error has not occurred. 1: A 1-bit ECC error has occurred.	R/(W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to clear the bit.

When ECCs are enabled, error checking is selected, and updating of the 1-bit error information is enabled, the ECC1ERR bit is set to 1 if a 1-bit error is found. The RAM error interrupt is also generated at this time. Writing of 0 to the ECC1ERR bit leads to clearing of the RAM error interrupt having the 1-bit ECC error as its source.

61.2.5 ECCRAM Protection Register (ECCRAMPRCR)

Address(es): 0008 12C4h

	b7	b6	b5	b4	b3	b2	b1	b0
	KW[6:0]							PRCR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRCR	ECCRAMMODE and ECCRAM1STSEN Registers Write Control	0: Disables writing to the ECCRAMMODE and ECCRAM1STSEN registers. 1: Enables writing to the ECCRAMMODE and ECCRAM1STSEN registers.	R/W
b7 to b1	KW[6:0]	Write Key Word	Enables or disables the rewriting of the ECCRAMPRCR register. When rewriting the ECCRAMPRCR register, write 1111000b to the KW[6:0] bits.	R/W

Writing 1 to the PRCR bit is possible when KW[6:0] = 1111000b.
Otherwise writing to PRCR clears the bit to 0. KW[6:0] is read as 0000000b.
The targets for write protection by the ECCRAMPRCR register are the ECCRAMMODE and ECCRAM1STSEN registers. Once the PRCR bit is set to 1, writing to ECCRAMMODE and ECCRAM1STSEN registers is enabled until the PRCR bit is cleared to 0. Clear the PRCR bit to 0 after writing to ECCRAMMODE and ECCRAM1STSEN registers.

61.2.6 ECCRAM 2-Bit Error Address Capture Register (ECCRAM2ECAD)

Address(es): 0008 12C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	ECC2EAD												—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0.	R
b14 to b3	ECC2EAD	2-Bit ECC Error Address	The address where a 2-bit ECC error was found is read.	R
b23 to b15	—	Reserved	These bits are read as 1.	R
b31 to b24	—	Reserved	These bits are read as 0.	R

When the ECCRAMMODE.RAMMOD[1:0] bits are set to 11b, this register will hold the address where a 2-bit ECC error was found. Specifically, the address of the 8-byte boundary below the location where the error was found is written to the ECC2EAD bits of this register at the same time as the ECCRAM2STS.ECC2ERR flag is set. The address is not updated whenever the ECC2ERR flag is set to 1 to indicate an error. Its value does not change if ECCs are disabled. The value is initialized by a reset, and this is the only condition that leads to clearing of the register.

61.2.7 ECCRAM 1-Bit Error Address Capture Register (ECCRAM1ECAD)

Address(es): 0008 12CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	ECC1EAD												—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0.	R
b14 to b3	ECC1EAD	1-Bit ECC Error Address	The address where a 1-bit ECC error was found is read.	R
b23 to b15	—	Reserved	These bits are read as 1.	R
b31 to b24	—	Reserved	These bits are read as 0.	R

When ECCs are enabled, error checking is selected, and updating of the 1-bit error information is enabled, this register will hold the address where a 1-bit error was found. Specifically, the address of the 8-byte boundary below the location where the error was found is written to the effective bits of this register at the same time as the ECCRAM1STS.ECC1ERR bit is set.

The address is not updated whenever the ECC1ERR bit is set to 1 to indicate an error. Its value does not change if ECCs are disabled. The value is initialized by a reset, and this is the only condition that leads to clearing of the register.

61.2.8 ECCRAM Protection Register 2 (ECCRAMPRCR2)

Address(es): 0008 12D0h

	b7	b6	b5	b4	b3	b2	b1	b0
	KW2[6:0]							PRCR2
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRCR2	ECCRAMETST Register Write Control	0: Writing to ECCRAMETST is disabled. 1: Writing to ECCRAMETST is enabled.	R/W
b7 to b1	KW2[6:0]	Write Keyword	Enables or disables the rewriting of the ECCRAMPRCR2 register. When rewriting the ECCRAMPRCR2 register, write 1111000b to the KW2[6:0] bits.	R/W

When KW2[6:0] = 1111000b, writing 1 to the PRCR2 bit is enabled.

Otherwise writing to PRCR2 clears the bit to 0. The KW2[6:0] bits are read as 0000000b.

The target for write protection by the PRCR2 bit is the ECCRAM test control register (ECCRAMETST). Once the PRCR2 bit is set to 1, writing to the ECCRAMETST register is enabled until the PRCR2 bit is cleared to 0. After writing to the ECCRAMETST register, clear the PRCR2 bit to 0.

61.2.9 ECCRAM Test Control Register (ECCRAMETST)

Address(es): 0008 12D4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TSTBYP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSTBYP	ECC Bypass Select	0: ECC bypass is disabled. 1: ECC bypass is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The protection register 2 (ECCRAMPRCR2) protects this register against writing. Accordingly, change the effective bit in the ECCRAMPRCR2 register to the enabled setting before attempting to write to the ECCRAMETST register. Do not write to ECCRAMETST while access to RAM is in progress.
Eight of the 72 bits of data in the RAM are used for the ECC. When the TSTBYP bit is set to 1 to select bypassing of the ECC circuit, the 8 bits for the ECC become directly accessible. Setting the TSTBYP bit for bypassing of the ECC is only possible when the ECC is disabled by setting the ECCRAMMODE.RAMMOD[1:0] bit to 00b.

61.2.10 RAM Operating Mode Control Register (RAMMODE)

Address(es): 0008 1200h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	RAMMODE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	RAMMODE[1:0]	RAM Operating Mode Select	b1 b0 0 0: Parity checking is disabled. 0 1: Parity checking is enabled. Settings other than above are prohibited.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RAMMODE register is write-protected by the RAM protection register (RAMPRCR). Before writing to the RAMMODE register, set the RAMPRCR.RAMPRCR bit to 1 to enable writing to it. Set the RAMMODE register before starting access to the RAM. If this register is modified after accessing to the RAM, RAM operation is not guaranteed.

61.2.11 RAM Error Status Register (RAMSTS)

Address(es): 0008 1201h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RAMERR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RAMERR	RAM Error Status Flag	0: A parity check error has not occurred. 1: A parity check error has occurred.	R/(W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to clear the flag.

When parity checking is enabled, the RAMERR flag is set to 1 if a parity check error is detected. The RAM error interrupt request is also generated at this time.
When parity checking is disabled, the RAMERR flag is not set to 1 because no parity check error is detected.
Writing 0 to the RAMERR flag clears the RAM error interrupt request corresponding to the parity check error.

61.2.12 RAM Error Address Capture Register (RAMECAD)

Address(es): 0008 1208h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	READ		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

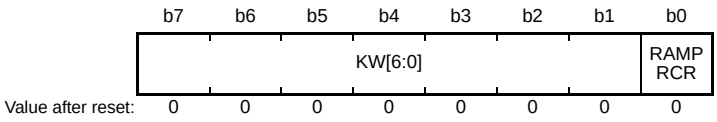
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	READ													—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0.	R
b18 to b3	READ	Error Address	The address where an error is found is read.	R
b31 to b19	—	Reserved	These bits are read as 0.	R

When parity checking is enabled, this register will hold the address where a parity check error was found.
The address of the 8-byte boundary below the location where the error was found is stored in this register at the same time the RAMSTS.RAMERR flag is set to 1.
The error address is not updated when the RAMERR flag is 1 (error has occurred). Its value does not change when parity checking is disabled because no parity check error is detected.
The RAMECAD register is initialized only by a reset.

61.2.13 RAM Protection Register (RAMPRCR)

Address(es): 0008 1204h



Bit	Symbol	Bit Name	Description	R/W
b0	RAMPRCR	RAMMODE Register Write Control	0: Disables writing to the RAMMODE register. 1: Enables writing to the RAMMODE register.	R/W
b7 to b1	KW[6:0]	Write Key Word	Enables or disables the rewriting of the RAMPRCR register. When rewriting the RAMPRCR register, write 1111000b to the KW[6:0] bits.	R/W

Writing 1 to the RAMPRCR bit is possible when KW[6:0] = 1111000b. Otherwise writing to RAMPRCR clears the bit to 0. The value of KW[6:0] is read as 0000000b.

The targets for write protection by the RAMPRCR register is the RAM operating mode control register (RAMMODE). Once the RAMPRCR bit is set to 1, writing to RAMMODE register is enabled until the RAMPRCR bit is cleared to 0. Clear the RAMPRCR bit to 0 after writing to RAMMODE register.

61.3 Operation

61.3.1 Low Power Consumption Function

Power consumption can be reduced by setting module stop control register C (MSTPCRC) to stop supply of the clock signal to RAM.
When the MSTPC0 bit in MSTPCRC is set to 1, supply of the clock signal to RAM0 is stopped. When the MSTPC6 bit in MSTPCRC is set to 1, supply of the clock signal to the ECCRAM is stopped.
The RAM is thus placed in the module-stop state by stopping supply of the clock signals. The RAM operates after a reset.
RAM is not accessible if it is in the module-stop state. A transition to the module-stop state should not be made while access to RAM is in progress.
Access to the RAM in the module-stop state is prohibited. If access is attempted, correct operation is not guaranteed.
For details on the MSTPCRC register, see section 11, Low Power Consumption.

61.3.2 Correction of ECC Errors

Enabling and disabling of ECC error correction can be selected through the register setting. In the initial state, ECC error correction is disabled. The ECC used in this device is SEC-DED (Single-Error Correction and Double-Error Detection Code).
When ECCs are enabled, an 8-bit ECC code is appended to 64-bit data for writing. For reading, 72-bit (data: 64 bits, ECC code: 8 bits) data is read out from the RAM.
When ECCs are enabled and error checking is selected, error correction is done if a 1-bit error occurs and the ECCRAM1STS.ECC1ERR bit is set to 1 if the ECCRAM1STSEN.ECC1STSEN bit is 1. When an error is detected, the address of the 8-byte boundary including the location where the error was found is set to the ECCRAM1ECAD register.
If a 2-bit error occurs, error detection is done and the ECCRAM2STS.ECC2ERR flag is set to 1, though error correction is not performed. When an error is detected, the address of the 8-byte boundary including the location where the error was found is set to the ECCRAM2ECAD register. In addition, there is no way to confirm the location where the error was found within the 8-byte boundary. Therefore, when the correct data is written after the occurrence of an error, update all the data in the 8-byte boundary.
When ECCs are enabled but error checking is not selected, error correction is done if a 1-bit error occurs when reading; however, the corrected data is not written back to the RAM, and ECC codes are neither generated nor written back. In writing, write data is replaced and ECC codes are generated then written back.
Since the RAM data is undefined after power on and release from deep software standby mode, accessing the RAM when ECCs are enabled and error checking is selected causes an ECC error to occur. Therefore, when using ECC error correction, initial writing to the area to be used in the RAM should be done beforehand when ECCs are enabled but error checking is not selected. Since the ECC error correction is done in 8-byte units, the initial write must be done for the 8 × N byte area including the area to be used.

61.3.3 Parity Checking

Enabling and disabling of parity checking can be selected through the RAMMODE register setting. In the initial state, parity checking is disabled. Even parity checking is used in this device.
1-bit parity check code is added to each 1-byte data for writing, and the parity is checked for reading.
If a 1-bit error is detected in the 1 byte when the parity is checked for reading, a RAM error interrupt can be generated. If a 2-bit error or more is detected in the 1 byte, errors cannot be correctly detected.
After power-on, parity check code is undefined until data is written. To use parity checking, write the initial value to all areas while parity checking is enabled before accessing to the RAM immediately after a reset.
Operation cannot be guaranteed if access is made to an area where the initial value is not written.

61.3.4 RAM Error Interrupt Function

A RAM error interrupt is generated if any of the following bits changes to 1:
When ECCs are enabled and error checking is selected
- ECCRAM2STS.ECC2ERR bit indicating an ECC2 bit error
- ECCRAM1STS.ECC1ERR bit indicating an ECC1 bit error

When parity checking is enabled
- RAMSTS.RAMERR bit indicating a parity check error

Writing 0 to a bit that has been set clears the RAM error interrupt due to the corresponding source.
If the error from the other source is still occurring at this time, the RAM error interrupt will remain generated.
When the ECC 1-bit error interrupt is to be masked, set the ECCRAM1STSEN.ECC1STSEN bit to 0 to disable updating of the ECC1ERR bit. A RAM error interrupt will not be generated while ECCs are disabled and when ECCs are enabled but error checking is not selected.

61.3.5 ECC Decoder Testing

In this MCU, the user can test the ECC code control circuit for the ECCRAM.
Figure 61.1 shows the flow for testing of the ECC code control circuit.

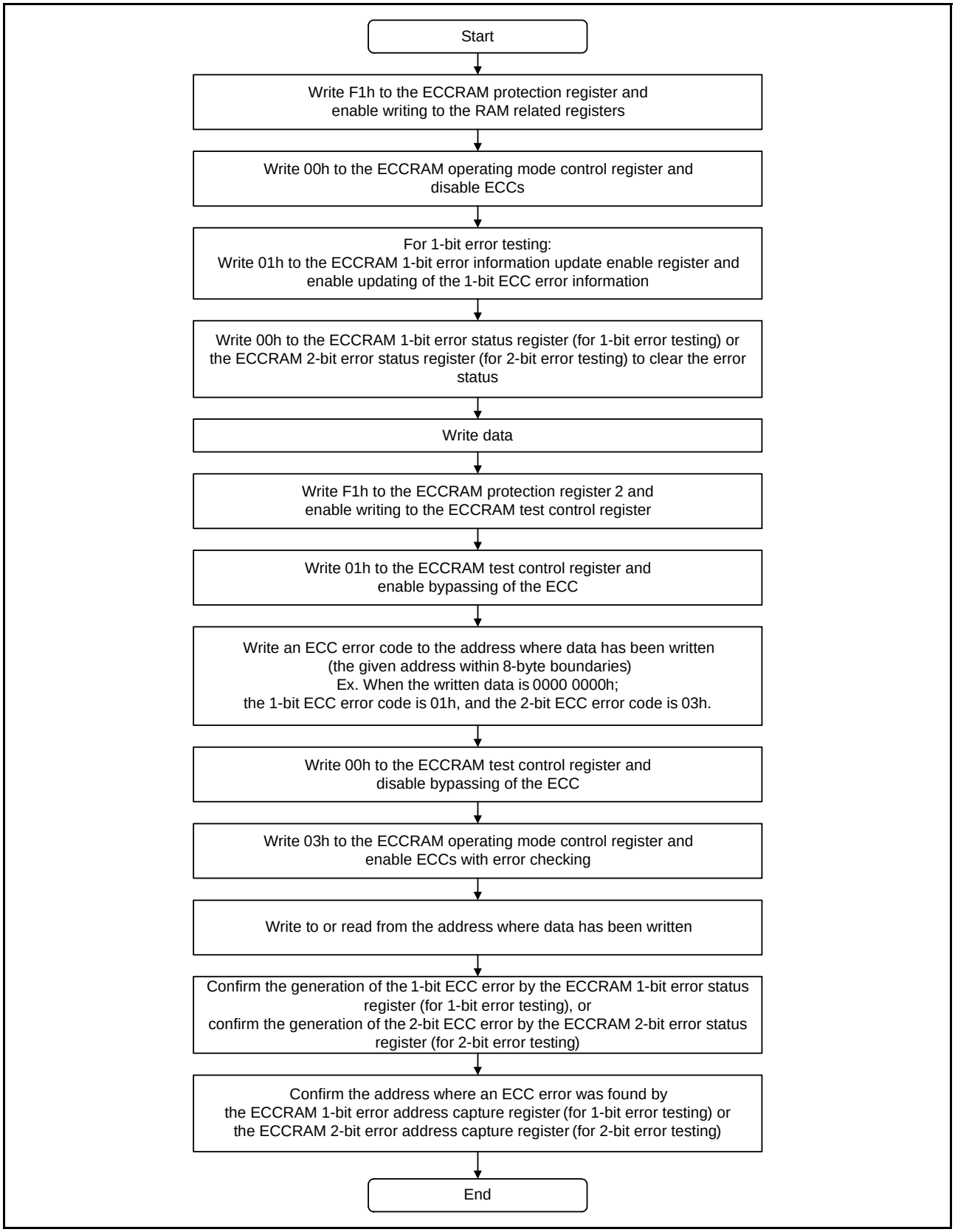


Figure 61.1 Flow for Testing of the ECC Code Control Circuit

61.3.6 Interrupt Source

The RAM has two interrupt sources; RAM error interrupt generated from the RAM when a parity check error occurred, or RAM error interrupt generated from the ECCRAM when an ECC error occurred, and can be used as a non-maskable interrupt or interrupt. For details, see section 15, Interrupt Controller (ICUA).

Table 61.2 RAM Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
RAMERR	ECC error	Not possible	Not possible

61.4 Usage Notes

61.4.1 Low Power Consumption Function

Power consumption can be reduced by setting module stop control register C (MSTPCRC) to stop supply of the clock signal to the RAM and ECCRAM.
Setting the MSTPCRC.MSTPC0 bit to 1 stops supply of the clock signal to the RAM. Setting the MSTPCRC.MSTPC6 bit to 1 stops supply of the clock signal to the ECCRAM.
Stopping supply of the clock signal places the RAM and ECCRAM individually in the module stop state.
The RAM and ECCRAM operate after a reset.
The RAM and ECCRAM are not accessible in the module stop state.
Do not allow transitions to the module stop state while accessing to the RAM or ECCRAM.
For details on the MSTPCRC register, refer to section 11, Low Power Consumption.

61.4.2 Notes on Using Error Checking of RAM0 and ECCRAM

When using RAM0 parity and ECCRAM ECCs for error checking to operate a program in the relevant RAM, initialize RAM0 and ECCRAM so that CPU can correctly prefetch data. If the CPU prefetches from a RAM area that is not initialized, a RAM error may occur.
The RAM0 and ECCRAM should be initialized at the 8-byte boundary. When the end address allocated in the RAM is below the 8-byte boundary, allocate a NOP instruction up to the boundary.

62. Standby RAM

This MCU provides an on-chip static RAM that can retain data in deep software standby (standby RAM).

62.1 Overview

Table 62.1 lists the specifications of the standby RAM.

Table 62.1 Specifications of Standby RAM

Item	Description
RAM capacity	8 Kbytes
RAM address	000A 4000h to 000A 5FFFh
Access	• Both read and write operations take 2 or 3 cycles of PCLKB when ICLK ≥ PCLKB; two cycles of ICLK are needed when ICLK < PCLKB. • Enabling or disabling of RAM access is selectable.*1 • Endian conforms to the endian setting of the chip. • Non-aligned access is prohibited. If non-aligned access is attempted, correct operation is not guaranteed.
Data retention function	Data can be retained in deep software standby mode
Low-power consumption function	The module-stop state is selectable.

Note 1. Selectable by the SBYRAME bit in SYSCR1. For details on SYSCR1, see section 3.2.4, System Control Register 1 (SYSCR1).

62.2 Operation

62.2.1 Data Retention

Whether or not the supply of internal power to the standby RAM continues in deep software standby mode is selectable by the DPSBYCR.DEEPCUT[1:0] bits.
If continuation of the supply of internal power is selected, data in the standby RAM are retained in deep software standby mode.
See section 11, Low Power Consumption, for details on the DPSBYCR.DEEPCUT[1:0] bits.

62.2.2 Low Power Consumption Function

Power consumption can be reduced by setting module stop control register C (MSTPCRC) to stop supply of the clock signal to the standby RAM.
If the MSTPC7 bit in MSTPCRC is set to 1, supply of the clock signal to the standby RAM is stopped.
The standby RAM is thus placed in the module-stop state by stopping supply of the clock signals. The standby RAM operates after a reset.
The standby RAM is not accessible if it is in the module-stop state. A transition to the module-stop state should not be made while access to the standby RAM is in progress.
For details on the MSTPCRC register, see section 11, Low Power Consumption.

63. Flash Memory

This MCU has a maximum 4-Mbyte code flash memory and a 64-Kbyte data flash memory.
The code flash memory and data flash memory are respectively used to hold instructions, operands, etc. and to hold data.

63.1 Overview

Table 63.1 lists the specifications of the code flash memory/data flash memory, and Figure 63.1 is a block diagram of the flash memory related modules.
The I/O pins used in boot mode, see Table 63.9.
The FCU (flash control unit) controls programming and erasure of the flash memory. The FCU RAM is a RAM to store the firmware (FCU firmware) to be executed by the FCU. The FACI (flash application command interface) controls the FCU according to the specified FACI commands.
Regarding the configuration of the code flash memory, see Figure 63.2, and for the configuration of the data flash memory, see Figure 63.3.

Table 63.1 Specifications of Code Flash Memory and Data Flash Memory

Item	Code Flash Memory	Data Flash Memory
Memory capacity	• User area: 4 Mbytes max. • User boot area: 32 Kbytes	Data area: 64 Kbytes
Read cycle	A high-speed read operation takes one cycle of ICLK	A read operation takes eight cycles of FCLK in word or byte access
Value after erasure	FFh	Undefined
Programming/erasing method	• Programming and erasing the code flash memory/data flash memory is handled by the FACI commands specified in the FACI command issuing area (007E 0000h). • Programming/erasure through transfer by a dedicated flash-memory programmer via a serial interface (serial programming) • Programming/erasure of flash memory by a user program (self-programming)	
Security function	Protects against illicit tampering with or reading out of data in flash memory	
Protection	Protects against erroneous rewriting of the flash memory	
Trusted memory (TM) function	Protects against illicit reading of blocks 8 and 9 in the code flash memory	
Background operations (BGOs)	The code flash memory can be read while the code flash memory is being programmed or erased.*1 The code flash memory can be read while the data flash memory is being programmed or erased.	
Units of programming and erasure	• Units of programming for the user area or user boot area: 256 bytes • Units of erasure for the user area: Block units	• Unit of programming for the data area: 4 bytes • Unit of erasure for the data area: 64 bytes
Other functions	Interrupts can be accepted during self-programming.	
	In the initial settings of this MCU, an expansion area of the option-setting memory can be set.	
On-board programming (four types)	Programming/erasure in boot mode (for the SCI interface) • The asynchronous serial interface (SCI1) is used. • The transfer rate is adjusted automatically. • The user boot area can also be programmed or erased. Programming/erasure in boot mode (for the USB interface) • USBb is used. • Dedicated hardware is not required, so direct connection to a PC is possible. Programming/erasure in user boot mode • Able to create original boot programs of the user's making. Programming/erasure by a routine for code flash memory/data flash memory programming within the user program • This allows code flash memory/data flash memory programming/erasure without resetting the system.	
Programming and Erasure by Dedicated Parallel Programmer	A flash programmer can be used to program or erase the user area and user boot area.	A flash programmer cannot be used to program or erase the data area.
Unique ID	A 12-byte ID code provided for each MCU	

Note 1. Limitations apply to the combinations of the address ranges for programming/erasure process and reading process: see Table 63.16.

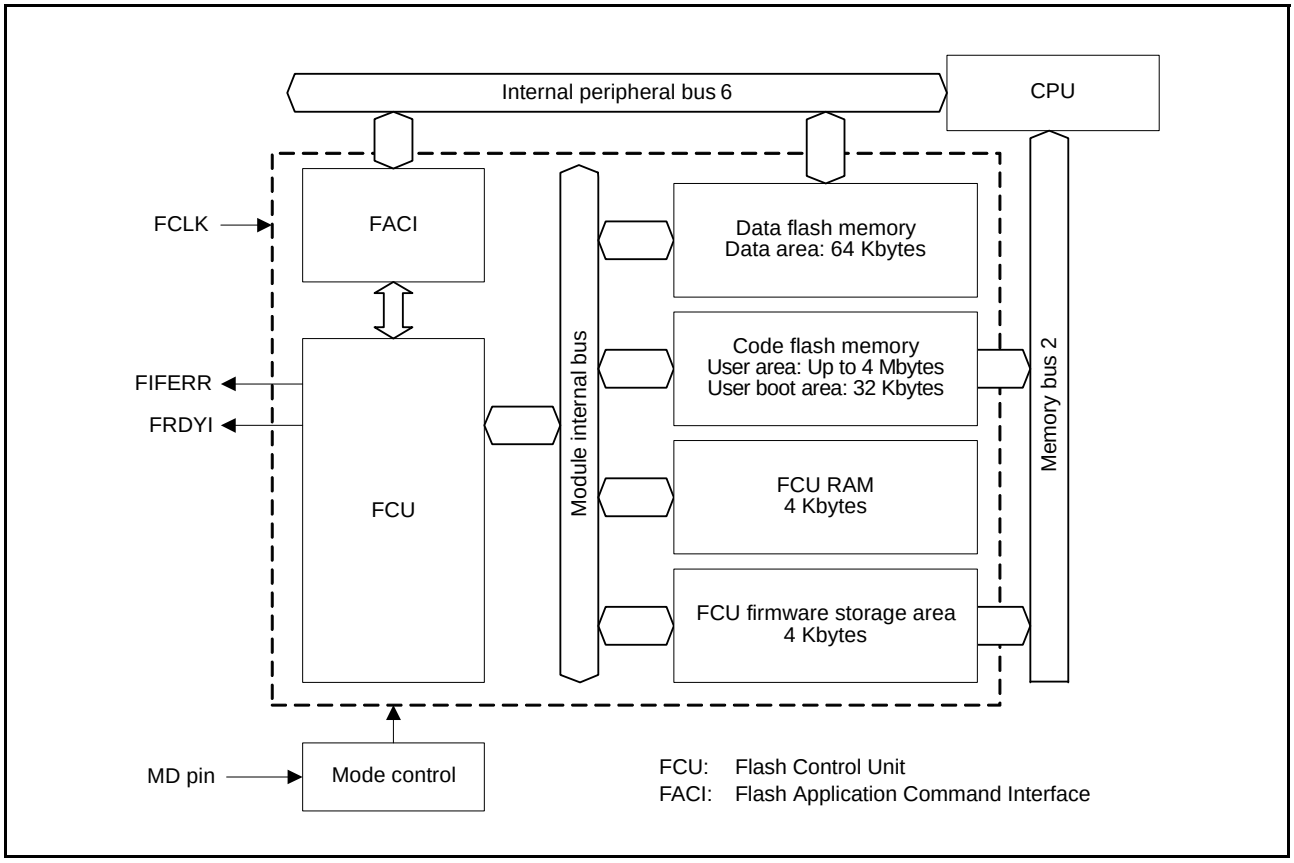
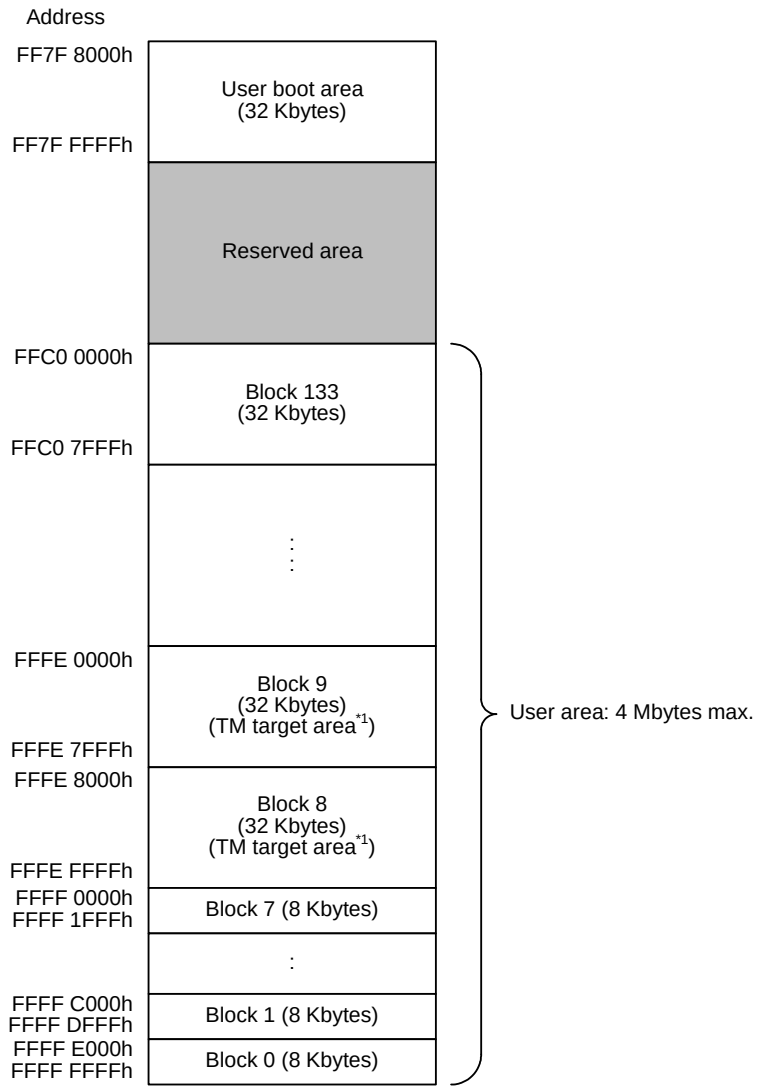


Figure 63.1 Block Diagram of Flash Memory Related Modules

63.2 Structure of Memory

Figure 63.2 illustrates the mapping of the code flash memory. The user area of the code flash memory in this MCU is divided into 8- and 32-Kbyte blocks, which serve as the units of erasure. When the TM function is enabled, blocks 8 and 9 are the TM target areas. A 32-Kbyte user boot area is also incorporated as a single block. The user area and user boot area are available as areas for storing the user program.
Furthermore, a 32-Kbyte user boot area, which is protected against programming by self-programming, is incorporated as a single block. This area is thus available as an area for storing boot programs, etc., for which rewriting while the user program is running has to be prohibited.



	Address	Number of Blocks
4.0-Mbyte product	FFC0 0000h to FFFF FFFFh	0 to 133
3.0-Mbyte product	FFD0 0000h to FFFF FFFFh	0 to 101
2.5-Mbyte product	FFD8 0000h to FFFF FFFFh	0 to 85
2.0-Mbyte product	FFE0 0000h to FFFF FFFFh	0 to 69

Note 1. Both blocks 8 and 9 are the target areas for handling as TM. Setting only block 8 or only block 9 is not allowed.

Figure 63.2 Mapping of the Code Flash Memory

The data area of the data flash memory in this MCU is divided into 64-byte blocks, with each being a unit for erasure. Figure 63.3 shows the mapping of the data flash memory.

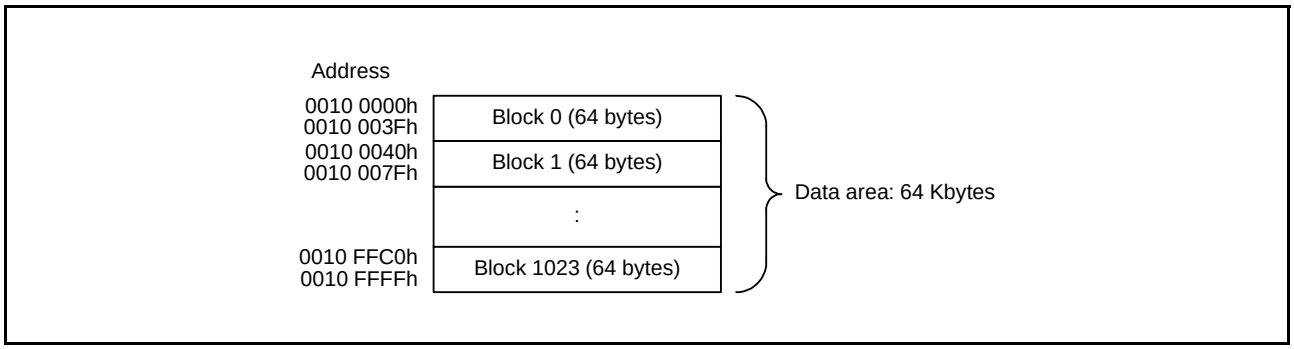


Figure 63.3 Mapping of the Data Flash Memory

63.3 Register Descriptions

There are registers for use in programming and erasure of the code flash, data flash, and option-setting memory. As for the registers for setting when the FACI is to be used, refer to Flash Memory User's Manual: Hardware Interface.

63.3.1 Flash P/E Protect Register (FWEPROR)

Address(es): 0008 C296h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	FLWE[1:0]	
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	FLWE[1:0]	Flash Programming and Erasure Enable	b1 b0 0 0: Disables programming and erasure, programming and erasure of lock bits, and blank checking. 0 1: Enables programming and erasure, programming and erasure of lock bits, and blank checking. 1 0: Disables programming and erasure, programming and erasure of lock bits, and blank checking. 1 1: Disables programming and erasure, programming and erasure of lock bits, and blank checking.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Programming and erasure of the code flash memory, data flash memory, or option-setting memory, programming and erasure of lock bits, and blank checking are enabled or disabled by hardware.
FWEPROR is initialized by a reset due to the signal on the RES# pin, a power-on reset, a voltage-monitoring 0 reset, an independent watchdog timer reset, a watchdog timer reset, a voltage-monitoring 1 reset, a voltage-monitoring 2 reset, and a software reset, and by transitions to software standby and deep software standby.

63.3.2 Unique ID Register n (UIDRn) (n = 0 to 2)

Address(es): UIDR0 007F B174h, UIDR1 007F B1E4h, UIDR2 007F B1E8h

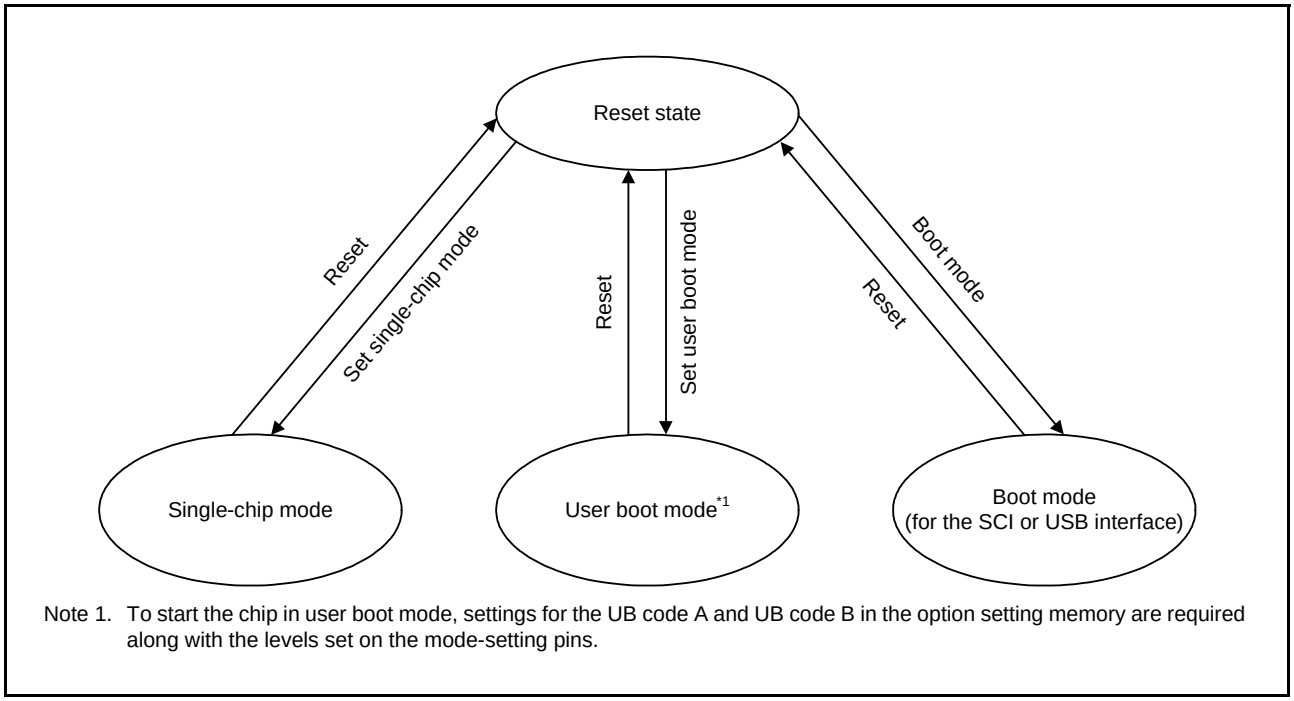
b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	Unique value for each chip														

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	Unique value for each chip														

The UIDRn is a read-only register that stores a 12-byte ID code (unique ID) for identifying the individual MCU. Use longword access to read the UIDRn register.

63.4 Operating Modes Associated with Flash Memory

Figure 63.4 is a diagram of the mode transitions associated with the flash memory. For the procedures for setting the modes, refer to section 3, Operating Modes.



Note 1. To start the chip in user boot mode, settings for the UB code A and UB code B in the option setting memory are required along with the levels set on the mode-setting pins.

Figure 63.4 Mode Transitions Associated with Flash Memory

The flash memory area where programming and erasure are permitted and the boot program after a reset are different according to each mode. The differences between modes are listed in Table 63.2.

Table 63.2 Differences between Modes

Item	Single-chip Mode	User Boot Mode	Boot Mode (for the SCI or USB Interface)
Programmable and erasable area	Code flash memory Data flash memory	Code flash memory Data flash memory	Code flash memory Data flash memory User boot area
Boot program at a reset	User area program	User boot area program	Boot program

63.4.1 Result of Reference by the ID Code Protection

Table 63.3 lists the operating mode at the time of booting up, ID code protection enable/disable, and access limitation in the code flash memory/data flash memory/user boot area by the reference result of the ID code.
For ID code security function settings, see Table 7.2, Settings of the Option-Setting Memory and ID Code Authentication, Reading, Programming, and Erasure, in section 7, Option-Setting Memory.

Table 63.3 Operating Mode at the Time of Booting up, ID Code Protection, and Access Limitation

Operating Mode at the Time of Booting Up	Matching or Non-Matching ID Codes	Access Limitation		
		Code Flash Memory	Data Flash Memory	User Boot Area
Boot mode (for the SCI or USB interface)	Matching	Access not limited	Access not limited	Access not limited
	Non-matching three consecutive times	Access disabled	Access disabled	Access disabled
User boot mode	The ID code protection is not available.			

63.5 Overview of Functions

By using a dedicated flash-memory programmer to program the flash memory of this MCU via a serial interface (serial programming), the device can be rewritten regardless of whether this is before or after it is mounted on the target system. Furthermore, security functions to prohibit overwriting or reading of the user program written to the flash memory are incorporated, and this can prevent falsification and illicit reading of the programs by third parties. Blocks 8 and 9 of the code flash memory can also be protected against reading by using the TM function.
Programming by the user program (self-programming) is available to suit applications where the application on the target system may require updating after manufacturing or shipment. Protection features for the safe overwriting of the flash memory area are also incorporated. Furthermore, interrupt processing during self-programming is supported, so programming can proceed at the same time as processing for external communications, etc., and this is the case in various situations. Table 63.4 lists the overview of the methods of programming and the corresponding operating modes.

Table 63.4 Methods of Programming

Method of Programming	Functional Overview	Operating Mode
Serial programming	A dedicated flash-memory programmer is capable of on-board programming of the flash memory after the device is mounted on the target system. The TM function can be in enabled or disabled at this time.	Boot mode
	A dedicated flash-memory programmer and dedicated programming adapter board allow off-board programming of the flash memory, i.e. programming of the device before it is mounted on the target system.	
Self-programming	The execution of the user program that is written to code flash memory in advance by serial programming executing is also capable of programming the flash memory. The TM function can be in the enabled state at this time. The background operation capability makes it possible to fetch instructions or otherwise read data from code flash memory while the data flash memory is being programmed. Therefore, the data flash memory can be rewritten by executing a program for writing contained by the code flash memory. Furthermore, background operation can also be used for reading and writing to code flash memory alone, but only when the address range of code flash memory that is the target for programming and the address range of code flash memory that is the target for reading satisfy particular conditions (see Table 63.16). When this is the case, at the time of self-programming, a program for programming that is in the code flash memory can be executed to rewrite the code flash memory. In cases where background operation is not possible, instructions in the code flash memory cannot be fetched and data cannot be accessed while code flash memory is being re-written by self-programming. In such cases, a program for programming must be transferred to the internal RAM or external memory in advance and executed.	Normal operating mode and user boot mode

This MCU supports FACI commands for self-programming. For details of the FACI commands, refer to Flash Memory User's Manual: Hardware Interface.

Table 63.5 lists the functions of the flash memory. Serial programmer commands realize each function of serial programming, while reading of the flash memory by an FACI command or the user program realizes each function of self-programming.
For security function settings, see section 7.2.1, Serial Programmer Command Control Register (SPCC), in section 7, Option-Setting Memory.

Table 63.5 List of Basic Functions

Function	Functional Overview	Support Status	
		Serial Programming	Self-programming
Blank check	This is used to check a specified block to ensure that writing to it has not already proceeded. Results of reading from data flash memory to which nothing has been written after erasure are not guaranteed, so use blank checking to confirm that programming to memory has not proceeded after erasure.	Supported	Supported
Block erasure	Erases the memory contents in the specified block.	Supported	Supported
Programming	Write to the specified address.	Supported	Supported
Verify/Checksum	Compares the data read from flash memory and the data transferred from a flash memory programmer.	Supported	Not supported (read by user program is possible)
Read	Read data programmed to flash memory.	Supported	Supported
ID configuration	An ID setting is made for use in controlling the connection of a serial programmer for serial programming.	Supported	Supported
Security configuration	Configures the security function for serial programming.	Supported	Supported with conditions (Only switching the configuration from enabled to disabled is possible)
Protection configuration	Configures lock bits for each block in code flash memory.	Supported	Supported
Option bytes configuration	Configures the option bytes, and modifies the initial settings of this MCU.	Supported	Supported
Configuration clearing	Initializes the ID configuration, security configuration, protection configuration, and option-setting memory. The TM function can be disabled.	Supported	Not supported
Enabling the TM function	The TM function can be enabled.	Supported	Supported

For details on serial programming, refer to PG-FP5 Flash Memory Programmer User's Manual and Renesas Flash Programmer Flash Programming Software User's Manual.
For details on self-programming, refer to Flash Memory User's Manual: Hardware Interface.

The flash memory supports various security functions.
Authentication of the ID code is a security function for use with serial programming.
In serial programming, authentication of the ID code prohibiting connection of a serial programmer, and prohibition of commands (for block erasure, programming, and reading) are available for use as security functions.
Table 63.6 lists the security functions supported by the flash memory, and Table 63.7 lists the operations with security settings.

Table 63.6 Lists of Security Functions

Function	Description
ID authentication	The result of ID authentication can be used to control the connection of a serial programmer for serial programming.
Prohibition of the connection of a serial programmer	The connection of a serial programmer for serial programming is prohibited. Since execution of the configuration clearing command is also prohibited when the connection of a serial programmer is prohibited, changing a security setting from “disabled” to “enabled” is not possible.
Prohibition of block erase commands	Block erase commands at the time of serial programming are prohibited. Since execution of the configuration clearing command is also prohibited when block erase commands are prohibited, changing a security setting from “disabled” to “enabled” is not possible.
Prohibition of programming commands	Programming commands are prohibited at the time of serial programming, and a condition applies to the execution of block erasure commands. Only through execution of the configuration clearing command can the prohibition be lifted.
Prohibition of read commands	Read commands at the time of serial programming are prohibited. Only through execution of the configuration clearing command can the prohibition be lifted.

Table 63.7 Operations with Security Settings

Function	Programming/Erasing/Reading with Respective Security Settings		Notes for Security Setting	
	Serial Programming	**Self-Programming**	**Serial Programming**	**Self-Programming**
ID authentication	(When the ID codes do not match) Block erase commands: Not supported Programming commands: Not supported Read commands: Not supported (When the ID codes match) Block erase commands: Supported Programming commands: Supported Read commands: Supported	(ID authentication is not performed) Block erase commands: Supported Programming commands: Supported Read commands: Supported	The configuration clearing command can initialize the setting for prohibition. The setting for prohibition of block erase commands is not available. The setting for prohibition of programming commands is not available. The setting for prohibition of read commands is not available.	(ID authentication is not performed)
Prohibition of the connection of a serial programmer	Block erase commands: Not supported Programming commands: Not supported Read commands: Not supported	Block erase commands: Supported Programming commands: Supported Read commands: Supported	Since execution of the configuration clearing command is prohibited, initialization of the setting for prohibition is not possible.	Since the configuration clearing command is not supported, initialization of the setting for prohibition is not possible.
Prohibition of block erase commands	Block erase commands: Not supported Programming commands: Supported Read commands: Supported	Block erase commands: Supported Programming commands: Supported Read commands: Supported	Since execution of the configuration clearing command is prohibited, initialization of the setting for prohibition is not possible. The setting to prohibit connection of a serial programmer is not available. The setting for ID authentication to be effective for serial programming is not available.	Since the configuration clearing command is not supported, initialization of the setting for prohibition is not possible.
Prohibition of programming commands	Block erase commands: Supported with conditions*1 Programming commands: Not supported Read commands: Supported	Block erase commands: Supported Programming commands: Supported Read commands: Supported	The configuration clearing command can initialize the setting for prohibition. The setting to prohibit connection of a serial programmer is not available. The setting for ID authentication to be effective for serial programming is not available.	Since the configuration clearing command is not supported, initialization of the setting for prohibition is not possible.
Prohibition of read commands	Block erase commands: Supported Programming commands: Supported Read commands: Not supported	Block erase commands: Supported Programming commands: Supported Read commands: Supported		

Note 1. For details, see section 63.12.41, Flow for Erasure when Programming Commands are Prohibited.

The flash memory supports various protection functions.
Table 63.8 lists the protection functions supported by the flash memory.

Table 63.8 Lists of Protection Functions

Function	Description
Block protection	Block protection settings can be individually made to enable or disable reading or verification of each block of the user area of code flash memory. Programming and erasure by serial programming and self-programming of an area for which the lock bit is set and the lock bit function is enabled are prohibited. Programming or erasure can proceed again when a lock bit function is disabled after having been enabled. When a block of code flash memory is erased, the lock bit for that block is also erased.
User boot protection	Programming and erasure of the user boot area by self-programming are prohibited. Programming and erasure of the user boot area by serial programming are permitted.

63.6 FACI

The FACI controls the FCU according to the specified FACI commands.
For details on the FACI, refer to Flash Memory User's Manual: Hardware Interface.

63.7 Suspend Operation

Reading from the code flash memory/data flash memory is not possible during programming or erasure if the conditions for background operation given in Table 63.16 are not satisfied. When a P/E suspend command is issued to suspend the programming or erasure of the code flash memory/data flash memory, reading from the code flash memory/data flash memory is enabled. Regarding P/E suspend commands, there are one suspend command mode for programming and two suspend command modes for erasure (suspension priority mode and erasure priority mode). To resume suspended programming or erasure, the P/E resume command is available.
For details on the suspend operation, refer to Flash Memory User's Manual: Hardware Interface.

63.8 Protection

This MCU provides three types of protection; software protection, error protection, and boot program protection.
For details on the protection, refer to Flash Memory User's Manual: Hardware Interface.

63.9 User Boot Mode

If the low level is on the MD pin and the high level is on the PC7 pin at the time of release from the reset state, the chip starts in user boot mode.
The reset vector at this time points to the address FF7F FFFCh of the user boot area. For other vectors, refer to a normal vector table. See section 15, Interrupt Controller (ICUA).
In user boot mode, a program can be created by using any interface and the programming or erasure of the code flash memory/data flash memory is enabled by issuing the FACI command. Note that the programming to the user boot area should be performed in boot mode.
In addition, the settings for UB code A and UB code B are required. See section 7.3, UB Codes.

63.10 Boot Mode

There are two serial programming modes; the boot mode (for the SCI interface) with SCI and the boot mode (for the USB interface) with USB. Table 63.9 lists the I/O pins used in boot mode.

Table 63.9 I/O Pins Used in Boot Mode

Pin Name	I/O	Mode to be Used	Use
MD	Input	Boot mode (for the SCI interface) User boot mode Boot mode (for the USB interface)	Selection of operating mode
PC7/UB	Input		Selection of boot mode (for the SCI interface) or user boot mode*1/boot mode (for the USB interface)
PF2/RXD1 (177/176-pin packages) P30/RXD1 (145/144/100-pin packages)	Input	Boot mode (for the SCI interface)	For host communication (to receive data through SCI)
PF0/TXD1 (177/176-pin packages) P26/TXD1 (145/144/100-pin packages)	Output		For host communication (to transmit data through SCI)
USB0_DP, USB0_DM	I/O	Boot mode (for the USB interface)	Data input/output of USB
P16/USB0_VBUS	Input		Detection of connection and disconnection of USB cables
P35/UPSEL	Input		Selection of USB bus-power mode or self-power mode

Note 1. To enable user boot mode, the settings for UB code A and UB code B are required.

63.10.1 Boot Mode (for the SCI Interface)

In boot mode (for the SCI interface), the host sends control commands and data for programming, and the user boot area in the code flash memory or data flash memory is programmed or erased accordingly. An on-chip SCI handles transfer between the host and this MCU in asynchronous mode. Tools for transmission of control commands and the data for programming must be prepared in the host.

When this MCU is activated in boot mode (for the SCI interface), the program in a dedicated area within the MCU is executed. The boot program automatically adjusts the bit rate of the SCI and controls programming/erasure by receiving control commands from the host.

Figure 63.5 shows the system configuration for operations in boot mode (for the SCI interface).

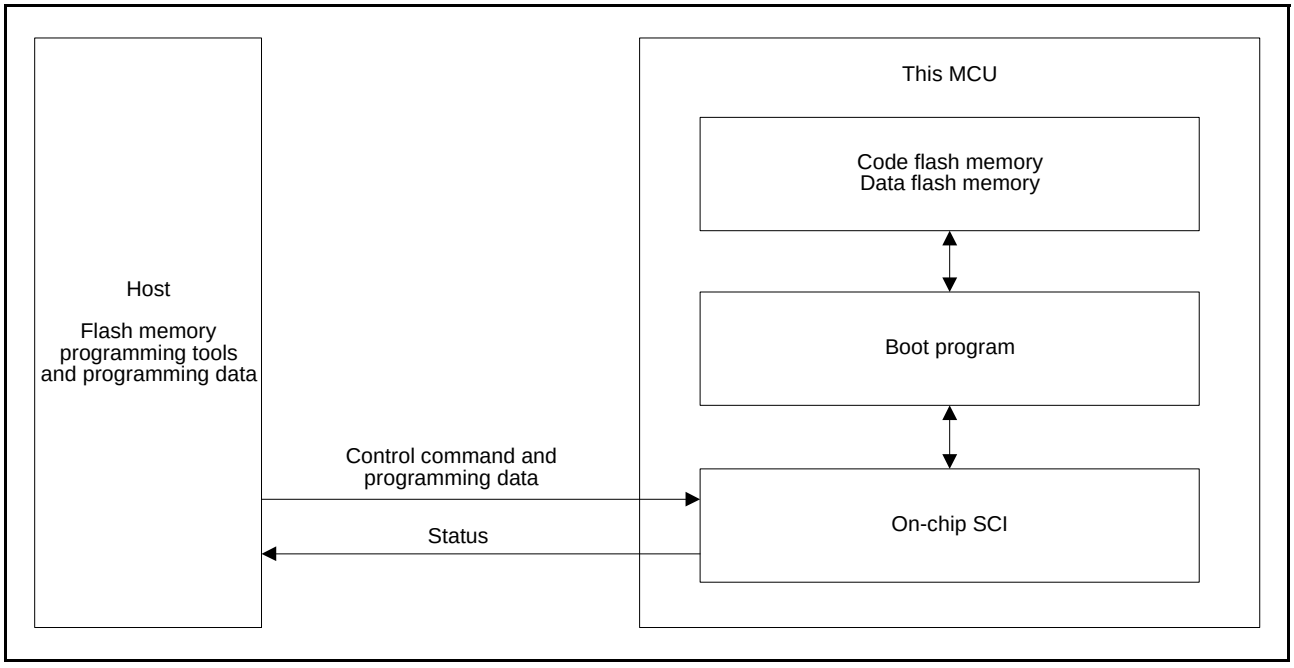


Figure 63.5 System Configuration for Operations in Boot Mode (for the SCI Interface)

63.10.2 Boot Mode (for the USB Interface)

In boot mode (for the USB interface), the code flash memory/data flash memory is programmed or erased by control commands and data for programming transmitted from an externally connected host via the USB.
Using boot mode (for the USB interface) requires preparation on the host side of tools for transmitting the control commands and data for programming, and of the data. Figure 63.6 shows the configuration of a system for use in boot mode (for the USB interface).

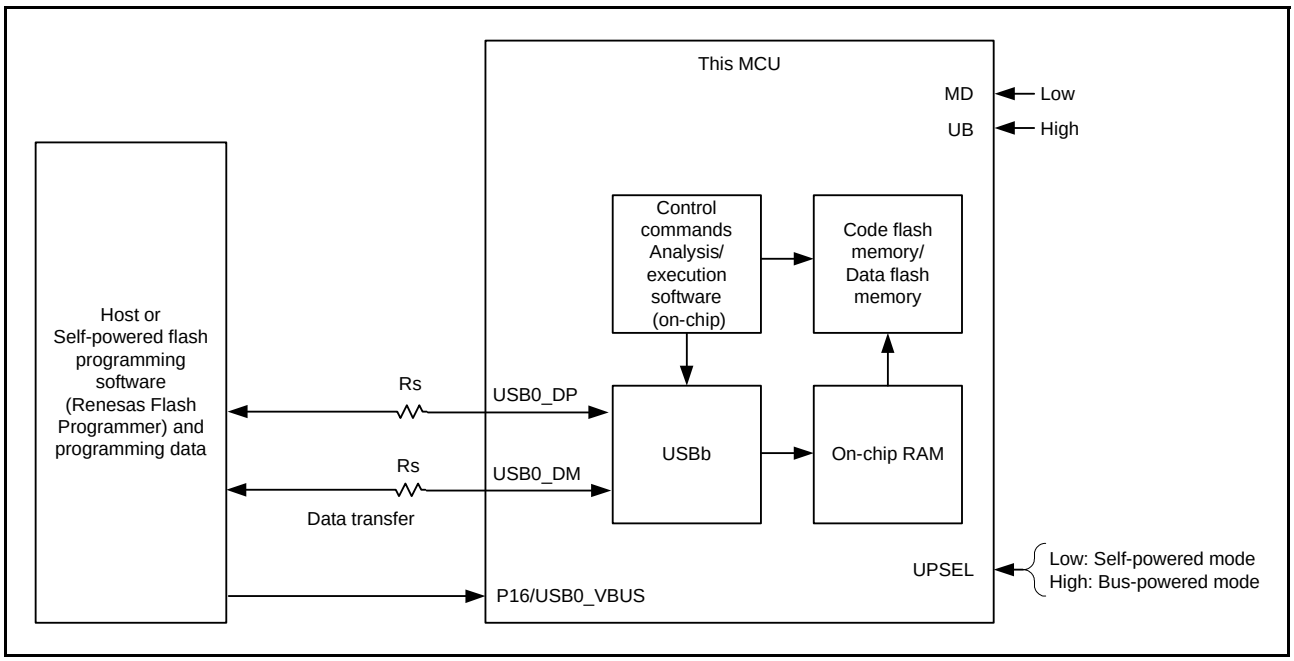


Figure 63.6 System Configuration in Boot Mode (for the USB Interface)

In boot mode (for the USB interface), self-power mode or bus power mode is selected for operation. Release from the reset state while the low level is being applied to the MD and UPSEL pins and the high level is being applied to the UB pin selects self-powered mode. Release from the reset state while the low is being applied to the MD pin and the high level is being applied to the UB and UPSEL pins are high selects bus-power mode.
Table 63.10 shows the enumeration information for each mode.

Table 63.10 Enumeration Information

USB Specification	Ver.2.0 (Full-Speed)	
Maximum current	Self-powered mode (pins P35 and UPSEL = Low)	100 mA
	Bus-powered mode (pins P35 and UPSEL = High)	500 mA

63.11 ID Code Protection

There are three types of ID code protection: on-chip debugger ID code protection, ID code protection, and ROM code protection.

63.11.1 ID Code Protection on Connection of the On-Chip Debugger

This function is used to prohibit connection with the on-chip debugger. When connecting an on-chip debugger, the control code and ID code that have been written to the option-setting memory are used to determine whether ID code protection on connection of the on-chip debugger is enabled or disabled and to judge ID code protection on connection of the on-chip debugger.
When the ID code protection is enabled, the code sent from the on-chip debugger is compared with the control code and ID code in the option-setting memory to determine whether they match. If they match, connection with the on-chip debugger is allowed.
If they do not match, the on-chip debugger cannot be connected.

63.11.2 ID Code Protection

This function is used to prohibit connection with the serial programmer. When connecting a serial programmer, the ID code set in the OCD/serial programmer ID setting register (OSIS) and written in the option-setting memory is used to judge ID code protection on connection of the serial programmer.
When the ID code protection is enabled, the code sent from the serial programmer is compared with the ID code set in the OCD/serial programmer ID setting register (OSIS) to determine whether they match. If they match, connection with the serial programmer is allowed. If they do not match, the serial programmer cannot be connected.

63.11.3 ROM Code Protection

ROM code protection is a function to prohibit reading from or programming/erasure to the flash memory when the parallel programmer is used during off-board programming. The ROM code in flash memory is a 32-bit code.
Figure 63.7 shows the configuration of the ROM code and Table 63.11 shows the specifications for ROM code protection. Set the ROM code in 32-bit units.
For release from ROM code protection, erase the block 0 of the user area that contains the ROM code in boot mode or by user program.

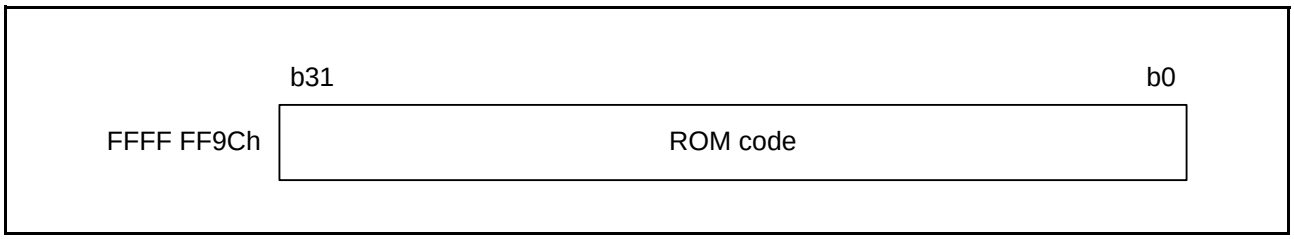


Figure 63.7 Configuration of the ROM Code

Table 63.11 Specifications for ROM Code Protection

ROM Code	State of Protection	Operations at the Time of Connection with the Flash Programmer
0000 0000h	Protection enabled (ROM code protection 1)	Access (both reading and programming/erasure) to the user area and the user boot area are prohibited.
0000 0001h	Protection enabled (ROM code protection 2)	Reading from the user area and the user boot area are prohibited.
Other than above	Protection disabled	Access (both reading and programming/erasure) to the user area and the user boot area are permitted.

63.12 Boot Mode Communications Protocol

This section describes the communications protocol for use in boot mode. When developing a programmer, use this communications protocol to control it.

63.12.1 How to Start the Chip Up in Boot Mode (for the SCI Interface)

The chip starts up in boot mode (for the SCI interface) if both the MD and PC7 pins are at the low level on release from the reset state (i.e. when the level on the RES# pin changes from low to high). A waiting time of at least 400 ms is required while the RES# pin is held at the high level after the chip starts up in boot mode (for the SCI interface) until communications with the MCU can proceed.

Figure 63.8 shows the states of pins up to communications (through the SCI interface) in boot mode becoming possible.

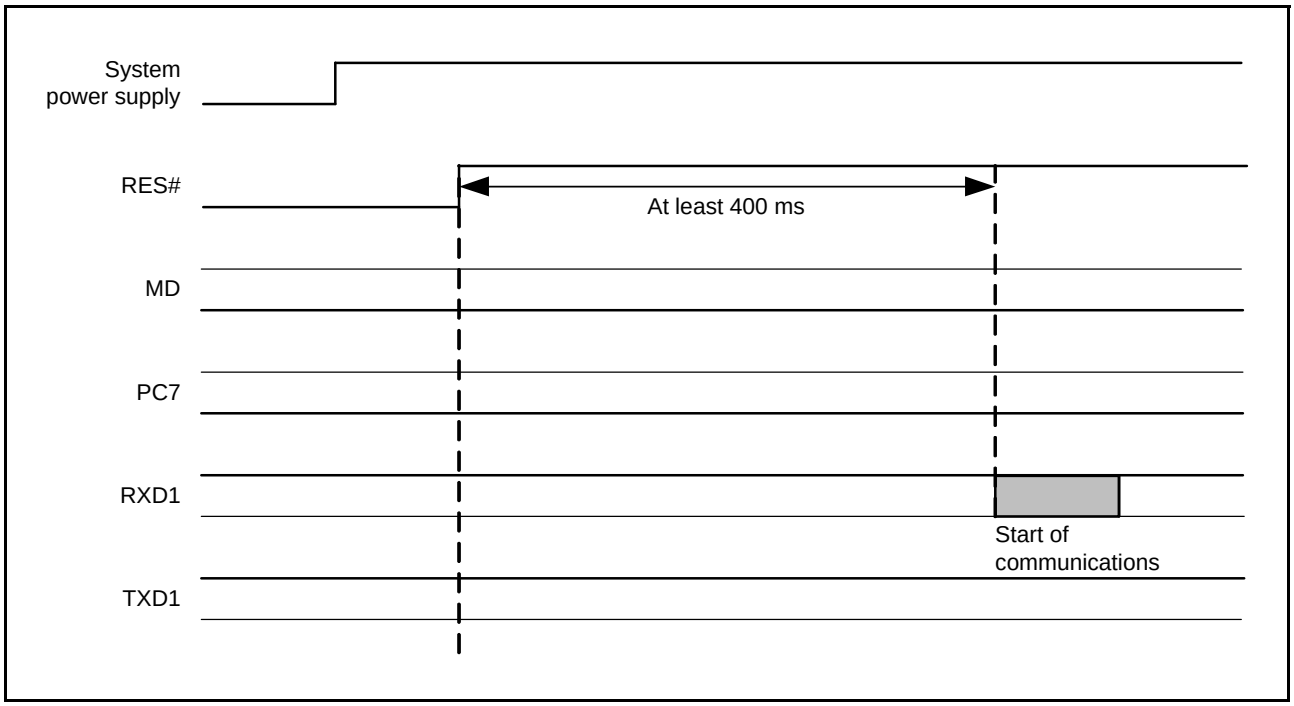


Figure 63.8 States of Pins Up to Communications (through the SCI Interface) in Boot Mode Becoming Possible

63.12.2 State Transitions in Boot Mode (for the SCI Interface)

Figure 63.9 is a diagram of the state transitions in boot mode (for the SCI interface).

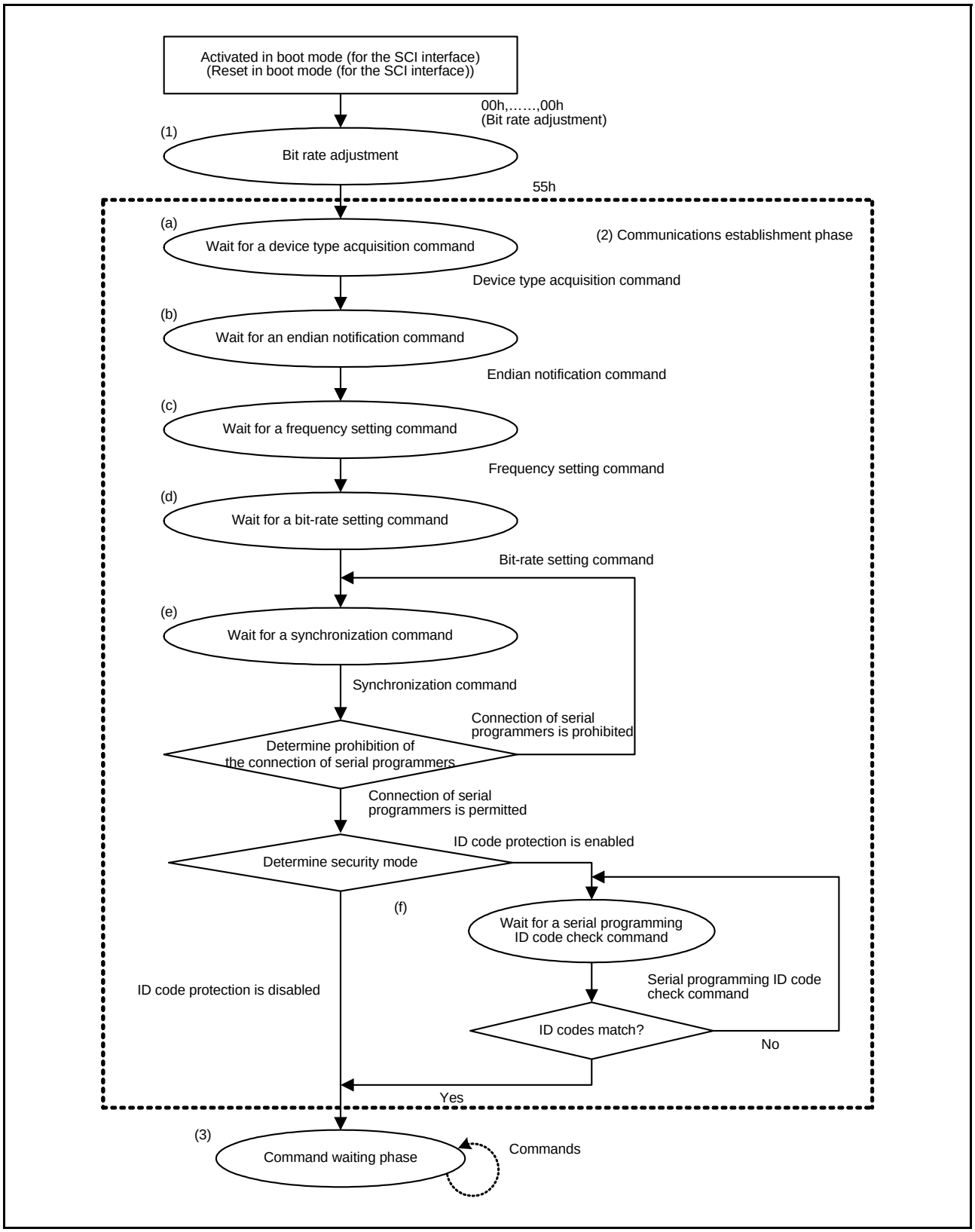


Figure 63.9 State Transitions in Boot Mode (for the SCI Interface)

(1) Matching the bit rates

When this MCU is activated in boot mode, the bit rate of the SCI is automatically adjusted to match that of the host. On completion of this automatic bit rate adjustment, this MCU transmits the value 00h to the host. On subsequent correct reception of the value 55h sent from the host, this MCU enters the communications establishment phase. For details on matching of the bit rates, see section 63.12.3, Automatic Adjustment of the Bit Rate.

(2) Communications establishment phase

The device, endian, frequency, and bit rate are selected in this phase. ID authentication also proceeds if ID code protection is enabled. For the commands to use in the communications establishment phase, see section 63.12.5, Communications Establishment Phase.

(a) Waiting for a device type acquisition command

In this state, the MCU is waiting for a device type acquisition command to be sent. When it receives a device type acquisition command from the host, the state shifts to waiting for an endian notification command. For details of the device type acquisition command, see section 63.12.9, Device Type Acquisition Command.

(b) Waiting for an endian notification command

In this state, the MCU is waiting for an endian notification command to be sent. When it receives an endian notification command from the host, the state shifts to waiting for a frequency setting command. For details of the endian notification command, see section 63.12.10, Endian Notification Command.

(c) Waiting for a frequency setting command

In this state, the MCU is waiting for a frequency setting command to be sent. When it receives a frequency setting command from the host, the state shifts to waiting for a bit-rate setting command. For details of the frequency setting command, see section 63.12.11, Frequency Setting Command.

(d) Waiting for a bit-rate setting command

In this state, the MCU is waiting for a bit-rate setting command to be sent. When it receives a bit-rate setting command from the host, the state shifts to waiting for a synchronization command. For details of the bit-rate setting command, see section 63.12.12, Bit-Rate Setting Command.

(e) Waiting for a synchronization command

In this state, the MCU is waiting for a synchronization command to be sent. When it receives a synchronization command from the host, it determines whether ID code protection is enabled or disabled. When ID code protection is disabled, the MCU enters the command waiting phase. When ID code protection is enabled, it enters the state of waiting for a serial programming ID code check command. If the MCU has been set to prohibit the connection of a serial programmer, this MCU transmits an error code to indicate that connecting a serial programmer is prohibited and remains in the state of waiting for a synchronization command. For details of the synchronization command, see section 63.12.13, Synchronization Command.

(f) Waiting for a serial programming ID code check command

In this state, the MCU is waiting for a serial programming ID code check command to be sent. The ID code sent from the host is compared with the ID code written in the option-setting memory area, and the command waiting phase is entered if the two match. If they do not match, the next transition is back to the state of waiting for a serial programming ID code check command. For details of the ID code check command, see section 63.12.15, Serial Programming ID Code Check Command.

(3) Phase of waiting for commands

In this state, programming and erasure proceed in accord with commands from the host. For details of the commands that can be issued in the command waiting phase, see section 63.12.6, Command Waiting Phase.

63.12.3 Automatic Adjustment of the Bit Rate

When this MCU is booted up in boot mode (for the SCI interface), asynchronous transfer by the SCI is used to measure the periods at low level of consecutive bytes with value 00h that are sent from the host. While the period at low level is being measured, set the host's SCI transfer format to 8-bit data, one stop bit, no parity, and a transfer rate of 9,600 bps. This MCU measures the periods at low level in the signal from the host, adjusts the bit rate of its SCI, and then sends the value 00h to the host.

If reception of the value 00h by the host is successful, the host responds by sending the value 55h to this MCU. If successful reception of 00h by the host is not possible, reboot this MCU in boot mode, and then repeat the process of automatically adjusting the bit rate. If reception of the value 55h by this MCU is successful, it responds by sending C1h to the host, and if successful reception of 55h by this MCU is not possible, it responds by sending FFh to the host.

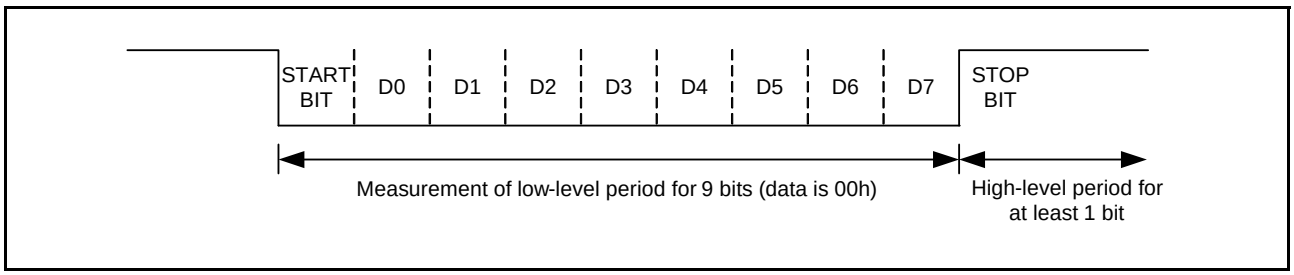


Figure 63.10 Transfer Format Used by SCI in Automatic Adjustment of Bit Rate

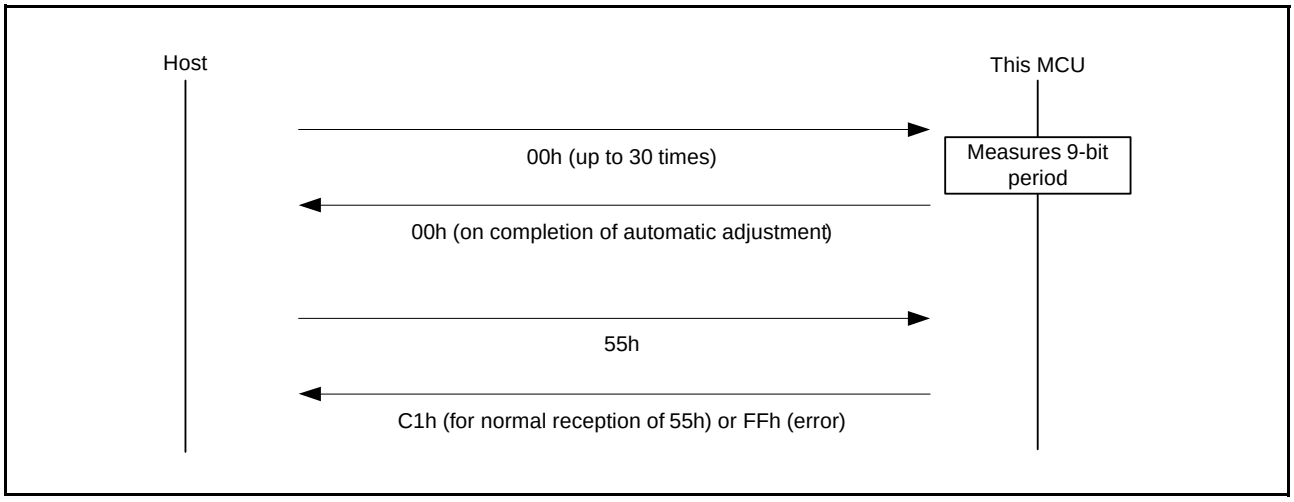


Figure 63.11 Sequence of Transfer between Host and This MCU

For the host's SCI bit rate, ensure that SCI communications proceed under the conditions given in Table 63.12.

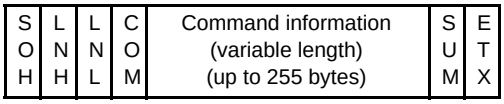
Table 63.12 Conditions for Automatic Bit-Rate Adjustment

Bit Rate of SCI in Host
9,600 bps

63.12.4 Packet Format

(1) Command packet

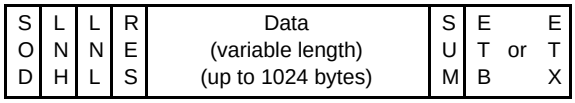
The host sends commands to this MCU in the format below.



Symbol	Code	Description
SOH	01h	Start of a packet (1 byte)
LNH	—	Packet length (length of COM and the command information) (8 to 15 bits) (1 byte)
LNL	—	Packet length (length of COM and the command information) (0 to 7 bits) (1 byte)
COM	—	Command code (1 byte)
Command information	—	Command information (up to 255 bytes)
SUM	—	Sum of values of LNH, LNL, COM, and the command information (1 byte)
ETX	03h	End of a packet (1 byte)

(2) Status packet and data packet

Data transmission proceeds between the host and this MCU in the format below.



Symbol	Code	Description
SOD	81h	Start of a packet (1 byte)
LNH	—	Packet length (length of RES and the data) (8 to 15 bits) (1 byte)
LNL	—	Packet length (length of RES and the data) (0 to 7 bits) (1 byte)
RES	—	Response code (1 byte)
Data	—	Data (up to 1024 bytes)
SUM	—	Sum of values of LNH, LNL, RES, and the data (1 byte)
ETB	17h	End of a packet (1 byte)
ETX	03h	End of the last packet (1 byte)

63.12.5 Communications Establishment Phase

Table 63.13 lists the commands available in the commands establishment phase.
The synchronization command and ID authentication mode acquisition command can also be used in the command waiting phase.

Table 63.13 Commands Available in the Communications Establishment Phase

Command Name	Function
Device type acquisition	Transmits the oscillation frequency and CPU operating frequency (in Hz) supported by boot mode to the host.
Endian notification	Indicates whether big-endian or little-endian is to be used.
Frequency setting	Sets the values of the oscillation frequency and CPU operating frequency (in Hz).
Bit-rate setting	Changes the bit rate.
Synchronization	This command is used in processing for communications synchronization. It is also used when confirming whether the MCU can accept commands.
ID authentication mode acquisition	Transmits an indication of whether ID authentication is enabled or disabled in boot mode to the host.
Serial programming ID code check	Determines whether the ID code written in the option-setting memory matches the ID code sent by the host.

In the communications establishment phase, send commands from the host in the order of the device type acquisition, endian notification, frequency setting, bit-rate setting, and synchronization commands according to the responses to commands. When ID authentication in boot mode is enabled, send the ID authentication mode acquisition or serial programming ID code check command following the synchronization command.
If commands are issued in an incorrect order or other commands are issued, this MCU returns a response indicating a flow error.

63.12.6 Command Waiting Phase

Table 63.14 lists the commands available in the command waiting phase.
The synchronization command and ID authentication mode acquisition command can also be used in the communications establishment phase.

Table 63.14 Commands Available in the Command Waiting Phase

Command Name	Function
Synchronization	See Table 63.13.
Blank check	Check that a selected area is blank.
Block erase	Erases a selected single block.
Area erasure	Erases the specified area.
Programming	Programs the selected area.
Read	Reads data from a selected area.
Lock-bit setting	Sets the lock bits.
Lock-bit acquisition	Obtains the settings of lock bits.
Lock bit enable	Enables the setting of lock bits.
Lock bit disable	Disables the setting of lock bits.
ID authentication mode acquisition	See Table 63.13.
Command protection setting	Enables prohibition of block erase commands, programming commands, or read commands.
Command protection acquisition	Obtains the settings for protection against block erase commands, programming commands, and read commands.
Boot mode ID code setting	Sets the ID code and enables ID authentication in boot mode.
ID code setting	Sets the ID code.
ID code acquisition	Obtains the ID code setting.
Serial programmer connection prohibition	Prohibits the connection of serial programmers.
OFS setting	Sets the OFS0 and OFS1 registers.
OFS acquisition	Obtains the setting of the OFS0 and OFS1 registers.
Endian setting	Sets the MDE register.
Endian acquisition	Obtain the setting of the MDE register.
Configuration clearing	Clears the security function, ID code, endian, and OFS.
TM setting	Enables the TM function and sets the TMINF register.
TM acquisition	Acquires an indicator of whether the TM function is currently enabled or disabled, the contents of the TMINF register, and the addresses where the TM target areas start and end.
Simple addition checksum	Calculates the sum of values in a selected area.
Signature acquisition	Obtains information of the flash memory configuration.

If the host has sent an undefined command, this MCU returns a response indicating an error in the form of a non-supported command.

63.12.7 Command Transfer Sequence

Though the sequence of transfer differs from command to command, common transfer sequences are used for the commands that only make settings for this MCU and for the commands that obtain information on the settings in this MCU. However, as the contents of the command packet, status packet, and data packet differ for each command, see the sections on the individual commands for details.

(1) Common transfer sequence for the commands that only make settings

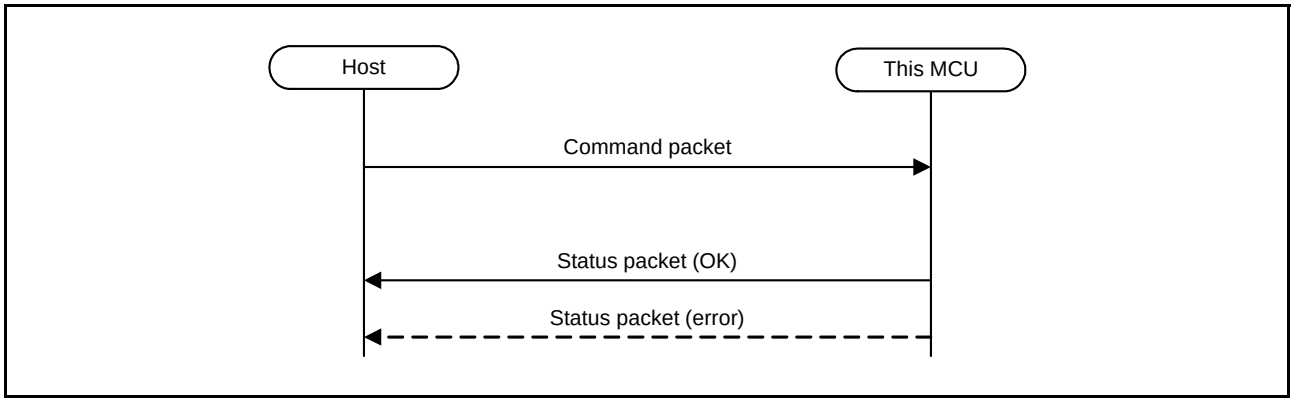


Figure 63.12 Common Transfer Sequence for the Commands that Only Make Settings

(2) Common transfer sequence for the commands that obtain information on settings

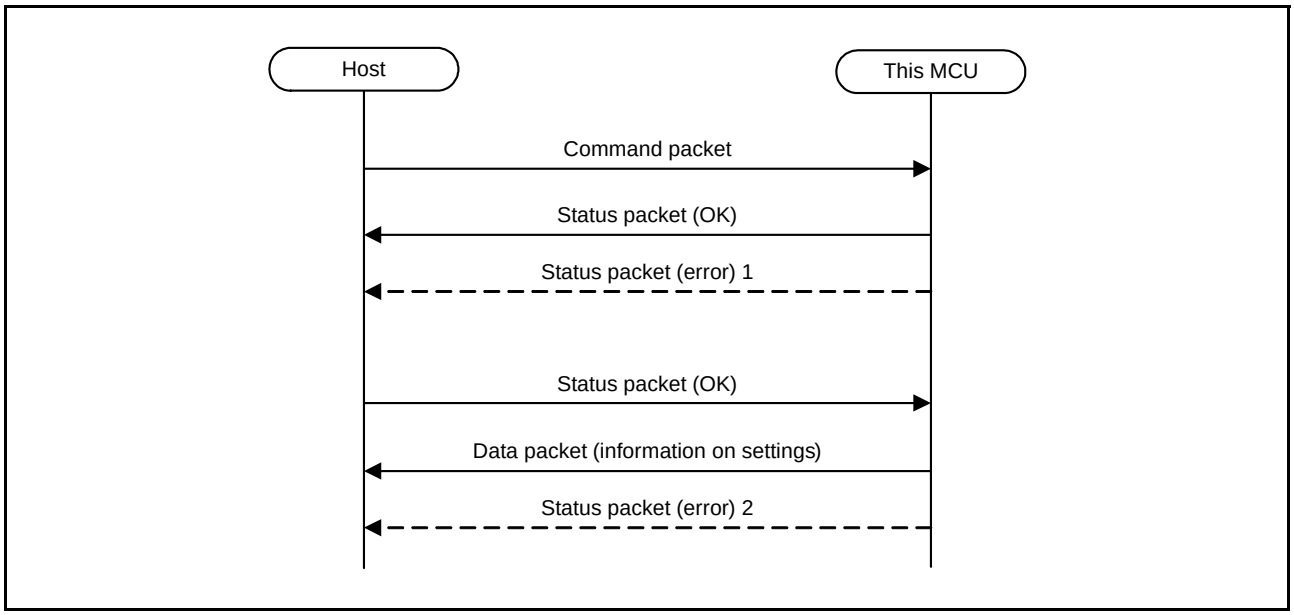


Figure 63.13 Common Transfer Sequence for the Commands that Obtain Information on Settings

Table 63.15 Common Transfer Sequence

Command Name	Common Transfer Sequence Type
Device type acquisition	Command that obtains information on settings
Endian notification	Command that only makes a setting
Frequency setting	Command that obtains information on settings
Bit rate setting	Not in a common transfer sequence
Synchronization	Command that only makes a setting
ID authentication mode acquisition	Command that obtains information on settings
Serial programming ID code check	Command that only makes a setting
Blank check	Command that only makes a setting
Block erase	Command that only makes a setting
Area erasure	Command that only makes a setting
Programming	Not in a common transfer sequence
Read	Not in a common transfer sequence
Lock-bit setting	Command that only makes a setting
Lock-bit acquisition	Command that obtains information on settings
Lock bit enable	Command that only makes a setting
Lock bit disable	Command that only makes a setting
Command protection setting	Command that only makes a setting
Command protection acquisition	Command that obtains information on settings
Boot mode ID code setting	Command that only makes a setting
ID code setting	Command that only makes a setting
ID code acquisition	Command that obtains information on settings
Serial programmer connection prohibition	Command that only makes a setting
OFS setting	Command that only makes a setting
OFS acquisition	Command that obtains information on settings
Endian setting	Command that only makes a setting
Endian acquisition	Command that obtains information on settings
Configuration clearing	Command that only makes a setting
TM setting	Command that only makes a setting
TM acquisition	Command that obtains information on settings
Simple addition checksum	Command that obtains information on settings
Signature acquisition	Command that obtains information on settings

For the command transfer sequences that are not in a common transfer sequence, see the sections on the individual commands.

63.12.8 Non-supported Commands

If an undefined command packet is sent, this MCU returns the “non-supported” error code (C0h) and returns to the command waiting state.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: Packet length (8 to 15 bits)
LNL: Packet length (0 to 7 bits)
COM: Command code*1
SUM: Sum of values
ETX: 03h

Note 1. Command code other than those specified in Table 63.14.

(2) Status packet structure

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 80h | COM (command code)
ERR: Error code
C0h (“non-supported” error)
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

63.12.9 Device Type Acquisition Command

This command is used to make the MCU send the input frequency and system clock frequency (in Hz) supported by boot mode (for the SCI interface).
This command can only be accepted in the communications establishment phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 38h
SUM: C7h
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	TYP	OSA	OSI	CPA	CPI	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 19h
RES: 38h (OK)
TYP: Type code (8 bytes)*1
OSA: Maximum input frequency (4 bytes)
OSI: Minimum input frequency (4 bytes)
CPA: Maximum system clock frequency (4 bytes)
CPI: Minimum system clock frequency (4 bytes)
SUM: Sum of values
ETX: 03h

An example of the values sent is given below.

Maximum input frequency = 16,000,000 Hz
- OSA (1st byte): 00h
- OSA (2nd byte): F4h
- OSA (3rd byte): 24h
- OSA (4th byte): 00h

Minimum input frequency = 16,000,000 Hz
- OSI (1st byte): 00h
- OSI (2nd byte): F4h
- OSI (3rd byte): 24h
- OSI (4th byte): 00h

Maximum system clock (ICLK) = 120,000,000 Hz
- CPA (1st byte): 07h
- CPA (2nd byte): 27h
- CPA (3rd byte): 0Eh
- CPA (4th byte): 00h

Minimum system clock: (ICLK) = 120,000,000 Hz
- CPI (1st byte): 07h
- CPI (2nd byte): 27h
- CPI (3rd byte): 0Eh
- CPI (4th byte): 00h

Note 1. Reserved data

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 38h (OK)
SUM: C7h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B8h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B8h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.10 Endian Notification Command

This command is used to inform the MCU of the endian (big or little).
Specify either endian as the endian information according to the data to be programmed.
This command can only be accepted in the communications establishment phase.

(1) Command packet structure

SOH	LNH	LNL	COM	END	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 02h
COM: 36h
END: Endian information
00h (big-endian)
01h (little-endian)
SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 36h (OK)
SUM: C9h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B6h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D7h (endian error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.11 Frequency Setting Command

This command is used to set the values of oscillation frequency and CPU operating frequency (in Hz).
In boot mode, the HOCO runs at 16 MHz and the ICLK at 120 MHz, so set the input frequency and system clock frequency bytes to the hexadecimal values corresponding to 16 MHz and 120 MHz, respectively. Additionally, since the FCLK and PCLKB run at 60 MHz in boot mode, the MCU returns the hexadecimal value corresponding to 60 MHz to indicate the peripheral clock frequency.
This command can only be accepted in the communications establishment phase.

(1) Command packet structure

SOH	LNH	LNL	COM	OC1	OC2	OC3	OC4	CC1	CC2	CC3	CC4	SUM	ETX

When the input frequency is 16,000,000 Hz and the system clock frequency is 120,000,000 Hz, send the values as below.
OC1: 00h CC1: 07h
OC2: F4h CC2: 27h
OC3: 24h CC3: 0Eh
OC4: 00h CC4: 00h

SOH: 01h
LNH: 00h
LNL: 09h
COM: 32h
OC1: Input frequency
OC2: Input frequency
OC3: Input frequency
OC4: Input frequency
CC1: System clock frequency
CC2: System clock frequency
CC3: System clock frequency
CC4: System clock frequency
SUM: Sum of values
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	FQ1	FQ2	FQ3	FQ4	PF1	PF2	PF3	PF4	SUM	ETX

An example of the values sent is given below.
System clock frequency = 120,000,000 Hz
Peripheral clock frequency = 60,000,000 Hz
FQ1: 07h PF1: 03h
FQ2: 27h PF2: 93h
FQ3: 0Eh PF3: 87h
FQ4: 00h PF4: 00h

SOD: 81h
LNH: 00h
LNL: 09h
RES: 32h
FQ1: System clock frequency
FQ2: System clock frequency
FQ3: System clock frequency
FQ4: System clock frequency
PF1: Peripheral clock frequency
PF2: Peripheral clock frequency
PF3: Peripheral clock frequency
PF4: Peripheral clock frequency
SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 32h (OK)
SUM: CDh
ETX: 03h

(4) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B2h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D1h (input frequency error)
D2h (system clock (ICLK) frequency error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.12 Bit-Rate Setting Command

This command is used to change the bit rate after receiving bit rate data (in bps).
If an error occurs, the bit rate is not changed.
This command can only be accepted in the communications establishment phase.

(1) Procedure

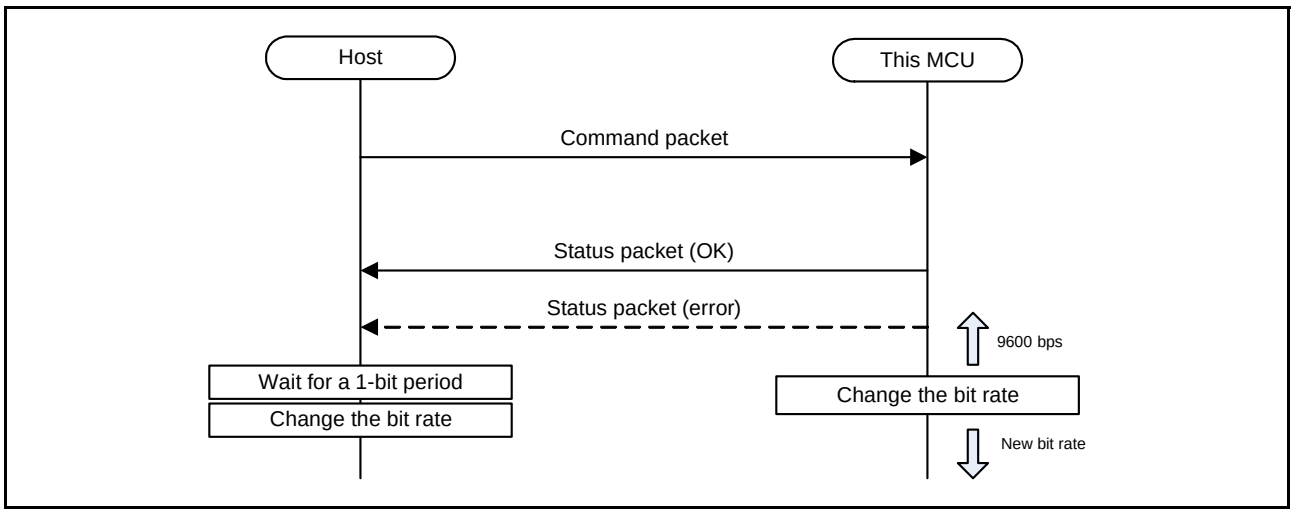


Figure 63.14 Bit-Rate Setting Command Transfer Sequence

(2) Command packet structure

SOH	LNH	LNL	COM	BR1	BR2	BR3	BR4	SUM	ETX

When the bit rate is 2,000,000 bps, send the values as below.
BR1: 00h
BR2: 1Eh
BR3: 84h
BR4: 80h

SOH: 01h
LNH: 00h
LNL: 05h
COM: 34h
BR1: Bit rate
BR2: Bit rate
BR3: Bit rate
BR4: Bit rate
SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 34h (OK)
SUM: CBh
ETX: 03h

(4) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B4h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D4h (bit rate error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.13 Synchronization Command

This command is used in processing to synchronize communications.
It is also used when checking whether the MCU is ready to accept commands. If a serial programmer connection is prohibited, a serial programmer connection prohibition error is returned.
This command can be accepted in both the communications establishment and command waiting phases.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 00h
SUM: FFh
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 00h (OK)
SUM: FFh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 80h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
DCh (serial programmer connection prohibition error)
SUM: Sum of values
ETX: 03h

63.12.14 ID Authentication Mode Acquisition Command

This command makes the MCU indicate whether ID authentication is enabled or disabled in boot mode.
This command can be accepted in both the communications establishment and command waiting phases.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 2Ch
SUM: D3h
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	MOD	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 2Ch (OK)
MOD: ID authentication information (1 byte)
00h (ID authentication enabled)
FFh (ID authentication disabled)
SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 2Ch (OK)
SUM: D3h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: ACh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: ACh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.15 Serial Programming ID Code Check Command

The MCU checks whether its own ID matches that sent from the host and notifies the host of the result.
This command can be accepted in the communications establishment phase. When ID authentication in boot mode is enabled, the MCU does not enter the command waiting phase unless processing in response to this command ends normally.

(1) Command packet structure

SOH	LNH	LNL	COM	IDC	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 11h
COM: 30h
IDC: ID code (16 bytes)*1
SUM: Sum of values
ETX: 03h

Note 1. Send the values as below.
<ID code>
ID = 128'h0F0E0D0C0B0A09080706050403020100
(ID0:00h, ID1:01h, ID2:02h, ... , ID15:0Fh)
<Data for transmission>

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	…
00h	01h	02h	03h	04h	05h	06h	07h	…

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 30h (OK)
SUM: CFh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B0h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
DBh (ID code mismatch error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.16 Blank Check Command

This command checks whether the specified area is blank.
Specify a range in the user area or user boot area with addresses on 256-byte boundaries or in the data area with addresses on 16-byte boundaries. When the TM function is enabled, an attempt at blank checking of the code flash memory in blocks 8 and 9 leads to an error since they are being handled as TM target area.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	EHH	EHL	ELH	ELL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM: 10h
SHH: Blank check start address (24 to 31 bits)
SHL: Blank check start address (16 to 23 bits)
SLH: Blank check start address (8 to 15 bits)
SLL: Blank check start address (0 to 7 bits)
EHH: Blank check end address (24 to 31 bits)
EHL: Blank check end address (16 to 23 bits)
ELH: Blank check end address (8 to 15 bits)
ELL: Blank check end address (0 to 7 bits)
SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 10h (OK)
SUM: EFh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 90h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D0h (address error)
E0h (non-blank error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.17 Block Erase Command

This command is used to erase a specified single block.
Specify the block for erasure as the first address in the block.
If the lock-bit function is disabled and the lock bit of a block for erasure is set, the lock bit is also erased so the block is released from locking. While the TM function is enabled, blocks 8 and 9 in the code flash memory are handled as TM target areas and so cannot be erased.
To erase areas being handled as TM target, use the configuration clearing command described in section 63.12.35, Configuration Clearing Command to restore them to normal handling.
If programming commands are prohibited, follow the procedure for erasure prescribed in section 63.12.41, Flow for Erasure when Programming Commands are Prohibited.
This command cannot be used when block erase commands are prohibited.
If processing of erasure is forcibly stopped in response to a command due to a reset, the lock bit may be set. In such a case, see (6) Abnormal Termination During Programming/Erasure or Blank Checking in section 63.17, Usage Notes.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 05h
COM: 12h
SHH: First address in the block for erasure (24 to 31 bits)
SHL: First address in the block for erasure (16 to 23 bits)
SLH: First address in the block for erasure (8 to 15 bits)
SLL: First address in the block for erasure (0 to 7 bits)
SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 12h (OK)
SUM: EDh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 92h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D0h (address error)
DAh (protection error)
E1h (erase error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.18 Area Erase Command

The area erase command erases the specified area successively, block by block, in ascending order of addresses.
The target area can be selected as the user area or user boot area, or as the data flash area.
When an area for which the lock bit function is disabled and the lock bit has been set is erased, the lock-bit setting is cleared.
When the TM function is enabled, only the blocks other than those in the TM target areas are erased.
This command can only be accepted in the command waiting phase.
If processing in response to the command is forcibly stopped due to a reset, etc., while erasure is in progress, the lock bit may be set. In this case, see section 63.17, Usage Notes, (6) Abnormal Termination During Programming/Erasure or Blank Checking.

(1) Command packet structure

SOH	LNH	LNL	COM	ARE	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 02h
COM: 50h
ARE: Area
00h (user area)
10h (user boot area)
20h (data area)
SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 50h (OK)
SUM: AFh
ETX: 03h

(3) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	SUM	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D0h (Error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D5h (area error)
DAh (protection error)
E1h (erase error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.19 Programming Command

This command sets up the MCU for data to be programmed to its flash memory and specifies the area where the data are to be programmed.
Specify addresses for a data length that is a multiple of 256 bytes for the user area or user boot area, or a multiple of 16 bytes for the data area. Furthermore, as the address where programming is to start, specify an address on a 256-byte boundary for the user area or user boot area or on a 16-byte boundary for the data area. When the TM function is enabled, blocks 8 and 9 in the code flash memory are handled as TM target areas and so cannot be programmed.
This command cannot be used when programming commands are prohibited.
This command can only be accepted in the command waiting phase.

(1) Procedure

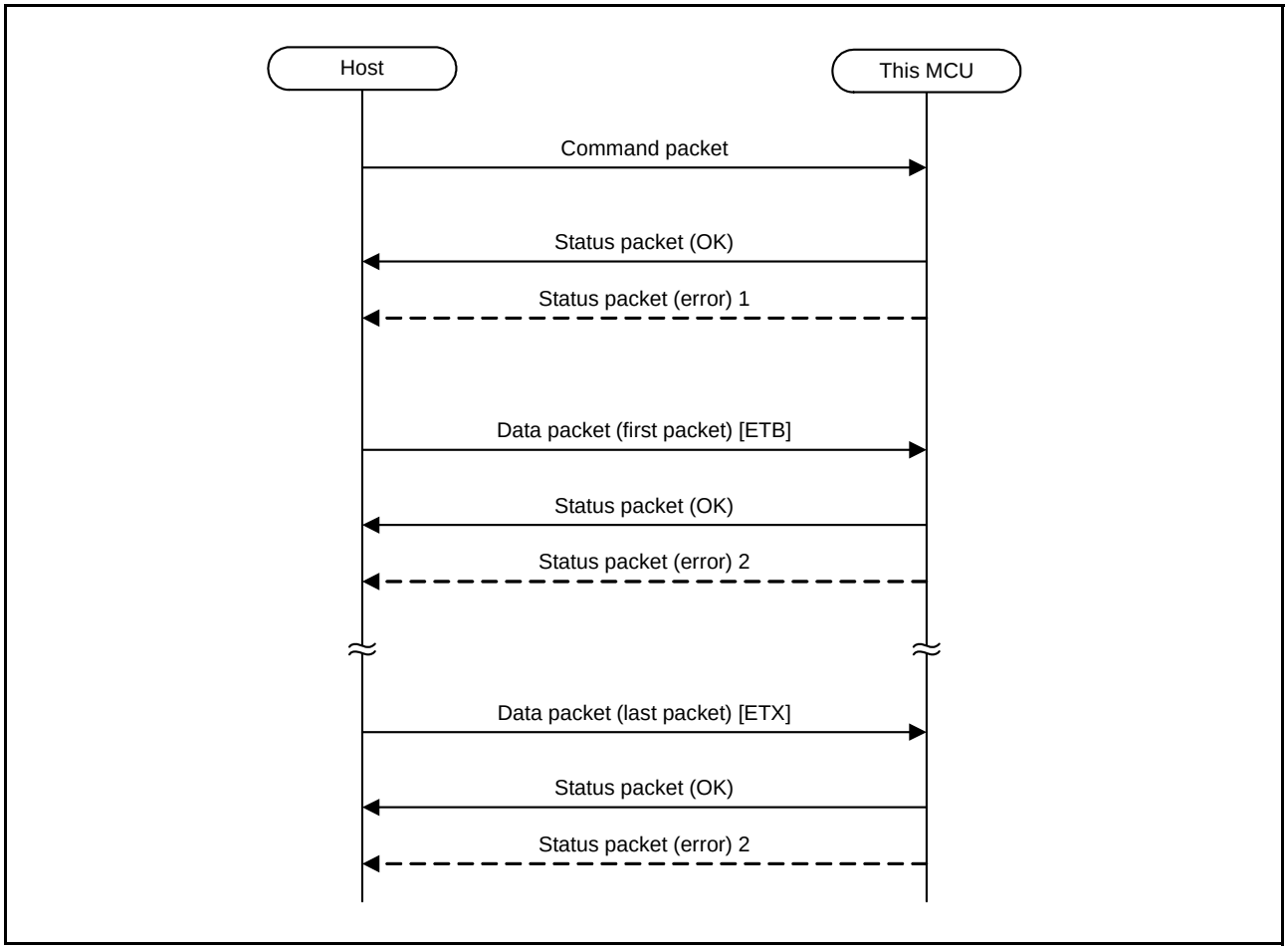


Figure 63.15 Programming Command Transfer Sequence

(2) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	EHH	EHL	ELH	ELL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM: 13h
SHH: Program start address (24 to 31 bits)
SHL: Program start address (16 to 23 bits)
SLH: Program start address (8 to 15 bits)
SLL: Program start address (0 to 7 bits)
EHH: Program end address (24 to 31 bits)
EHL: Program end address (16 to 23 bits)
ELH: Program end address (8 to 15 bits)
ELL: Program end address (0 to 7 bits)
SUM: Sum of values
ETX: 03h

(3) Data packet structure

SOD	LNH	LNL	RES	Data	SUM	ETB or ETX

SOD: 81h
LNH: Data length + 1 (8 to 15 bits)
LNL: Data length + 1 (0 to 7 bits)
RES: 13h (OK)
Data: Program data
SUM: Sum of values
ETB: 17h
ETX: 03h

(4) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 13h (OK)
SUM: ECh
ETX: 03h

(5) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 93h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D0h (address error)
DAh (protection error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(6) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 93h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
E2h (program error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.20 Read Command

This command is used to read data from the specified area in the flash memory and send it to the host.
Specify addresses for a data length that is a multiple of 256 bytes for the user area or user boot area, or a multiple of 16 bytes for the data area. Furthermore, as the address where reading is to start, specify an address on a 256-byte boundary for the user area or user boot area or on a 16-byte boundary for the data area. When the TM function is enabled, reading blocks 8 and 9 in the code flash memory being handled as TM target areas returns 0.
This command cannot be used when read commands are prohibited.
This command can only be accepted in the command waiting phase.

(1) Procedure

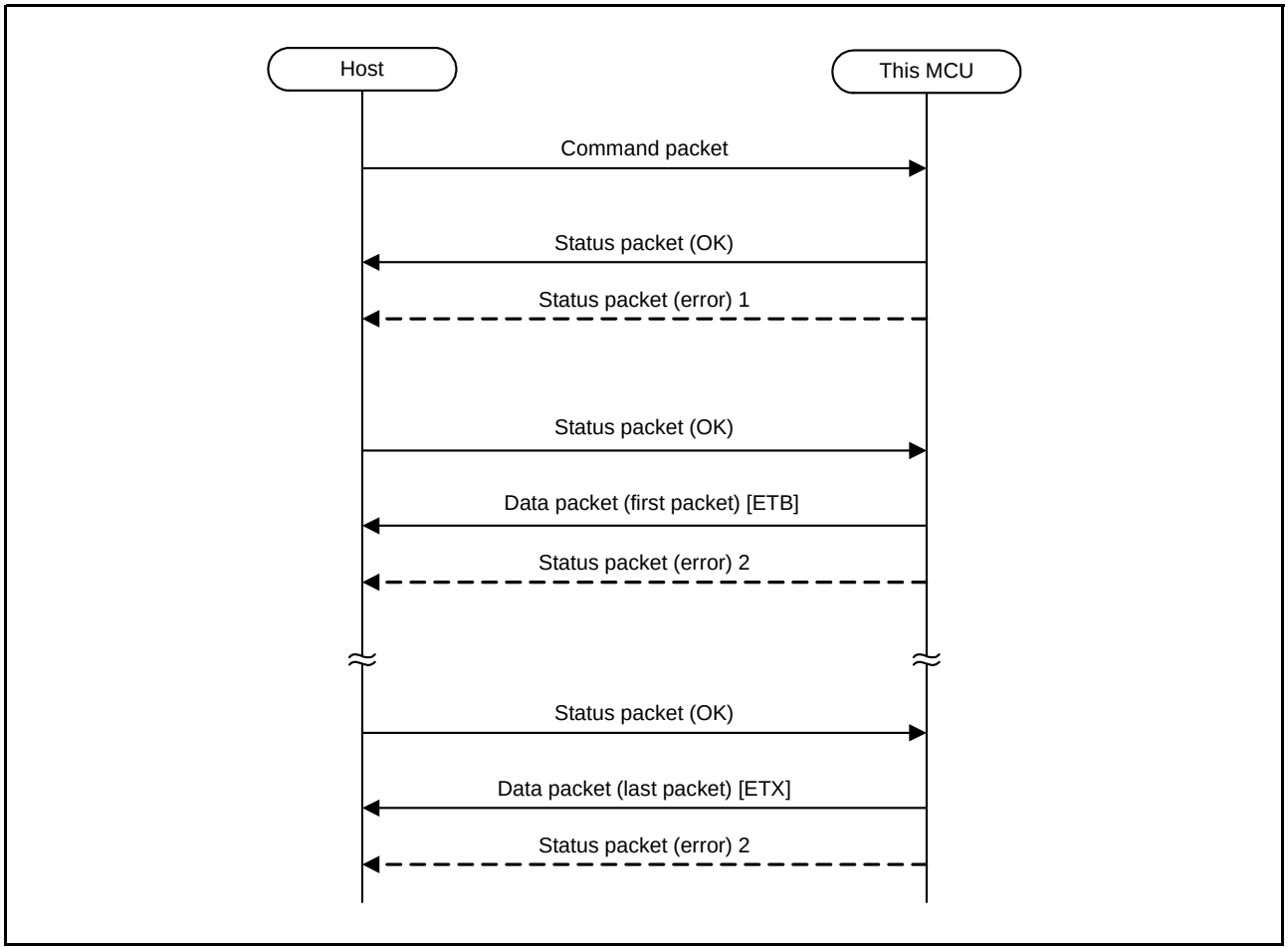


Figure 63.16 Read Command Transfer Sequence

(2) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	EHH	EHL	ELH	ELL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM: 15h
SHH: Read start address (24 to 31 bits)
SHL: Read start address (16 to 23 bits)
SLH: Read start address (8 to 15 bits)
SLL: Read start address (0 to 7 bits)
EHH: Read end address (24 to 31 bits)
EHL: Read end address (16 to 23 bits)
ELH: Read end address (8 to 15 bits)
ELL: Read end address (0 to 7 bits)
SUM: Sum of values
ETX: 03h

(3) Data packet structure

SOD	LNH	LNL	RES	Data	SUM	ETB or ETX

SOD: 81h
LNH: Data length + 1 (8 to 15 bits)
LNL: Data length + 1 (0 to 7 bits)
RES: 15h (OK)
Data: Read data
SUM: Sum of values
ETB: 17h
ETX: 03h

(4) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 15h (OK)
SUM: EAh
ETX: 03h

(5) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 95h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D0h (address error)
DAh (protection error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(6) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 95h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.21 Lock-Bit Setting Command

This command is used to set lock bits for the user area.
When the chip starts up in boot mode, programming or erasure of blocks for which the lock bit has been set is possible since the lock bit function is disabled. To enable the lock bit function during operation in boot mode, use the lock-bit enable command.
This command cannot be used to clear a lock bit that has already been set. Canceling the setting of a lock bit requires erasure of the target block.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	LB1	LB2	LBU	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 63h
COM: 22h
LB1: User area (65 bytes)
LB2: Reserved data (32 bytes)
LBU: Reserved data (1 byte)
SUM: Sum of values
ETX: 03h

[Lock-bit setting]

(0: Set the lock bit; 1: Do not set the lock bit.)

	b7	b6	b5	b4	b3	b2	b1	b0
1st byte	Block 7	Block 6	Block 5	Block 4	Block 3	Block 2	Block 1	Block 0

⋮

	b7	b6	b5	b4	b3	b2	b1	b0
65th byte	Fixed to 1	Fixed to 1	Block 517	Block 516	Block 515	Block 514	Block 513	Block 512
66th byte	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1

⋮

	b7	b6	b5	b4	b3	b2	b1	b0
97th byte	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1
98th byte	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1

Set the value to 1 for blocks that do not exist.

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 22h (OK)
SUM: DDh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A2h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
E2h (program error)
DDh (Lock-bit setting error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.22 Lock-Bit Acquisition Command

This command is used to make the MCU send the lock bit information for the user area to the host.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 23h
SUM: DCh
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	LB1	LB2	LBU	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 63h
RES: 23h (OK)
LB1: User area (65 bytes)
LB2: Reserved data (32 bytes)
LBU: Reserved data (1 byte)
SUM: Sum of values
ETX: 03h

[Lock-bit setting]

(0: The lock bit has been set; 1: The lock bit has not been set.)

	b7	b6	b5	b4	b3	b2	b1	b0
1st byte	Block 7	Block 6	Block 5	Block 4	Block 3	Block 2	Block 1	Block 0

⋮

	b7	b6	b5	b4	b3	b2	b1	b0
65th byte	Fixed to 1	Fixed to 1	Block 517	Block 516	Block 515	Block 514	Block 513	Block 512
66th byte	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1

⋮

	b7	b6	b5	b4	b3	b2	b1	b0
97th byte	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1
98th byte	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1	Fixed to 1

The value is set to 1 for blocks that do not exist.

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 23h (OK)
SUM: DCh
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A3h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A3h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.23 Lock-Bit Enable Command

This command can be used to enable the lock-bit function.
When the function is enabled, blocks for which the lock bit is set cannot be programmed or erased.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 24h
SUM: DBh
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 24h (OK)
SUM: DBh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A4h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.24 Lock-Bit Disable Command

This command can be used to disable the lock-bit function.
When the function is disabled, blocks for which the lock bit is set can be programmed or erased.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 25h
SUM: DAh
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 25h (OK)
SUM: DAh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A5h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.25 Command Protection Setting Command

This command is used to prohibit block erase commands, programming commands, and read commands.
Command protection cannot be disabled if it is already enabled. This command cannot be used when ID authentication in boot mode is enabled.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SEC	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 02h
COM: 20h
SEC: Security data (1 byte)
- Bit 7: Prohibition of block erase commands (1: Disabled; 0: Enabled)
- Bit 6: Prohibition of programming commands (1: Disabled; 0: Enabled)
- Bit 5: Prohibition of block erase commands (1: Disabled; 0: Enabled)
- Bit 4: Reserved bit (fixed to 1)
- Bit 3: Reserved bit (fixed to 1)
- Bit 2: Reserved bit (fixed to 1)
- Bit 1: Reserved bit (fixed to 1)
- Bit 0: Reserved bit (fixed to 1)

SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 20h (OK)
SUM: DFh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A0h (error)
ERR: Error code
- C1h (packet error)
- C2h (checksum error)
- C3h (flow error)
- DAh (protection error)
- E1h (erase error)
- E2h (program error)

SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.26 Command Protection Acquisition Command

This command is used to make the MCU send information on the settings for prohibition of block erase commands, programming commands, and read commands to the host. This command cannot be used when ID authentication in boot mode is enabled.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 21h
SUM: DEh
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	SEC	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 21h (OK)
SEC: Security data (1 byte)
- Bit 7: Prohibition of block erase commands (1: Disabled; 0: Enabled)
- Bit 6: Prohibition of programming commands (1: Disabled; 0: Enabled)
- Bit 5: Prohibition of block erase commands (1: Disabled; 0: Enabled)
- Bit 4: Reserved bit (fixed to 1)
- Bit 3: Reserved bit (fixed to 1)
- Bit 2: Reserved bit (fixed to 1)
- Bit 1: Reserved bit (fixed to 1)
- Bit 0: Reserved bit (fixed to 1)

SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 21h (OK)
SUM: DEh
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A1h (error)
ERR: Error code
- C1h (packet error)
- C2h (checksum error)
- C3h (flow error)

SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A1h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.27 Serial Programming ID Code Setting Command

This command is used to set the ID code in boot mode and by the on-chip debugger (OCD) to enable ID authentication in boot mode.
After this command is issued, ID authentication is required for reconnection in boot mode.
This command cannot be used if block erase commands, programming commands, or read commands are prohibited.
Apply a reset after executing this command.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	IDC	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 11h
COM: 28h
IDC: ID code (16 bytes)*1
SUM: Sum of values
ETX: 03h

Note 1. Send the values as below.
<ID code>
ID = 128'h0F0E0D0C0B0A09080706050403020100
(ID0:00h, ID1:01h, ID2:02h, ... , ID15:0Fh)
<Data for transmission>

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	…
00h	01h	02h	03h	04h	05h	06h	07h	…

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 28h (OK)
SUM: D7h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A8h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
DAh (protection error)
E1h (erase error)
E2h (program error)
E3h (verify error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.28 ID Code Setting Command

This command is used to set the ID code for the on-chip debugger (OCD).
Setting the ID code with this command does not enable ID authentication in boot mode.
This command cannot be used when ID authentication in boot mode is enabled, or when block erase commands, programming commands, or read commands are prohibited.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	IDC	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 11h
COM: 2Ah
IDC: ID code (16 bytes)*1
SUM: Sum of values
ETX: 03h

Note 1. Send the values as below.
<ID code>
ID = 128'h0F0E0D0C0B0A09080706050403020100
(ID0:00h, ID1:01h, ID2:02h, ... , ID15:0Fh)
<Data for transmission>

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	…
00h	01h	02h	03h	04h	05h	06h	07h	…

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 2Ah (OK)
SUM: D5h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: AAh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
DAh (protection error)
E1h (erase error)
E2h (program error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.29 ID Code Acquisition Command

This command is used to make the MCU send the ID code for the on-chip debugger (OCD) to the host.
This command cannot be used when ID authentication in boot mode is enabled, or when read commands are prohibited.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 2Bh
SUM: D4h
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	IDC	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 11h
RES: 2Bh (OK)
IDC: ID code (16 bytes) *1
SUM: Sum of values
ETX: 03h

Note 1. The values are sent as below.
<ID code>
ID = 128'h0F0E0D0C0B0A09080706050403020100
(ID0:00h, ID1 = 01h, ID2 = 02, ... ID15 = 0Fh)
<Data for transmission>

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	…
00h	01h	02h	03h	04h	05h	06h	07h	…

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 2Bh (OK)
SUM: D4h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: ABh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: ABh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
DAh (protection error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.30 Serial Programmer Connection Prohibition Command

This command is used to prohibit the connection of serial programmers.
Applying a reset on execution of this command disables the connection of serial programmers.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 29h
SUM: D6h
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 29h (OK)
SUM: D6h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: A9h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
E1h (erase error)
E2h (program error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.31 OFS Setting Command

This command is used to set the values in the OFS0 and OFS1 registers.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	OFS	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM: 48h
OFS: OFS (8 bytes)*1
SUM: Sum of values
ETX: 03h

Note 1. Send the values as below.
< OFS >
OFS0 register = 01234567h
OFS1 register = 89ABCDEFh
<Data for transmission>

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
67h	45h	23h	01h	EFh	CDh	ABh	89h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 48h (OK)
SUM: B7h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: C8h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
E1h (erase error)
E2h (program error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.32 OFS Acquisition Command

This command is used to make the MCU send the values of the OFS0 and OFS1 registers to the host.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 49h
SUM: B6h
ETX: 03h

(2) Data packet structure (data packet)

SOD	LNH	LNL	COM	OFS	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 09h
RES: 49h (OK)
OFS: OFS (8 bytes)*1
SUM: Sum of values
ETX: 03h

Note 1. The values are sent as below.
< OFS >
OFS0 register = 01234567h
OFS1 register = 89ABCDEFh
<Data for transmission>

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
67h	45h	23h	01h	EFh	CDh	ABh	89h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 49h (OK)
SUM: B6h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: C9h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: C9h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.33 Endian Setting Command

This command is used to set the values in the MDE register.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	OFS	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 02h
COM: 4Ah
END: Endian information
00h (big-endian)
Other than the above (little-endian)
SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 4Ah (OK)
SUM: B5h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CAh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
E1h (erase error)
E2h (program error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.34 Endian Acquisition Command

This command is used to make the MCU send information on the setting of the MDE register to the host.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 4Bh
SUM: B4h
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	END	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 4Bh (OK)
END: Endian information
00h (big-endian)
FFh (little-endian)
SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 4Bh (OK)
SUM: B4h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CBh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CBh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.35 Configuration Clearing Command

This command is used to clear configuration data. When the TM function is enabled, the command can also be used to erase blocks 8 and 9 in the code flash memory being handled as TM target areas.
For the states after execution of this command, see (4) Configuration data after clearing.
This command cannot be issued in the following cases: the user area, user boot area, and data area are not blank; the lock bit is set for any block; or prohibition of block erase commands is enabled.
When ID authentication in boot mode is enabled, a reset is required to disable ID authentication.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 1Ch
SUM: E3h
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 1Ch (OK)
SUM: E3h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 9Ch (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
DAh (protection error)
E0h (non-blank error)
E1h (erase error)
E2h (program error)
SUM: Sum error
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(4) Configuration data after clearing

Data	Setting
Prohibition of block erase commands	Disabled
Prohibition of programming commands	Disabled
Prohibition of read commands	Disabled
ID authentication in boot mode	Disabled
Prohibition of the connection of serial programmers	Disabled
ID code	All FFh
MDE	Little-endian
OFS0 and OFS1	All FFh
TM function	Disabled

63.12.36 TM Setting Command

The TM setting command writes 000b to the TMEF.TMEF[2:0] bits and the desired 4-byte value to the TMINF register to enable the TM function. The TMINF register is used to store codes for identification of the programs, etc., in the areas being handled as TM. The TM function is enabled after a reset.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	TMI	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 05h
COM: 4Eh
TMI: TMINF (4 bytes)*1
SUM: Sum of value
ETX: 03h

Note 1. Send the values as below.
< TMINF >
TMINF = 01234567h

<Data for transmission>

1st byte	2nd byte	3rd byte	4th byte
67h	45h	23h	01h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 4Eh (OK)
SUM: B1h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CEh (error)
ERR: Error code
C0h (non-supported command error)
C1h (packet error)
C2h (checksum error)
C3h (flow error)
DAh (protection error)
E1h (erase error)
E2h (write error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.37 TM Acquisition Command

The TM acquisition command acquires an indicator of whether the TM function is enabled or disabled, the setting of the TMINF register, and the addresses where the areas being handled as TM target start and end, and conveys the results to the programmer.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 4Fh
SUM: B0h
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	TME	TMI	SHH	SHL	SLH	SLL
EHH	EHL	ELH	ELL	SUM	EXT				

SOD: 81h
LNH: 00h
LNL: 0Eh
RES: 4Fh (OK)
TME: TM function is enabled or disabled
00h (TM function is enabled)
FFh (TM function is disabled)
TMI: TMINF (4 bytes*1)
SHH: TM target area start addresses (high and high)
SHL: TM target area start addresses (high and low)
SLH: TM target area start addresses (low and high)
SLL: TM target area start addresses (low and low)
EHH: TM target area end addresses (high and high)
EHL: TM target area end addresses (high and low)
ELH: TM target area end addresses (low and high)
ELL: TM target area end addresses (low and low)
SUM: Sum of values
ETX: 03h

Note 1. Send the values as below.
< TMINF >
TMINF = 01234567h

<Data for transmission>

1st byte	2nd byte	3rd byte	4th byte
67h	45h	23h	01h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 4Fh (OK)
SUM: B0h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CEh (error)
ERR: Error code
C0h (non-supported command error)
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CEh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.38 Simple Addition Checksum Command

This command is used to calculate the sum of values in the specified area and send the result to the host. While the TM function is enabled, however, the values in the TM target areas are not included in the calculation.
The target area of this command can be selected from the user area, user boot area, and data area. Calculation is by simple addition. The initial value is 0 and the sum of values in the specified area is obtained by adding the values of all bytes.
If this command is issued for a data area that includes erased blocks, the result is undefined. When a simple addition checksum is to be executed for the data area, make sure that data have been written throughout the specified area.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	ARE	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 02h
COM: 4Dh
ARE: Area information
- 00h (user area)
- 10h (user boot area)
- 20h (data area)

SUM: Sum of values
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	SD1	SD2	SD3	SD4	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 05h
RES: 4Dh (OK)
SD1: Sum of values
SD2: Sum of values
SD3: Sum of values
SD4: Sum of values
- If the sum of values is 01234567h, the settings are as follows.
- SD1 = 01h
- SD2 = 23h
- SD3 = 45h
- SD4 = 67h

SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 4Dh (OK)
SUM: B2h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CDh (error)
ERR: Error code
- C1h (packet error)
- C2h (checksum error)
- C3h (flow error)
- D5h (area error)

SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CDh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.39 Signature Acquisition Command

This command is used to make the MCU send information on the flash memory configuration to the host.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 3Ah
SUM: C5h
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	DEV	CF1	CS1	CN1	CF2	CS2	CN2		
			SP1	SS1	SN1	CF3	CSS	CN3	UF1	US1	UN1	
								DF1	DS1	DN1	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 3Bh
RES: 3Ah (OK)
DEV: Reserved data (16 bytes)
CF1: 00h (8-Kbyte-block portion of the user area)
CS1: Size of the 8-Kbyte-block portion of the user area
(in bytes; 4-byte value)
CN1: Number of 8-Kbyte blocks in the user area (2 bytes)
CF2: 00h (32-Kbyte-block portion of the user area)
CS2: Size of the 32-Kbyte-block portion of the user area
(in bytes; 4-byte value)
CN2: Number of 32-Kbyte blocks in the user area (2 bytes)
SP1: Reserved data (1 byte)
SS1: Reserved data (4 bytes)
SN1: Reserved data (2 bytes)
CF3: Reserved data (1 byte)
CS3: Reserved data (4 bytes)
CN3: Reserved data (2 bytes)
UF1: 02h (user boot area)
US1: Block size of the user boot area (in bytes; 4-byte value)
UN1: Number of blocks in the user boot area (2 bytes)
DF1: 03h (data area)
DS1: Block size of the data area (in bytes; 4-byte value)
DN1: Number of blocks in the data area (2 bytes)
SUM: Sum of values
ETX: 03h

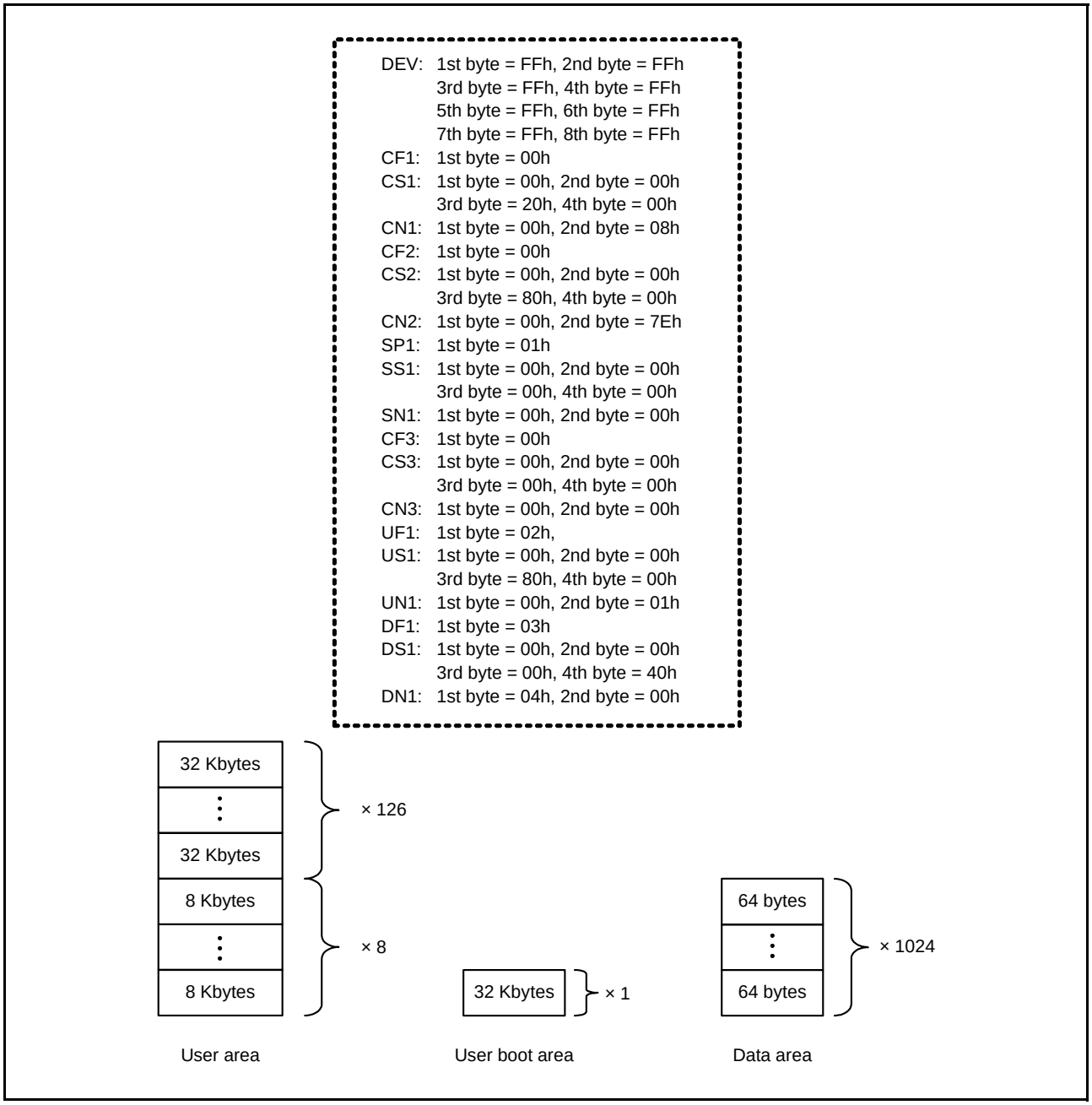


Figure 63.17 Example of Flash Memory Configuration Information

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 3Ah (OK)
SUM: C5h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: BAh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: BAh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

63.12.40 Usage Example

(1) Example of the Procedure for Reprogramming

Figure 63.18 shows an example of the procedure for reprogramming.

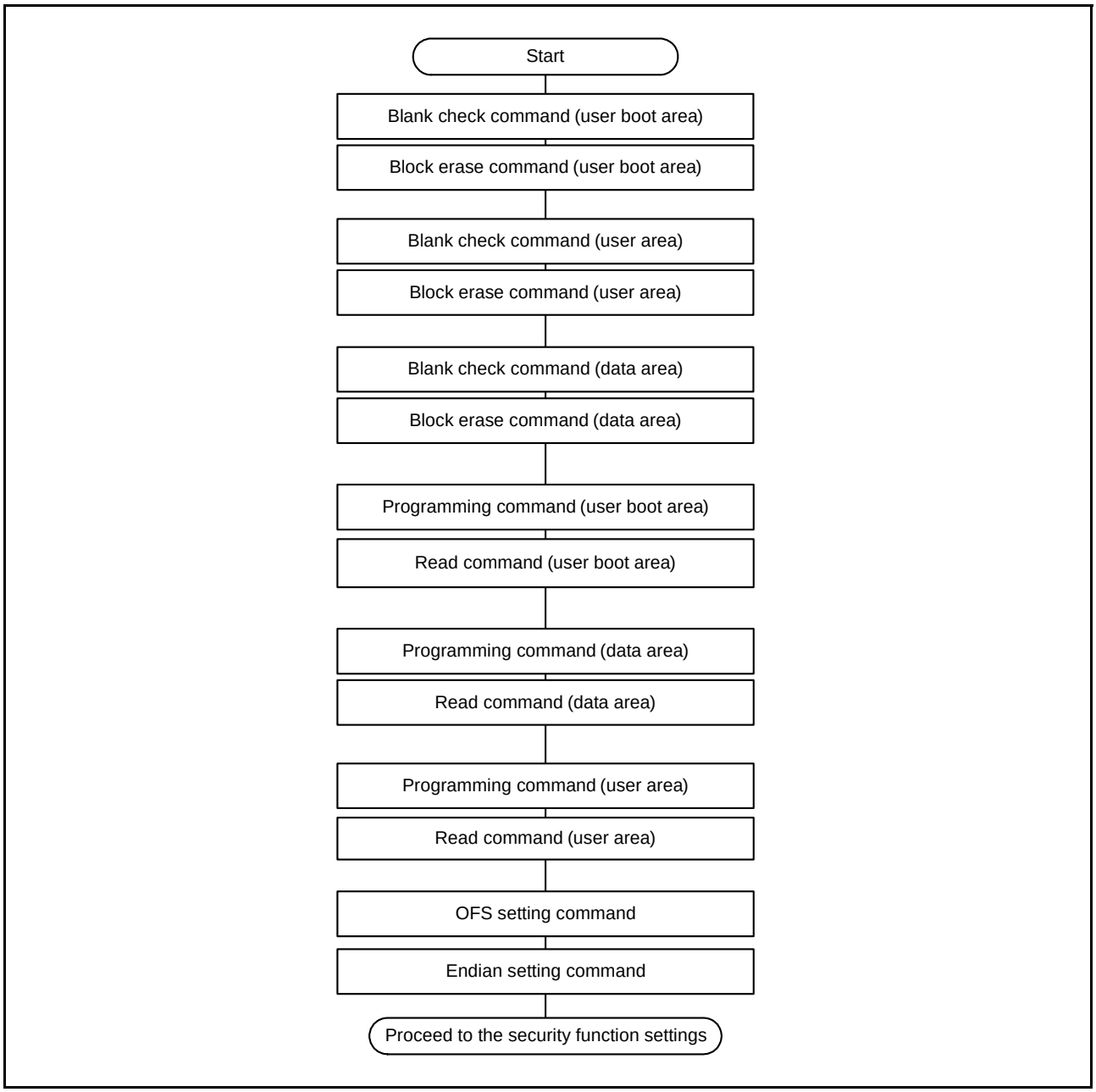
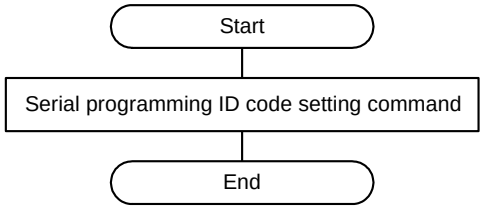


Figure 63.18 Example of Reprogramming Procedure

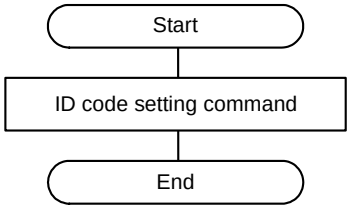
(2) Security Function Settings

When the security functions are to be used, follow any of the procedures below to set the required security functions.

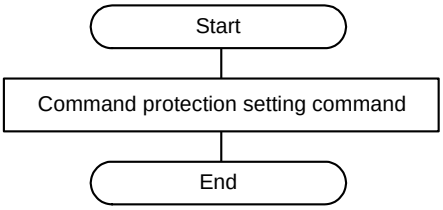
1. Setting the OCD ID code and enabling ID authentication in boot mode



2. Setting the OCD ID code



3. Prohibiting block erase commands, programming commands, and read commands



4. Prohibiting the connection of serial programmers

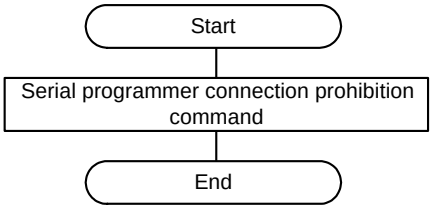


Figure 63.19 Settings for Use of the Security Functions

63.12.41 Flow for Erasure when Programming Commands are Prohibited

When programming commands are prohibited, follow the procedure below for erasure.

1. Erase the user boot area.
2. Erase the data area in order from the first block.
3. Erase the user area in order from the last block.

Note that neither a reset nor interrupts, including NMI, should be applied during the steps from erasure of the user boot area to erasure of the user area.

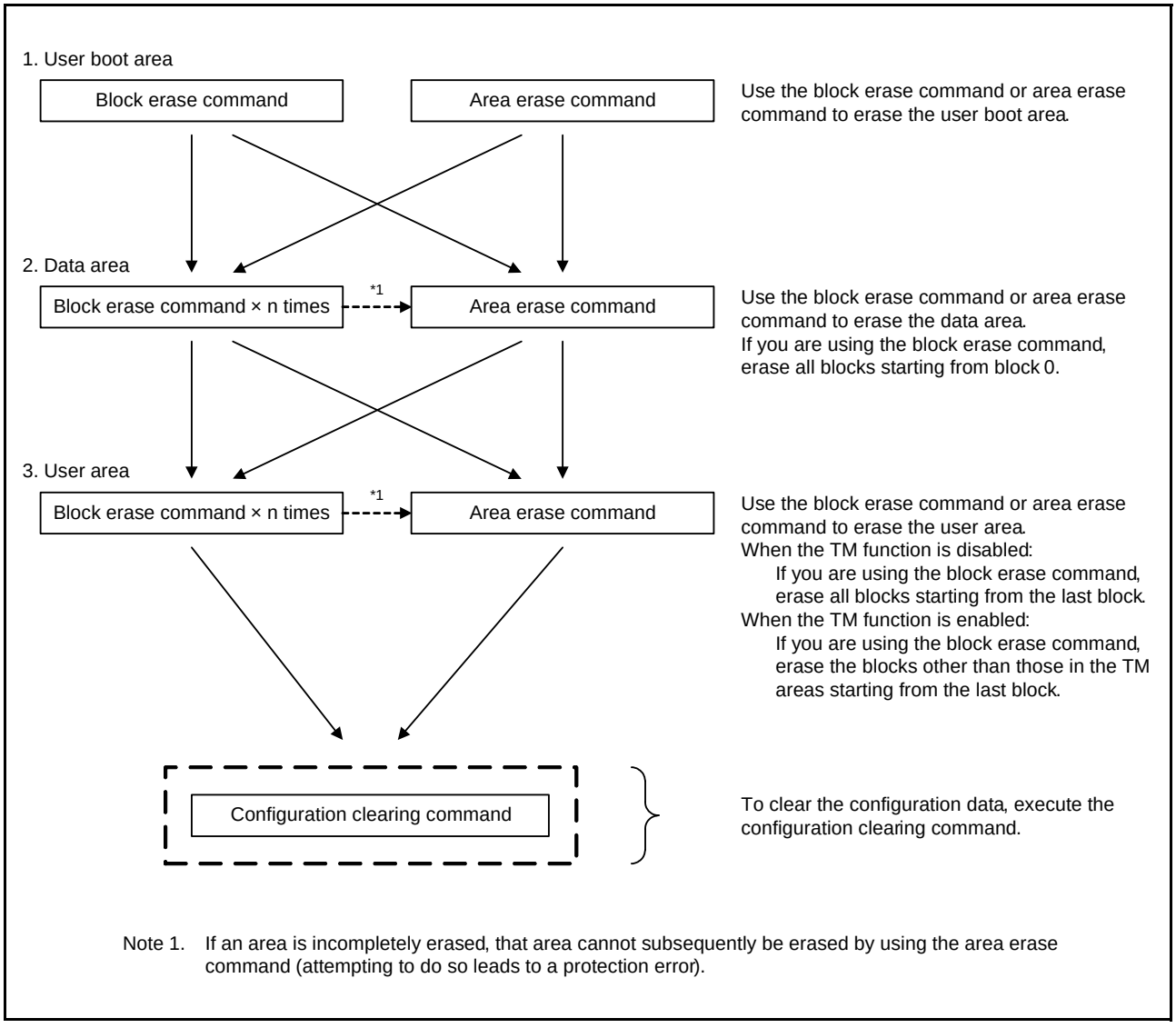


Figure 63.20 Flow for Erasure when Programming Commands are Prohibited

63.13 Using the Serial Programmer for Rewriting

A dedicated flash memory programmer can be used to rewrite flash memory in boot mode.

(1) Serial Programming

A flash programmer can program this MCU by providing a connector to the board while the MCU is implemented to the board that the user will use.

63.13.1 Environments for Serial Programming

The recommended environments for rewriting the flash memory of the MCU with data are described below.

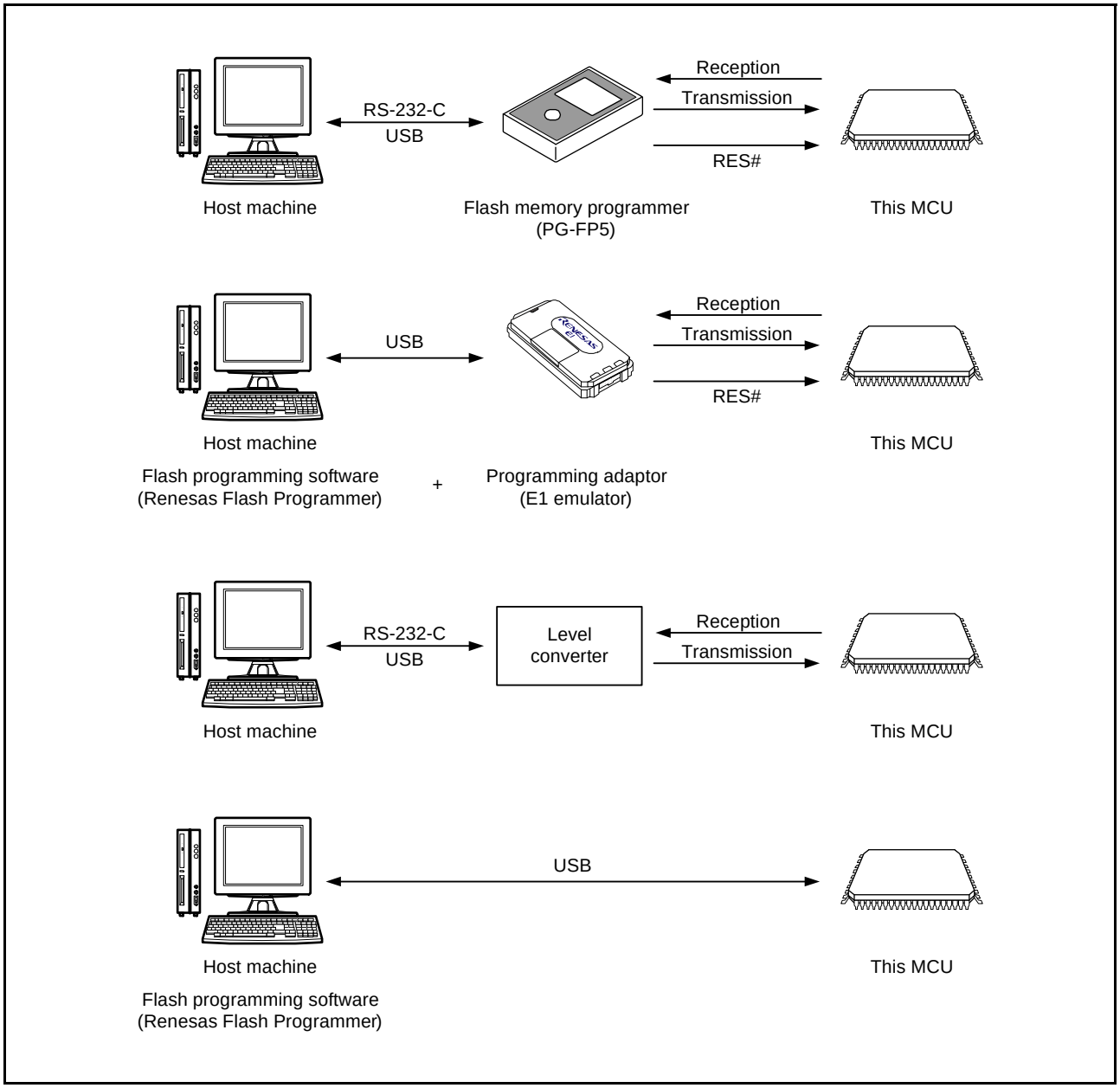


Figure 63.21 Environments for Rewriting Programs to the Flash Memory

By using the PG-FP5 flash memory programmer or the combination of the Renesas Flash Programmer (software for programming to flash memory) running on the host machine and the E1 emulator as an programming adaptor, the user is easily able to program, erase, and verify the contents of the on-chip memory of flash-memory-equipped microcontrollers from Renesas Electronics.
The PG-FP5 flash memory programmer handles rewriting the flash memory from a host machine or in stand-alone mode.
The flash programming software (Renesas Flash Programmer) handles rewriting the flash memory from a host machine.

Note: For details of the PG-FP5, refer to the PG-FP5 Flash Memory Programmer User's Manual; for details of the Renesas Flash Programmer flash programming software, refer to the Renesas Flash Programmer Flash Programming Software User's Manual.

63.14 Programming through Self-Programming

63.14.1 Overview

This MCU supports programming of the flash memory by the user program itself. The FACI commands can be used with user programs for writing to the code flash memory and to the data flash memory. This allows upgrading of user programs and overwriting of constant data fields.
When the data flash memory is programmed, the background operation facility makes it possible to execute a programming program from the code flash memory to program the data flash memory. Furthermore, the programming program can be copied to internal RAM or external memory in advance of the programming operation, and executed from the given destination to perform the programming.
Background operation is also available for use when the address ranges of the area of code flash memory to be programmed and the area of code flash memory to be read satisfy particular conditions (see Table 63.16). At the time of self-programming in this case, a programming program in code flash memory can be used to program the code flash memory. Also, the programming program can be copied to internal RAM or external memory in advance of the programming operation, and executed from the given destination to program the code flash memory. This is useful when the address ranges do not satisfy the conditions for background operation.

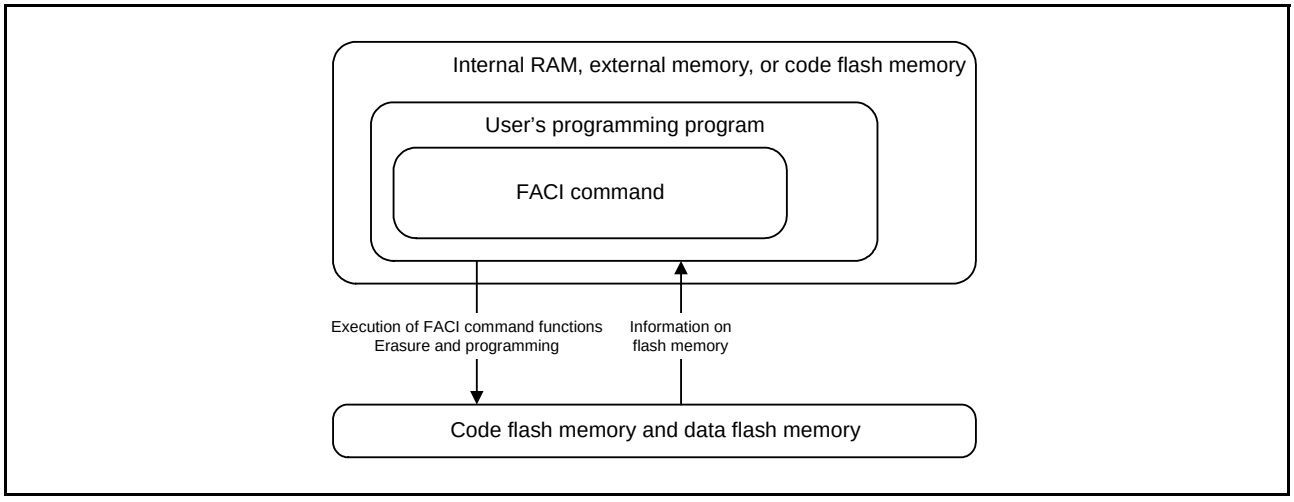


Figure 63.22 Schematic View of Self-Programming

For comprehensive information on the self-programming of flash memory, refer to Flash Memory User's Manual: Hardware Interface.

63.14.2 Background Operation

Background operations can be used when the combination of the flash memory for rewriting and the flash memory for reading is any of those listed below.

Table 63.16 Conditions under which Background Operation is Usable

	Range for Rewriting	Range for Reading
Common to all products	Data flash memory	Code flash memory
2.5-Mbyte product	First half (0.5 Mbytes) of the user area of the code flash memory (addresses FFD8 0000h to FFDF FFFFh)	Second half (2 Mbytes) of the user area of the code flash memory (addresses FFE0 0000h to FFFF FFFFh) or user boot area of the code flash memory
	Second half (2 Mbytes) of the user area of the code flash memory (addresses FFE0 0000h to FFFF FFFFh)	First half (0.5 Mbytes) of the user area of the code flash memory (addresses FFD8 0000h to FFDF FFFFh)
3-Mbyte product	First half (1 Mbyte) of the user area of the code flash memory (addresses FFD0 0000h to FFDF FFFFh)	Second half (2 Mbytes) of the user area of the code flash memory (addresses FFE0 0000h to FFFF FFFFh) or user boot area of the code flash memory
	Second half (2 Mbytes) of the user area of the code flash memory (addresses FFE0 0000h to FFFF FFFFh)	First half (1 Mbyte) of the user area of the code flash memory (addresses FFD0 0000h to FFDF FFFFh)
4-Mbyte product	First half (2 Mbytes) of the user area of the code flash memory (addresses FFC0 0000h to FFDF FFFFh)	Second half (2 Mbytes) of the user area of the code flash memory (addresses FFE0 0000h to FFFF FFFFh) or user boot area of the code flash memory
	Second half (2 Mbytes) of the user area of the code flash memory (addresses FFE0 0000h to FFFF FFFFh)	First half (2 Mbytes) of the user area of the code flash memory (addresses FFC0 0000h to FFDF FFFFh)

63.15 Reading Flash Memory

63.15.1 Reading Code Flash Memory

Special settings are not required to read code flash memory in normal mode or user boot mode. Data can simply be read out through access to addresses in the code flash memory.
When reading code flash memory that has been erased but not yet been programming again (i.e. that is in the non-programmed state), all bits are read as 1.

63.15.2 Reading Data Flash Memory

Special settings are not required to read data flash memory in normal mode or user boot mode. Data can simply be read out through access to addresses in the data flash memory.
Values read from data flash memory that has been erased but not yet been programming again (i.e. that is in the non-programmed state) are undefined. Use blank checking when you need to confirm that an area is in the non-programmed state.

63.16 Trusted Memory

This MCU has a trusted memory (hereafter called TM) facility to prevent the reading of blocks 8 and 9 in the code flash memory by third party software. When set as trusted memory, these areas are suitable for storing software that handles processing for encryption algorithms, device control software that is associated with proprietary or confidential know-how, purchased middleware, and so on.

Table 63.17 lists the specifications of the TM, Table 63.18 lists access restrictions within the TM target area when TM is enabled, and Figure 63.23 shows the cases where the CPU is able to operate in relation to the TM target area.

Table 63.17 TM Specifications

Item	Description
TM target area	Blocks 8 and 9 in the code flash memory (64 Kbytes in total)
Access restrictions when TM is enabled	See Table 63.18, Restrictions on Access to the TM Area while TM is Enabled.
How to run program code when the TM function is enabled	When TM is enabled, starting to run program code in an area being handled as TM is only possible with a branch instruction from program code outside the areas being handled as TM.
Interrupt processing during the execution of program code in an area being handled as TM while the TM function is enabled	Both the acceptance of requests for interrupt processing and return from interrupt processing are possible.
Security function	Enabling the TM function restricts access to program code in the areas for handling as TM to instruction fetching only
Protection functions	• Restrictions on access by a program running in an area other than the areas being handled as TM to data in those areas while the TM function is enabled*1 • Once enabled, the TM function prevents its own disabling until the areas being handled as TM are erased. • Once enabled, the TM function prevents further writing to the areas being handled as TM

Note 1. Access to data in operations that include the borders of the TM areas is also not allowed.

Table 63.18 Restrictions on Access to the TM Area while TM is Enabled

Type of Access	CPU	DMAC/DTC/EXDMAC/EDMAC
Instruction fetching	Yes	—
Access to data	No*1	No

Note: The same restrictions apply to on-chip debuggers as to the DMAC in the table above. For the operation of the OCD you are using in relation to the TM area, see the manual of the given OCD.

Note 1. Place data to which access will be required in areas other than the TM areas.

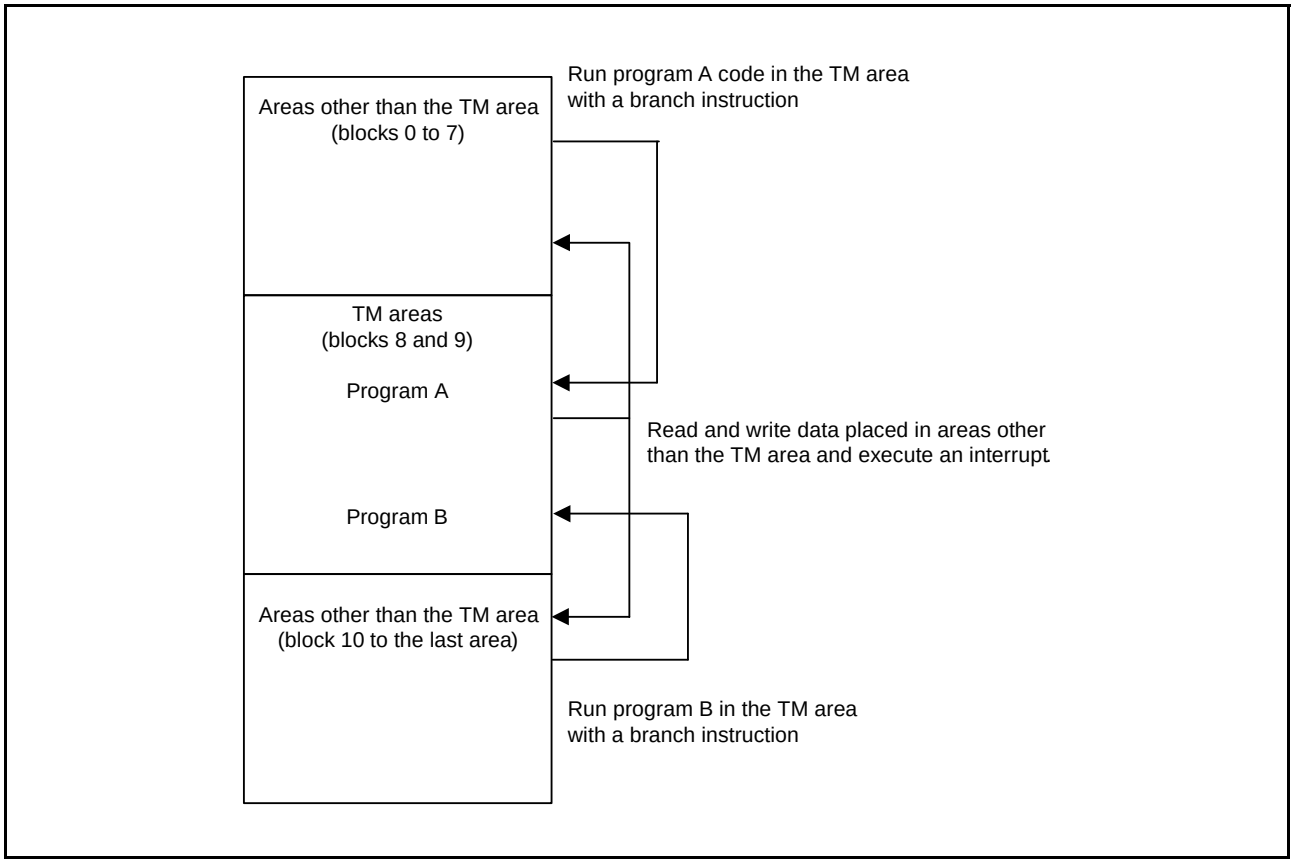


Figure 63.23 Cases where CPU is Allowed to Operate in Relation to the TM Area when the TM Function is Enabled

63.16.1 Allocating Program Code to the TM Area

When the TM function is enabled, implement countermeasures in the form of software for the TM area as required as further measures to prevent the running of programs for access to consecutive addresses in the TM areas from areas outside the TM areas.

63.16.2 How to Enable the TM Function

63.16.2.1 By Self-Programming

After writing to blocks 8 and 9 of the code flash memory, i.e. the target areas for TM, use the configuration setting command of the FACI to enable the TM function.
For the configuration setting command of the FACI, refer to Flash Memory User's Manual: Hardware Interface.
Figure 63.24 is a flowchart of the procedure for enabling the TM function by self-programming.

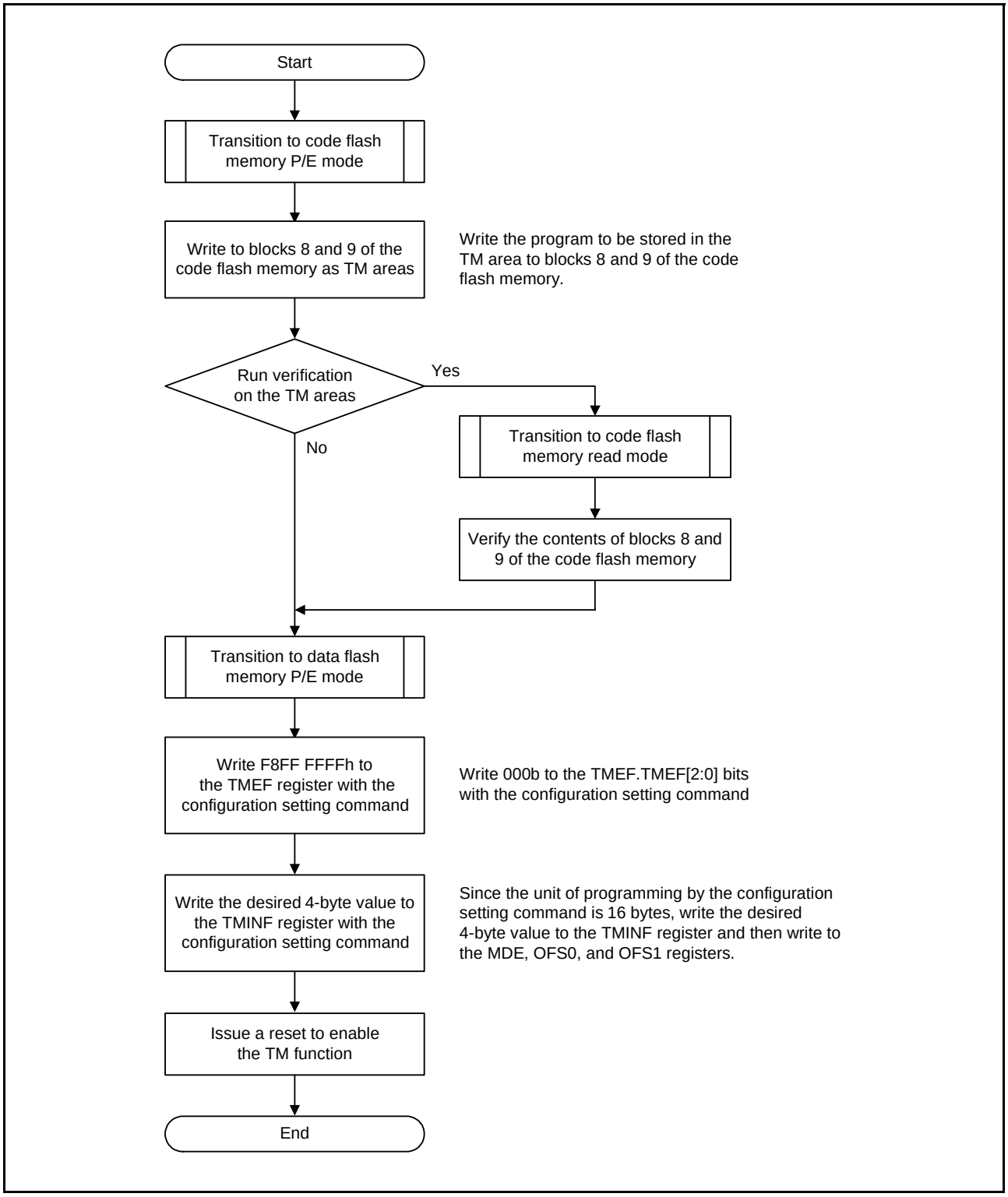


Figure 63.24 Flowchart for Enabling the TM Function by Self-Programming

63.16.3 By Using Boot Mode

In boot mode, use the TM setting command among the boot commands to enable the TM function after writing to blocks 8 and 9 of the code flash memory.

For the TM setting command among the boot commands, see section 63.12.36, TM Setting Command.

Figure 63.25 is a flowchart of enabling the TM function in boot mode.

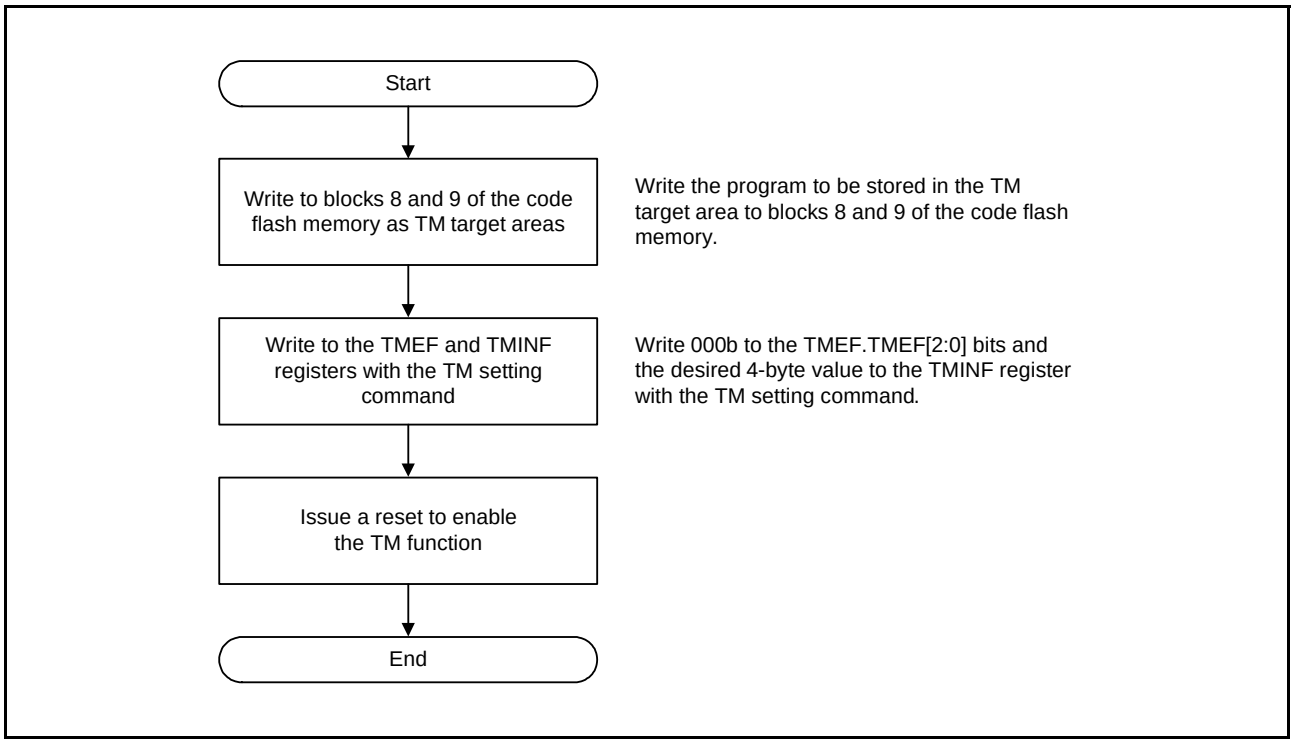
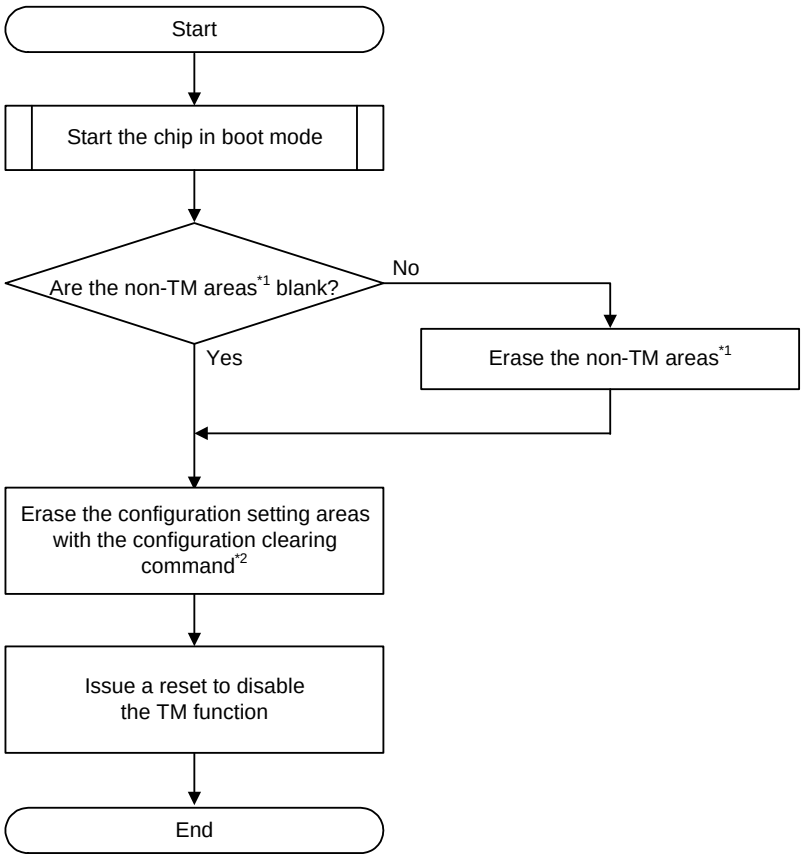


Figure 63.25 Flowchart for Enabling the TM Function by Using Boot Mode

63.16.4 How to Disable the TM function

The TM function cannot be disabled unless the TM target areas are first erased with the configuration clearing command. Do not use the configuration clearing command for this purpose unless the TM function is to be disabled.
Figure 63.26 is a flowchart of disabling the TM function in boot mode.



Note 1. Non-TM areas are the data flash, user boot area, and user area other than blocks 8 and 9 of the user area.
Note 2. Issuing the configuration clearing command leads to erasure of the configuration setting areas, including the TMEF and TMINF registers, as well as the TM areas (blocks 8 and 9 of the code flash memory).

Figure 63.26 Flowchart for Disabling the TM Function in Boot Mode

63.16.5 Notes on Enabling the TM function

63.16.5.1 Protection against Access to the TM Areas

While the TM function is enabled, the only type of access to the TM areas is instruction fetching by the CPU, so do not allocate data to the TM areas.
If the CPU, DMAC, DTC, EXDMAC, or EDMAC, or OCD attempt access to data in the TM areas while the TM function is enabled, values are always read as 0 instead of the actual values.

63.16.5.2 Further Writing to the TM Areas

Further writing to the TM areas is not allowed as long as the TM function is enabled.
If you do need to write further data, follow the procedure described in section 63.16.4, How to Disable the TM function, to disable the TM function, write the modified or extended data to blocks 8 and 9 of the code flash memory, and then follow the procedure described in section 63.16.2, How to Enable the TM Function, to enable the TM function.
To erase the user area, user boot area, and data area, see section 63.12.41, Flow for Erasure when Programming Commands are Prohibited.

63.16.5.3 Executing the Configuration Clearing Command

Follow the procedure in section 63.16.4, How to Disable the TM function, before issuing the configuration clearing command.

63.16.5.4 When the MPU Setting is for Access to the TM Areas

When the TM function is enabled, even if the MPU is set to allow access to the TM target areas, the TM function takes priority.

63.16.5.5 FACI Block Erase Command for the TM Areas

There are no special restrictions on block erasure of the TM areas with the FACI block erase command. Accordingly, since each area corresponds to an erase block, the areas can be erased by issuing the block erase command.

63.16.5.6 Effect on the Setting for Command Protection

Executing the configuration clearing command to disable the TM function also initializes the command protected state at the same time.

63.16.5.7 Conditions for Correct Operation of the TM Function

The TM function operates normally under the conditions prescribed in section 64, Electrical Characteristics.

63.17 Usage Notes

(1) Reading Area Where Programming/Erasure was Interrupted and Area Targeted for Suspension

The data stored in the area where programming or erasure has been suspended or the area where programming or erasure has been suspended by using the suspend command are undefined. To avoid faulty operation caused by reading undefined data, take care not to fetch instructions or read data from areas where programming or erasure was suspended and where programming or erasure was suspended by using the suspend command.

(2) Suspension During Programming/Erasure

When processing of programming/erasure is stopped by issuing the P/E suspend command, the programming/erasure processing can be resumed by issuing the P/E resume command. If the flash sequencer enters the command-locked state for any reason and issues the forced stop command after the suspended processing is normally completed and the ERSSPD flag or PRGSPD flag is set to 1, the suspended processing cannot be resumed. In addition, the values in the area where the processing was suspended are not guaranteed. Erase that area.

(3) Prohibition of Additional Programming

Programming a given area twice is not possible. Erase the area first before programming an area of flash memory after programming to the area has been completed.

(4) Resets During Programming/Erasure or Black Checking

In the case of a reset due to the signal on the RES# pin during programming/erasure or black checking, wait for at least t_{RESWF} (see section 64, Electrical Characteristics) of the reset input period once the operating voltage is within the range stipulated in the electrical characteristics after assertion of the reset signal, and then release the device from the reset state.

(5) Allocation of Vectors for Interrupts and Other Exceptions During Programming/Erasure

Generation of an interrupt or other exception during programming/erasure may lead to fetching of the vector from the code flash memory. If the conditions for using the background operation (BGO) are not satisfied, set the address for vector to an address that is not in the code flash memory.

(6) Abnormal Termination During Programming/Erasure or Blank Checking

Even if programming/erasure ends abnormally due to the generation of a reset by the RES# pin, the programming/erasure state of the flash memory with undefined data cannot be verified or checked. For the area where programming/erasure ends abnormally, the blank check function cannot judge whether the area is erased successfully or not. Erase the area again and confirm that the corresponding area is completely erased before using.
When programming/erasure or blank checking is not completed successfully due to a voltage change that exceeds the operational voltage range, a reset on the RES# pin, the command locked state in response to error detection, or prohibited actions described in (7), the lock bit may be enabled.
In this case, erase the target block while the lock bit is disabled to erase the lock bit.

(7) Items Prohibited During Programming/Erasure or Blank Checking

High voltage is applied to the flash memory during programming/erasure or blank checking. To prevent damage to the flash memory, do not perform the following operations.

- Have the operating voltage from the power supply go beyond the permitted range.
- Change the FWEPROR.FLWE[1:0] bits.
- Change the SYSCR0.ROME bit.
- Change the OPCCR.OPCM[2:0] bits.
- Change the SCKCR.FCK[3:0] and PCLKB[3:0] bits.
- Change the SCKCR3.CKSEL[2:0] bits.

- Change the RSTCKCR.RSTCKEN bit.
- Transition to the all module clock stop mode, software standby mode, or deep software standby mode.

(8) Notes on Program Execution in Boot Mode (for the USB Interface)

- An oscillator is usable in boot mode (USB interface) when its frequency is 20 or 24 MHz and when the setting value of the main clock oscillator driving ability 2 switching bits (MOFCR.MODRV2[1:0]) are 00b in the result of the matching test conducted by the oscillator manufacturer (the recommended setting value).
- To ensure that the power supply is stable during programming/erasure of flash memory, do not connect a cable via a bus-powered HUB.

(9) Programming/Erasure in Low-Speed Operating Modes 1 and 2

Do not programming/erasure the flash memory when low-speed operating mode 1 or 2 is selected with the operating power control register (OPCCR).

64. Electrical Characteristics

64.1 Absolute Maximum Ratings

Table 64.1 Absolute Maximum Rating

Conditions: VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V

Item	Symbol	Value	Unit
Power supply voltage	VCC, VCC_USB	–0.3 to +4.6	V
V_{BATT} power supply voltage	V_{BATT}	–0.3 to +4.6	V
Input voltage (except for ports for 5 V tolerant*1)	V_{in}	–0.3 to VCC + 0.3	V
Input voltage (ports for 5 V tolerant*1)	V_{in}	–0.3 to VCC + 4.6 (≤ 5.8 max.)	V
Reference power supply voltage	VREFH0	–0.3 to AVCC0 + 0.3	V
Analog power supply voltage	AVCC0, AVCC1*2	–0.3 to +4.6	V
USBA power supply voltage	VCC_USBA*2	–0.3 to +4.6	V
USBA analog power supply voltage	AVCC_USBA*2	–0.3 to +4.6	V
Analog input voltage	V_{AN}	–0.3 to AVCC + 0.3	V
Operating temperature	T_{opr}	–40 to +85	°C
Operating temperature (high-temperature products)	T_{opr}	–40 to +105 (Under planning)	°C
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

Note 1. Ports 07, 11 to 17, 20, 21, 30 to 33, 67, and C0 to C3 are 5 V tolerant.

Note 2. Connect the AVCC0, AVCC1, and VCC_USB pins to VCC, and the AVSS0, AVSS1, and VSS_USB pins to VSS.
When the A/D converter unit 0 is not to be used, connect the VREFH0 pin to VCC and the VREFL0 pin to VSS, respectively. Do not leave these pins open.
When the USBA is not to be used, connect the VCC_USBA and AVCC_USBA pins to VCC and the VSS1_USBA, VSS2_USBA, PVSS_USBA, and AVSS_USBA pins to VSS, respectively. Do not leave these pins open.

64.2 DC Characteristics

Table 64.2 DC Characteristics (1)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	IRQ input pin*1 MTU input pin*1 GPT input pin*1 POE3 input pin*1 TPU input pin*1 TMR input pin*1 SCI input pin*1 ADTRG# input pin*1 RES#, NMI	V_{IH}	VCC × 0.8	—	VCC + 0.3	V	
		V_{IL}	–0.3	—	VCC × 0.2		
		ΔV_T	VCC × 0.06	—	—		
	RIIC input pin (except for SMBus)	V_{IH}	VCC × 0.7	—	VCC + 3.6 (≤ 5.8 max.)		
		V_{IL}	–0.3	—	VCC × 0.3		
		ΔV_T	VCC × 0.05	—	—		
	Ports for 5 V tolerant*2	V_{IH}	VCC × 0.8	—	VCC + 3.6 (≤ 5.8 max.)		
		V_{IL}	–0.3	—	VCC × 0.2		
	Other input pins excluding ports for 5 V tolerant*3	V_{IH}	VCC × 0.8	—	VCC + 0.3		
		V_{IL}	–0.3	—	VCC × 0.2		
Input high voltage (except for Schmitt trigger input pin)	MD pin, EMLE	V_{IH}	VCC × 0.9	—	VCC + 0.3	V	
	EXTAL, RSPI input pin, EXDMAC input pin, WAIT#, TCK, SSI input pin, SDHI input pin, MMC input pin, PDC input pin, QSPI input pin		VCC × 0.8	—	VCC + 0.3		
	ETHERC input pin		2.3	—	VCC + 0.3		
	D0 to D31		VCC × 0.7	—	VCC + 0.3		
	RIIC (SMBus)		2.1	—	5.8		
Input low voltage (except for Schmitt trigger input pin)	MD pin, EMLE	V_{IL}	–0.3	—	VCC × 0.1	V	
	EXTAL, RSPI input pin, ETHERC input pin, EXDMAC input pin, WAIT#, TCK, SSI input pin, SDHI input pin, MMC input pin, PDC input pin, QSPI input pin		–0.3	—	VCC × 0.2		
	D0 to D31		–0.3	—	VCC × 0.3		
	RIIC (SMBus)		–0.3	—	0.8		

Note 1. This does not include the pins, which are multiplexed as ports for 5 V tolerant.
Note 2. Ports 07, 11 to 17, 20, 21, 30 to 33, 67, and C0 to C3 are 5 V tolerant.
Note 3. For P32, P31, and P30, input as follows when the V_{BATT} power supply is selected.
VIH Min. = V_{BATT} × 0.8, VIH Max. = V_{BATT} + 0.3, VIL Min. = –0.3, VIL Max. = V_{BATT} × 0.2 (V_{BATT} = 2.0 to 3.6 V)

Table 64.3 DC Characteristics (2)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high voltage	All output pins	V_{OH}	VCC – 0.5	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins (except for RIIC pins and ETHERC output pin)	V_{OL}	—	—	0.5	V	I_{OL} = 1.0 mA
	RIIC output pin		—	—	0.4		I_{OL} = 3.0 mA
			—	—	0.6		I_{OL} = 6.0 mA
	RIIC output pin (only P12 and P13 in channel 0)	V_{OL}	—	—	0.4	V	I_{OL} = 15.0 mA (ICFER.FMPE = 1)
			—	0.4	—		I_{OL} = 20.0 mA (ICFER.FMPE = 1)
	ETHERC output pin	V_{OL}	—	—	0.4	V	I_{OL} = 1.0 mA
Input leakage current	RES#, MD pin, EMLE*1, BSCANP*1, NMI	$\lvert I_{in} \rvert$	—	—	1.0	μA	V_{in} = 0 V V_{in} = VCC
Three-state leakage current (off state)	Other than ports for 5 V tolerant	$\lvert I_{TSI} \rvert$	—	—	1.0	μA	V_{in} = 0 V V_{in} = VCC
	Ports for 5 V tolerant		—	—	5.0		V_{in} = 0 V V_{in} = 5.5 V
Input pull-up MOS current	Ports 0 to 2, 30 to 34, 36, 37, 4 to G, J3, J5	I_p	–300	—	–10	μA	VCC = 2.7 to 3.6 V V_{in} = 0 V
Input pull-down MOS current	EMLE, BSCANP	I_p	10	—	300	μA	V_{in} = VCC
Input capacitance	All input pins (except for ports 03, 05, 12, 13, 16, 17, EMLE, BSCANP, USB0_DP, USB0_DM, USBA_DP, and USBA_DM)	C_{in}	—	—	8	pF	Vbias = 0 V Vamp = 20 mV f = 1 MHz T_a = 25°C
	Ports 03, 05, 12, 13, 16, 17, EMLE, BSCANP, USB0_DP, USB0_DM, USBA_DP, and USBA_DM		—	—	16		

Note 1. The input leakage current value at the EMLE and BSCANP pins are only when V_{in} = 0 V.

Table 64.4 DC Characteristics (3)

Conditions: VCC = AVCC0 = AVCC1 = VREFH0 = VCC_USB = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item				Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Supply current*1	High-speed operating mode	Max.*2		I_{CC}*3	—	—	110	mA	ICLK = 120 MHz PCLKA = 120 MHz PCLKB = 60 MHz PCLKC = 60 MHz PCLKD = 60 MHz FCLK = 60 MHz BCLK = 120 MHz BCLK pin = 60 MHz
		Normal	Peripheral function clock signal supplied*4		—	39	—		
			Peripheral function clock signal stopped*4		—	16	—		
		Coremark	Peripheral function clock signal stopped*4		—	21	—		
		Sleep mode: Supply of the clock signal to peripheral modules is stopped*4				32	61		
		All-module-clock-stop mode (reference value)			—	10	28		
		Increased by BGO operation *5	Reading from the code flash memory while the data flash memory is being programmed		—	7	—		
			Reading from the code flash memory while the code flash memory is being programmed		—	10	—		
	Low-speed operating mode 1: Supply of the clock signal to peripheral modules is stopped*4				—	3	—		All clocks 1 MHz
	Low-speed operating mode 2: Supply of the clock signal to peripheral modules is stopped*4				—	1.2	—		All clocks 32.768 kHz
	Software standby mode				—	0.7	10		
	Deep software standby mode	Power supplied to standby RAM and USB resume detecting unit (USB0 only)			—	22	63	μA	
		Power not supplied to standby RAM and USB resume detecting unit (USB0 only)	Power-on reset circuit and low-power consumption function disabled*6		—	12.5	26		
			Power-on reset circuit and low-power consumption function enabled*7		—	3.1	13.5		
		Increased by RTC operation	When a crystal resonator for low clock loads is in use		—	0.6	—		
			When a crystal resonator for standard clock loads is in use		—	2.0	—		
	RTC operating while VCC is off (with the battery backup function, only the RTC and sub-clock oscillator operate)		When a crystal resonator for low clock loads is in use		—	0.9	—		V_{BATT} = 2.0 V, VCC = 0 V
					—	1.6	—		V_{BATT} = 3.3 V, VCC = 0 V
			When a crystal resonator for standard clock loads is in use		—	1.7	—		V_{BATT} = 2.0 V, VCC = 0 V
					—	3.3	—		V_{BATT} = 3.3 V, VCC = 0 V

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Supply of the clock signal to peripheral modules is stopped in this state. This does not include operations as BGO (background operations).
Note 3. I_{CC} depends on f (ICLK) as follows. (ICLK/PCLKA:PCLKB/PCLKC/PCLKD:BCLK:BCLK pin = 10:5:10:5 when EXTAL = 12 MHz)
I_{CC} Max. = 0.77 × f + 18 (max. operation in high-speed operating mode)
I_{CC} Typ. = 0.08 × f + 6 (normal operation in high-speed operating mode)
I_{CC} Typ. = 0.5 × f + 2.6 (low-speed operating mode 1)
I_{CC} Max. = 0.36 × f + 18 (sleep mode)
Note 4. This does not include operations as BGO (background operations). Whether supply of the clock signal to peripheral modules continues or is stopped only depends on the state determined by the settings of the bits in module stop control registers A to D. The setting for the peripheral module clock stopped state is FCLK = BCLK = PCLKA = PCLKB = PCLKC = PCLKD = BCLK pin = 3.75 MHz (division by 64).
Note 5. This is the increase for programming or erasure of the code flash memory (limitations apply to the combinations of ranges in which writing proceed) or data flash memory during program execution in the code flash memory.
Note 6. The low power consumption function is disabled and DEEPCUT[1:0] = 01b.
Note 7. The low power consumption function is enabled and DEEPCUT[1:0] = 11b.

Table 64.5 DC Characteristics (4)

Conditions: VCC = AVCC0 = AVCC1 = VREFH0 = VCC_USB = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Analog power supply current*1	During 12-bit A/D conversion (unit 0)		AI_{CC}	—	0.7	1.0	mA	IAVCC0_AD
	During 12-bit A/D conversion (unit 0) with the channel-dedicated sample-and-hold circuits for 3 channels operating			—	1.7	2.5	mA	IAVCC0_AD+SH
	During 12-bit A/D conversion (unit 1)			—	0.6	1.0	mA	IAVCC1_AD
	During 12-bit A/D conversion (unit 1) with the temperature sensor operating			—	0.7	1.1	mA	IAVCC1_AD+TEMP
	During D/A conversion (per unit)	Without AMP output		—	0.24	0.4	mA	IAVCC1_DA
		With AMP output		—	0.4	0.7	mA	
	Waiting for A/D, D/A, or temperature sensor conversion (all units)			—	0.9	1.4	mA	IAVCC0 + IAVCC1
	A/D, D/A converter, temperature sensor in standby mode (all units)			—	1.3	3.0	µA	IAVCC0 + IAVCC1
Reference power supply current	During 12-bit A/D conversion (unit 0)		AI_{REFH}	—	70	120	µA	IVREFH0
	Waiting for 12-bit A/D conversion (unit 0)			—	0.07	0.4	µA	IVREFH0
	12-bit A/D converter in standby mode (unit 0)			—	0.07	0.2	µA	IVREFH0
USB operating current	Low speed	USB0	$I_{CCUSBLS}$	—	3.5	6.5	mA	VCC_USB
		USBA		—	8.5	12.0	mA	VCC_USBA = AVCC_USBA (PHYSET.HSEB = 0)
		USBA		—	2.8	3.6	mA	VCC_USBA = AVCC_USBA (PHYSET.HSEB = 1)
	Full speed	USB0	$I_{CCUSBFS}$	—	4.0	10.0	mA	VCC_USB
		USBA		—	12.0	20.0	mA	VCC_USBA = AVCC_USBA (PHYSET.HSEB = 0)
		USBA		—	6.5	13.0	mA	VCC_USBA = AVCC_USBA (PHYSET.HSEB = 1)
	Standby mode (direct power down)	USBA	$I_{CCUSBSBY}$	—	0.1	3.0	µA	VCC_USBA = AVCC_USBA
RAM standby voltage			V_{RAM}	2.7	—	—	V	
VCC rising gradient			SrVCC	8.4	—	20000	µs/V	
VCC falling gradient*2			SfVCC	8.4	—	—	µs/V	

Note 1. The reference power supply current is included in the power supply current value for 12-bit A/D conversion (unit 1) and D/A conversion.

Note 2. This applies when V_{BATT} is used.

Table 64.6 Permissible Output Currents

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item			Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (average value per pin)	All output pins*1	Normal drive	I_{OL}	—	—	2.0	mA
	All output pins*2	High drive	I_{OL}	—	—	3.8	mA
Permissible output low current (max. value per pin)	All output pins*1	Normal drive	I_{OL}	—	—	4.0	mA
	All output pins*2	High drive	I_{OL}	—	—	7.6	mA
Permissible output low current (total)	Total of all output pins		ΣI_{OL}	—	—	80	mA
Permissible output high current (average value per pin)	All output pins*1	Normal drive	I_{OH}	—	—	–2.0	mA
	USB_DPUPE pin*2	High drive	I_{OH}	—	—	–3.8	mA
Permissible output high current (max. value per pin)	All output pins*1	Normal drive	I_{OH}	—	—	–4.0	mA
	All output pins*2	High drive	I_{OH}	—	—	–7.6	mA
Permissible output high current (total)	Total of all output pins		ΣI_{OH}	—	—	–80	mA

Caution: To protect the LSI's reliability, the output current values should not exceed the values in this table.

Note 1. This is the value when normal driving ability is set with a pin for which normal driving ability is selectable.

Note 2. This is the value when high driving ability is set with a pin for which normal driving ability is selectable or the value of the pin to which high driving ability is fixed.

64.3 AC Characteristics

Table 64.7 Operating Frequency (High-Speed Operating Mode)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item			Symbol	Min.	Typ.	Max.	Unit
Operating frequency	System clock (ICLK)		f	—	—	120	MHz
	Peripheral module clock (PCLKA)			—	—	120	
	Peripheral module clock (PCLKB)			—	—	60	
	Peripheral module clock (PCLKC)			—	—	60	
	Peripheral module clock (PCLKD)			—	—	60	
	Flash-IF clock (FCLK)			—*1	—	60	
	External bus clock (BCLK)	Packages with 177 to 144 pins only		—	—	120	
		Package with 100 pins only		—	—	60	
	BCLK pin output	Packages with 177 to 144 pins only		—	—	60	
		Package with 100 pins only		—	—	30	
	SDRAM clock (SDCLK)	Packages with 177 to 144 pins only		—	—	60	
	SDCLK pin output	Packages with 177 to 144 pins only		—	—	60	

Note 1. The FCLK must run at a frequency of at least 4 MHz when changing the flash memory contents.

Table 64.8 Operating Frequency (Low-Speed Operating Mode 1)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item			Symbol	Min.	Typ.	Max.	Unit
Operating frequency	System clock (ICLK)		f	—	—	1	MHz
	Peripheral module clock (PCLKA)			—	—	1	
	Peripheral module clock (PCLKB)			—	—	1	
	Peripheral module clock (PCLKC)*1			—	—	1	
	Peripheral module clock (PCLKD)*1			—	—	1	
	Flash-IF clock (FCLK)			—	—	1	
	External bus clock (BCLK)	Packages with 177 to 144 pins only		—	—	1	
		Package with 100 pins only		—	—	1	
	BCLK pin output	Packages with 177 to 144 pins only		—	—	1	
		Package with 100 pins only		—	—	1	
	SDRAM clock (SDCLK)	Packages with 177 to 144 pins only		—	—	1	
	SDCLK pin output	Packages with 177 to 144 pins only		—	—	1	

Note 1. When the 12-bit A/D converter is used, the frequency must be set to at least 1 MHz.

Table 64.9 Operating Frequency (Low-Speed Operating Mode 2)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item			Symbol	Min.	Typ.	Max.	Unit
Operating frequency	System clock (ICLK)		f	32	—	264	kHz
	Peripheral module clock (PCLKA)			—	—	264	
	Peripheral module clock (PCLKB)			—	—	264	
	Peripheral module clock (PCLKC)*1			—	—	264	
	Peripheral module clock (PCLKD)*1			—	—	264	
	Flash-IF clock (FCLK)			32	—	264	
	External bus clock (BCLK)	Packages with 177 to 144 pins only		—	—	264	
		Package with 100 pins only		—	—	264	
	BCLK pin output	Packages with 177 to 144 pins only		—	—	264	
		Package with 100 pins only		—	—	264	
	SDRAM clock (SDCLK)	Packages with 177 to 144 pins only		—	—	264	
	SDCLK pin output	Packages with 177 to 144 pins only		—	—	264	

Note 1. The 12-bit A/D converter cannot be used.

64.3.1 Reset Timing

Table 64.10 Reset Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
RES# pulse width	Power-on	t_{RESWP}	1	—	—	ms	Figure 64.1
	Deep software standby mode	t_{RESWD}	0.6	—	—	ms	Figure 64.2
	Software standby mode, low-speed operating mode 2	t_{RESWS}	0.3	—	—	ms	
	Programming or erasure of the code flash memory, or programming, erasure or blank checking of the data flash memory	t_{RESWF}	200	—	—	µs	
	Other than above	t_{RESW}	200	—	—	µs	
Waiting time after release from the RES# pin reset		t_{RESWT}	62	—	63	t_{Lcyc}	Figure 64.1
Internal reset time (independent watchdog timer reset, watchdog timer reset, software reset)		t_{RESW2}	108	—	116	t_{Lcyc}	

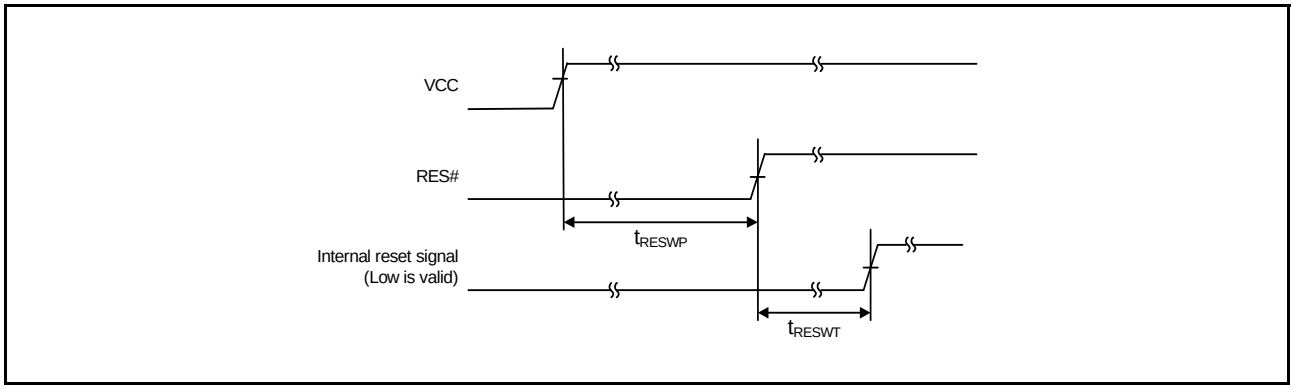


Figure 64.1 Reset Input Timing at Power-On

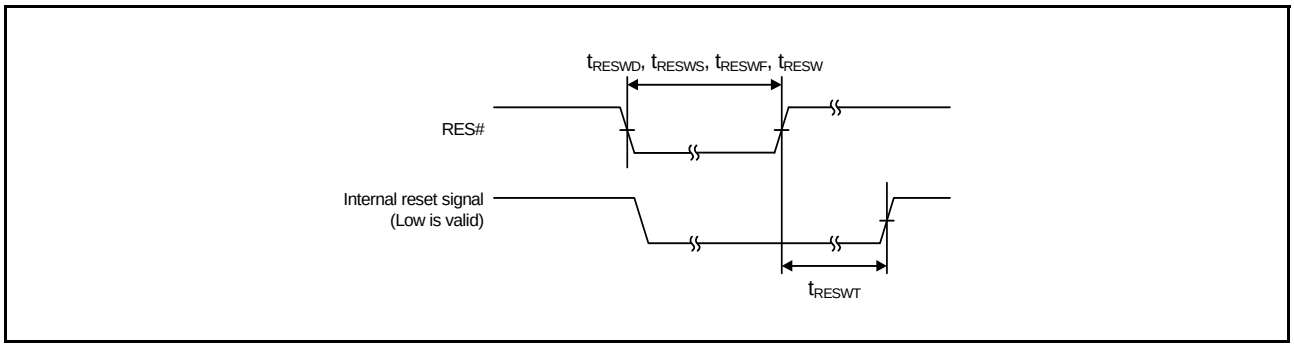


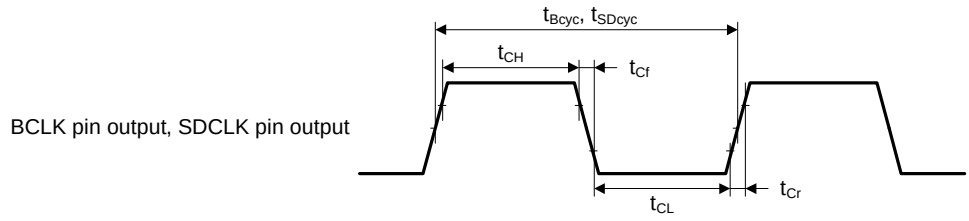
Figure 64.2 Reset Input Timing

64.3.2 Clock Timing

Table 64.11 BCLK Pin Output, SDCLK Pin Output Clock Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
BCLK pin output cycle time	Packages with 177 to 144 pins	t_{Bcyc}	16.6	—	—	ns	Figure 64.3
	Packages with 100 pins or less		33.2	—	—	ns	
BCLK pin output high pulse width		t_{CH}	3.3	—	—	ns	
BCLK pin output low pulse width		t_{CL}	3.3	—	—	ns	
BCLK pin output rising time		t_{Cr}	—	—	5	ns	
BCLK pin output falling time		t_{Cf}	—	—	5	ns	
SDCLK pin output cycle time	Packages with 177 to 144 pins	t_{Bcyc}	16.6	—	—	ns	
SDCLK pin output high pulse width		t_{CH}	3.3	—	—	ns	
SDCLK pin output low pulse width		t_{CL}	3.3	—	—	ns	
SDCLK pin output rising time		t_{Cr}	—	—	5	ns	
SDCLK pin output falling time		t_{Cf}	—	—	5	ns	



Test conditions: VOH = VCC × 0.7, VOL = VCC × 0.3, C = 30 pF

Figure 64.3 BCLK Pin and SDCLK Pin Output Timing

Table 64.12 EXTAL Clock Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
EXTAL external clock input cycle time	t_{EXcyc}	41.66	—	—	ns	Figure 64.4
EXTAL external clock input high pulse width	t_{EXH}	15.83	—	—	ns	
EXTAL external clock input low pulse width	t_{EXL}	15.83	—	—	ns	
EXTAL external clock rising time	t_{EXr}	—	—	5	ns	
EXTAL external clock falling time	t_{EXf}	—	—	5	ns	

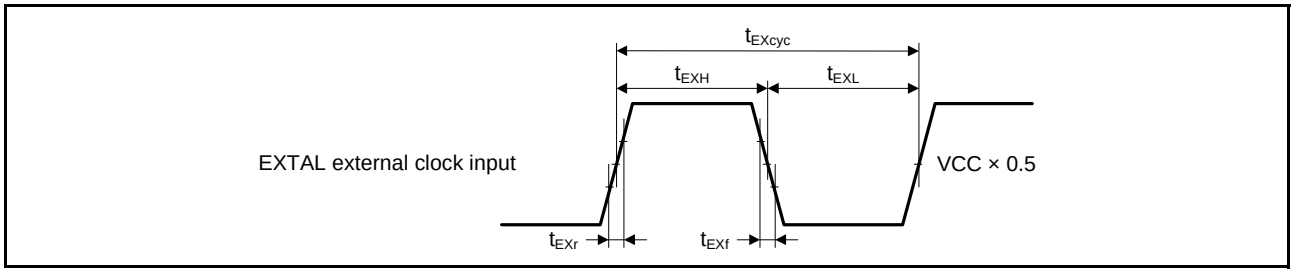


Figure 64.4 EXTAL External Clock Input Timing

Table 64.13 Main Clock Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Main clock oscillation frequency	f_{MAIN}	8	—	24	MHz	
Main clock oscillator stabilization time (crystal)	$t_{MAINOSC}$	—	—	—*1	ms	Figure 64.5
Main clock oscillation stabilization wait time (crystal)	$t_{MAINOSCWT}$	—	—	—*2	ms	

Note 1. When using a main clock, ask the manufacturer of the oscillator to evaluate its oscillation. Refer to the results of evaluation provided by the manufacturer for the oscillation stabilization time.

Note 2. The number of cycles selected by the value of the MOSCWTCR.MSTS[7:0] bits determines the main clock oscillation stabilization wait time in accord with the formula below.
$t_{MAINOSCWT}$ = [(MSTS[7:0] bits × 32) + 10] / f_{LOCO}

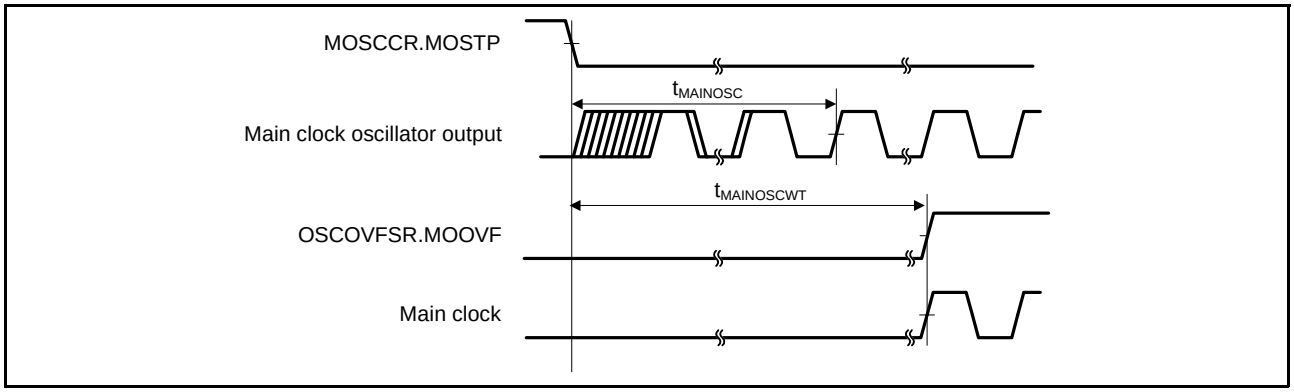


Figure 64.5 Main Clock Oscillation Start Timing

Table 64.14 LOCO and IWDT-Dedicated Low-Speed Clock Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
LOCO clock cycle time	t_{Lcyc}	4.63	4.16	3.78	µs	
LOCO clock oscillation frequency	f_{LOCO}	216	240	264	kHz	
LOCO clock oscillation stabilization wait time	t_{LOCOWT}	—	—	44	µs	Figure 64.6
IWDT-dedicated low-speed clock cycle time	t_{ILcyc}	9.26	8.33	7.57	µs	
IWDT-dedicated low-speed clock oscillation frequency	f_{ILOCO}	108	120	132	kHz	
IWDT-dedicated low-speed clock oscillation stabilization wait time	$t_{ILOCOWT}$	—	142	190	µs	Figure 64.7

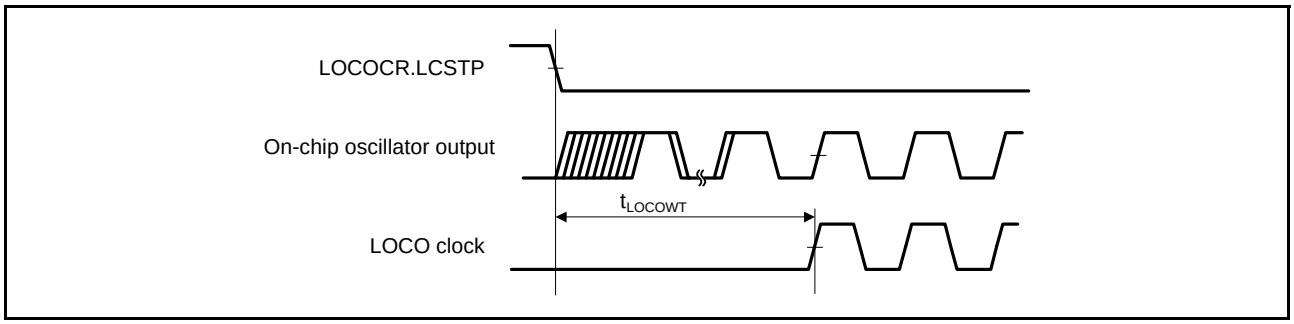


Figure 64.6 LOCO Clock Oscillation Start Timing

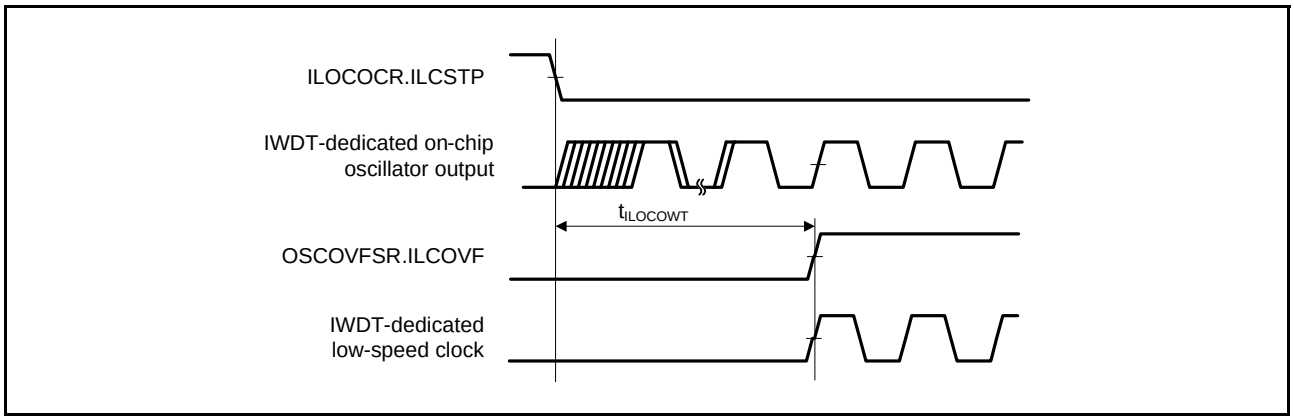


Figure 64.7 IWDT-dedicated Low-Speed Clock Oscillation Start Timing

Table 64.15 HOCO Clock Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
HOCO clock oscillation frequency	f_{HOCO}	15.61	16	16.39	MHz	-20°C ≤ Ta ≤ 85°C
		17.56	18	18.44	MHz	
		19.52	20	20.48	MHz	
		15.52	16	16.48	MHz	-40°C ≤ T_a < -20°C
		17.46	18	18.54	MHz	
		19.40	20	20.60	MHz	
HOCO clock oscillation stabilization wait time	t_{HOCOWT}	—	105	149	µs	Figure 64.8
HOCO clock power supply stabilization time	t_{HOCOP}	—	—	150	µs	Figure 64.9

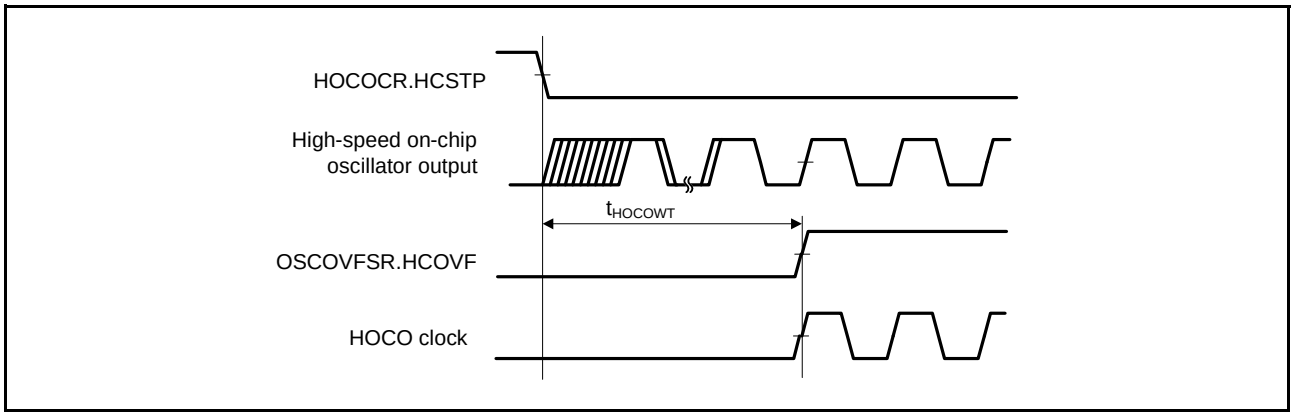


Figure 64.8 HOCO Clock Oscillation Start Timing (Oscillation is Started by Setting the HOCOCR.HCSTP Bit)

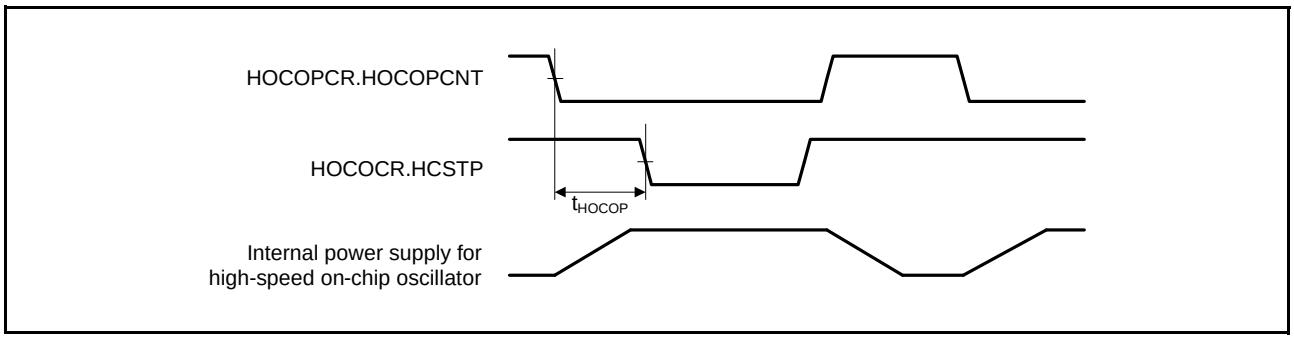


Figure 64.9 High-Speed On-Chip Oscillator Power Supply Control Timing

Table 64.16 PLL Clock Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
PLL clock oscillation frequency	f_{PLL}	120	—	240	MHz	
PLL clock oscillation stabilization wait time	t_{PLLWT}	—	259	320	µs	Figure 64.10

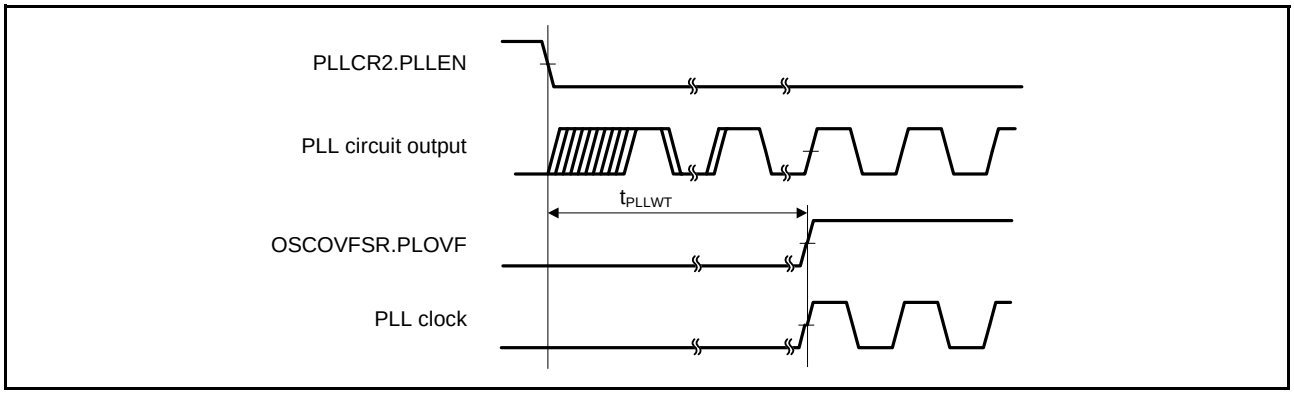


Figure 64.10 PLL Clock Oscillation Start Timing

Table 64.17 Sub-Clock Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
V_{BATT} = 2.0 to 3.6 V, $T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Sub-clock oscillation frequency	f_{SUB}	—	32.768	—	kHz	
Sub-clock oscillation stabilization time	t_{SUBOSC}	—	—	*1	s	Figure 64.11
Sub-clock oscillation stabilization wait time	$t_{SUBOSCWT}$	—	—	*2	s	

Note 1. When using a sub-clock, ask the manufacturer of the oscillator to evaluate its oscillation. Refer to the results of evaluation provided by the manufacturer for the oscillation stabilization time.

Note 2. The number of cycles selected by the value of the SOSCWTCR.SSTS[7:0] bits determines the sub-clock oscillation stabilization wait time in accord with the formula below.
$t_{SUBOSCWT}$ = [(SSTS[7:0] bits × 16384) + 10] / f_{LOCO}

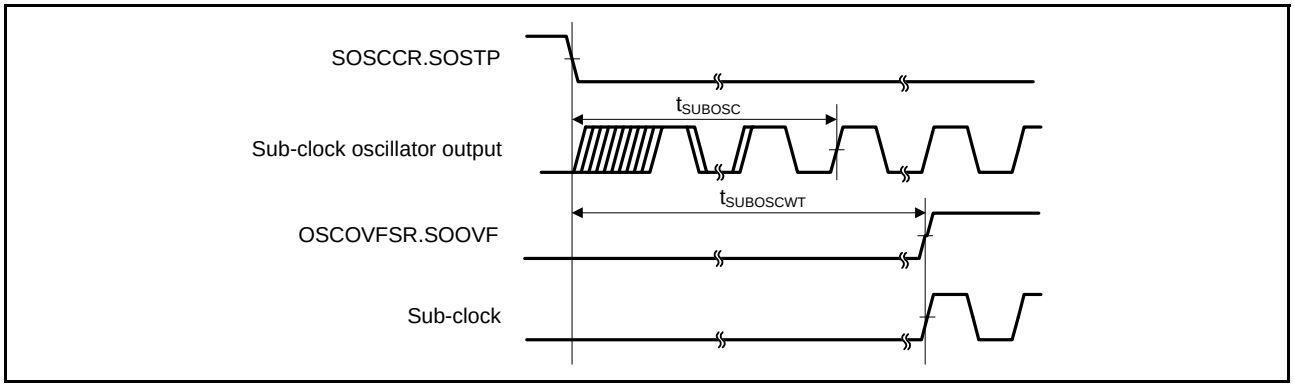


Figure 64.11 Sub-Clock Oscillation Start Timing

64.3.3 Timing of Recovery from Low Power Consumption Modes

Table 64.18 Timing of Recovery from Low Power Consumption Modes (1)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item			Symbol	Min.	Typ.	Max.		Unit	Test Conditions
						$t_{SBYOSCWT}$*2	t_{SBYSEQ}*3		
Recovery time after cancellation of software standby mode*1	Crystal resonator connected to main clock oscillator	Main clock oscillator operating	t_{SBYMC}	—	—	{(MSTS[7:0] bits × 32) + 76 } / 0.216	100 µs + 7/f_{ICLK} + 2n/f_{MAIN}	µs	Figure 64.12
		Main clock oscillator and PLL circuit operating	t_{SBYPC}			{(MSTS[7:0] bits × 32) + 138 } / 0.216	100 µs + 7/f_{ICLK} + 2n/f_{PLL}		
	External clock input to main clock oscillator	Main clock oscillator operating	t_{SBYEX}			352	100 µs + 7/f_{ICLK} + 2n/f_{EXMAIN}		
		Main clock oscillator and PLL circuit operating	t_{SBYPE}			639	100 µs + 7/f_{ICLK} + 2n/f_{PLL}		
	Sub-clock oscillator operating		t_{SBYSC}			{(SSTS[7:0] bits × 16384) + 13 } / 0.216 + 10/f_{FCLK}	100 µs + 4/f_{ICLK} + 2n/f_{SUB}		
	High-speed on-chip oscillator operating	High-speed on-chip oscillator operating	t_{SBYHO}			454	100 µs + 7/f_{ICLK} + 2n/f_{HOCO}		
		High-speed on-chip oscillator operating and PLL circuit operating	t_{SBYPH}			741	100 µs + 7/f_{ICLK} + 2n/f_{PLL}		
	Low-speed on-chip oscillator operating*4		t_{SBYLO}			338	100 µs + 7/f_{ICLK} + 2n/f_{LOCO}		

Note 1. The time for return after release from software standby is determined by the value obtained by adding the oscillation stabilization waiting time ($t_{SBYOSCWTO}$) and the time required for operations by the software standby release sequencer (t_{SBYSEQ}).
Note 2. When several oscillators were running before the transition to software standby, the greatest value of the oscillation stabilization waiting time $t_{SBYOSCWT}$ is selected.
Note 3. For n, the greatest value is selected from among the internal clock division settings.
Note 4. This condition applies when f_{ICLK}:f_{FCLK} = 1:1, 2:1, or 4:1.

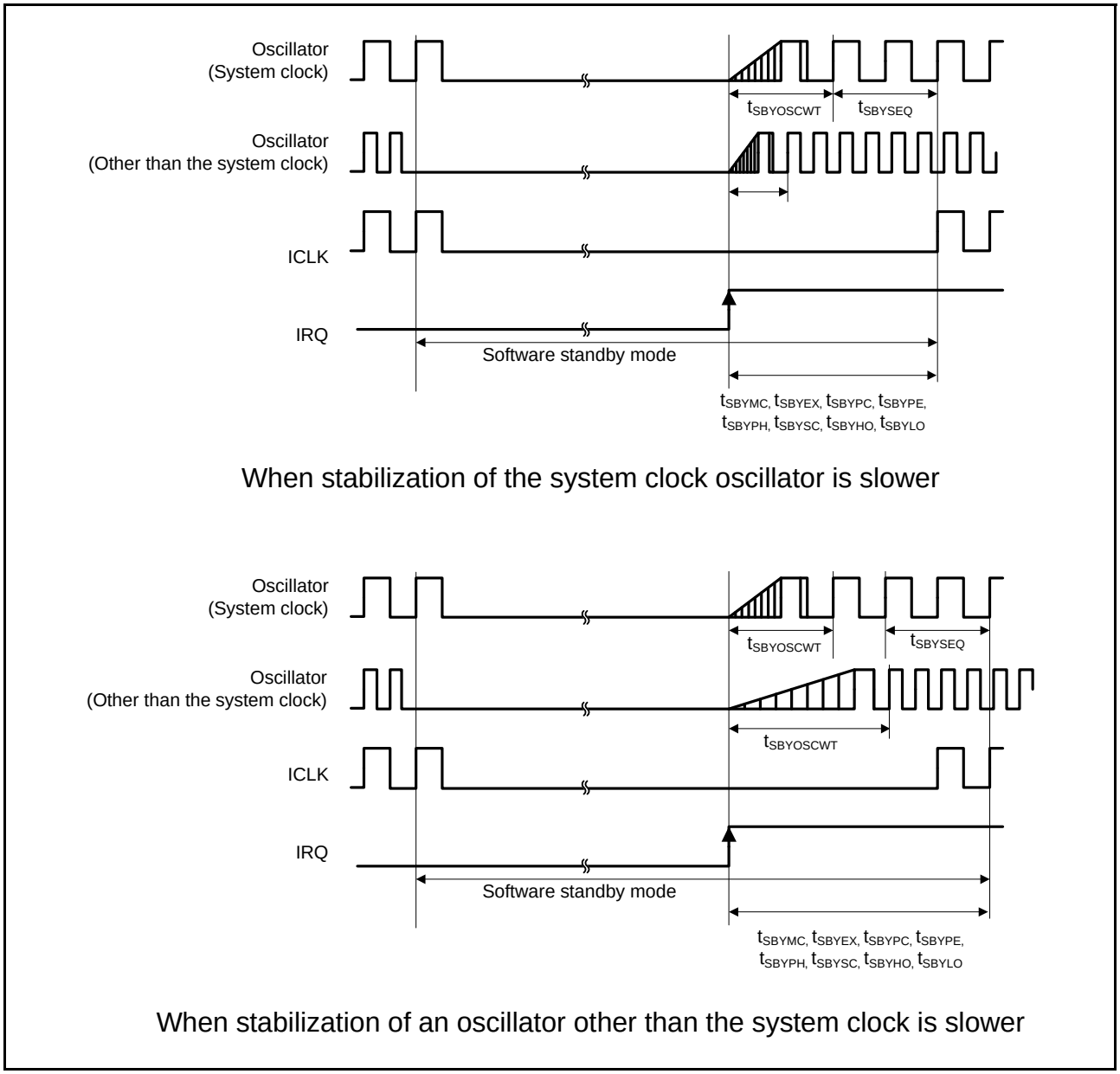


Figure 64.12 Software Standby Mode Cancellation Timing

Table 64.19 Timing of Recovery from Low Power Consumption Modes (2)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item	Symbol	min	typ	max	Unit	Test Conditions
Recovery time after cancellation of deep software standby mode	t_{DSBY}	—	—	0.9	ms	Figure 64.13
Wait time after cancellation of deep software standby mode	t_{DSBYWT}	31	—	32	t_{Lcyc}	

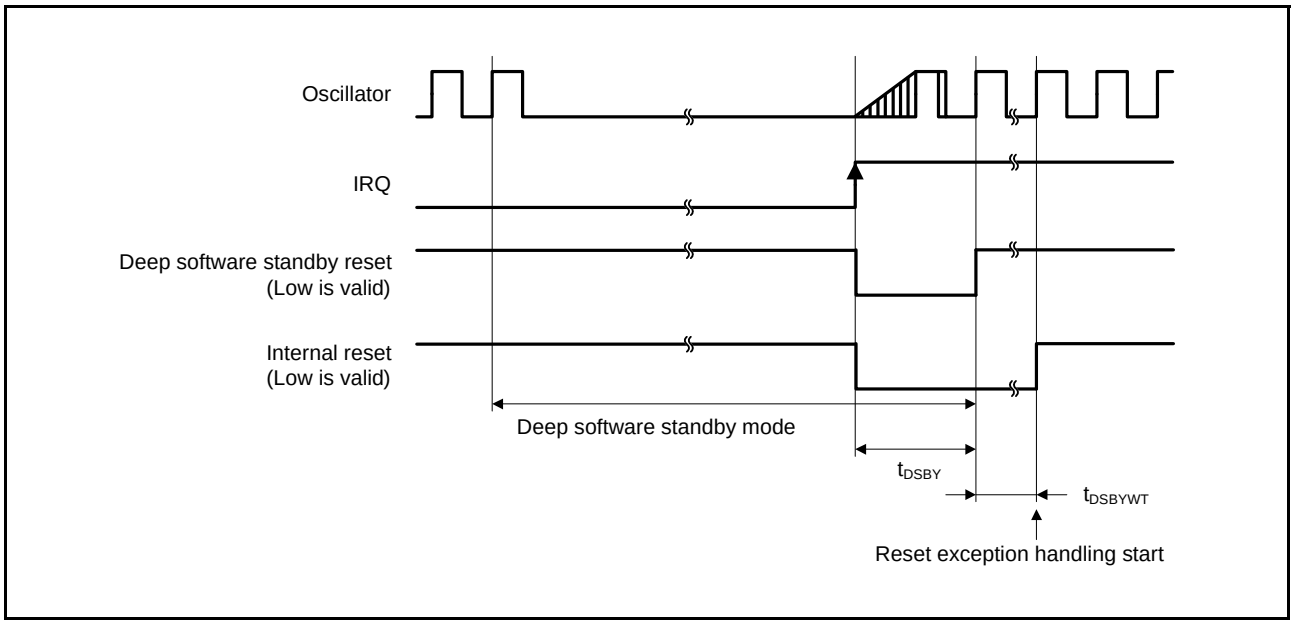


Figure 64.13 Deep Software Standby Mode Cancellation Timing

64.3.4 Control Signal Timing

Table 64.20 Control Signal Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PLCKB = 8 to 60 MHz, $T_a = T_{opr}$

Item	Symbol	Min.*1	Typ.	Max.	Unit	Test Conditions*1
NMI pulse width	t_{NMIW}	200	—	—	ns	$t_{PBcyc} \times 2 \leq 200$ ns, Figure 64.14
		$t_{PBcyc} \times 2$	—	—	ns	$t_{PBcyc} \times 2 > 200$ ns, Figure 64.14
IRQ pulse width	t_{IRQW}	200	—	—	ns	$t_{PBcyc} \times 2 \leq 200$ ns, Figure 64.15
		$t_{PBcyc} \times 2$	—	—	ns	$t_{PBcyc} \times 2 > 200$ ns, Figure 64.15

Note 1. t_{PBcyc}: PCLKB cycle

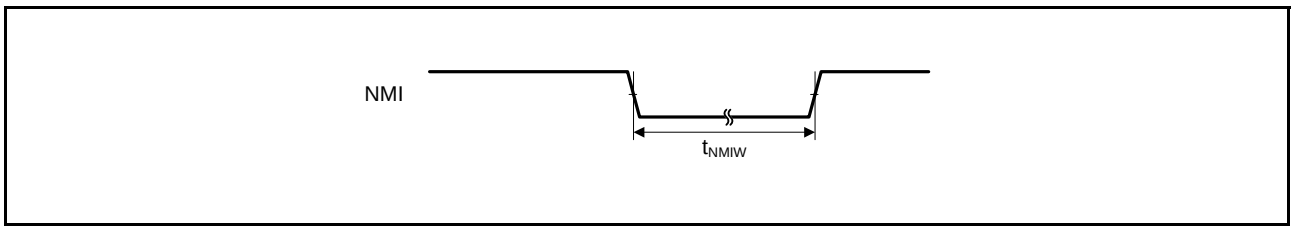


Figure 64.14 NMI Interrupt Input Timing

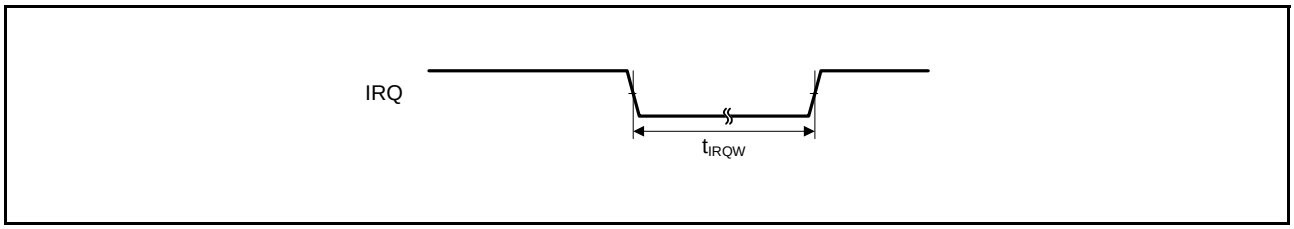


Figure 64.15 IRQ Interrupt Input Timing

64.3.5 Bus Timing

Table 64.21 Bus Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
ICLK = PCLKA = 8 to 120 MHz, PCLKB = BCLK = SDCLK = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item	Symbol	Min.	Max.	Unit	Test Conditions
Address delay time	t_{AD}	—	12.5	ns	Figure 64.16 to Figure 64.21
Byte control delay time	t_{BCD}	—	12.5	ns	
CS# delay time	t_{CSD}	—	12.5	ns	
ALE delay time	t_{ALED}	—	12.5	ns	
RD# delay time	t_{RSD}	—	12.5	ns	
Read data setup time	t_{RDS}	12.5	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
WR# delay time	t_{WRD}	—	12.5	ns	
Write data delay time	t_{WDD}	—	12.5	ns	
Write data hold time	t_{WDH}	0	—	ns	
WAIT# setup time	t_{WTS}	12.5	—	ns	Figure 64.22
WAIT# hold time	t_{WTH}	0	—	ns	
Address delay time 2 (SDRAM)	t_{AD2}	1	12.5	ns	Figure 64.23
CS# delay time 2 (SDRAM)	t_{CSD2}	1	12.5	ns	
DQM delay time (SDRAM)	t_{DQMD}	1	12.5	ns	
CKE delay time (SDRAM)	t_{CKED}	1	12.5	ns	
Read data setup time 2 (SDRAM)	t_{RDS2}	10	—	ns	
Read data hold time 2 (SDRAM)	t_{RDH2}	0	—	ns	
Write data delay time 2 (SDRAM)	t_{WDD2}	—	12.5	ns	
Write data hold time 2 (SDRAM)	t_{WDH2}	1	—	ns	
WE# delay time (SDRAM)	t_{WED}	1	12.5	ns	
RAS# delay time (SDRAM)	t_{RASD}	1	12.5	ns	
CAS# delay time (SDRAM)	t_{CASD}	1	12.5	ns	

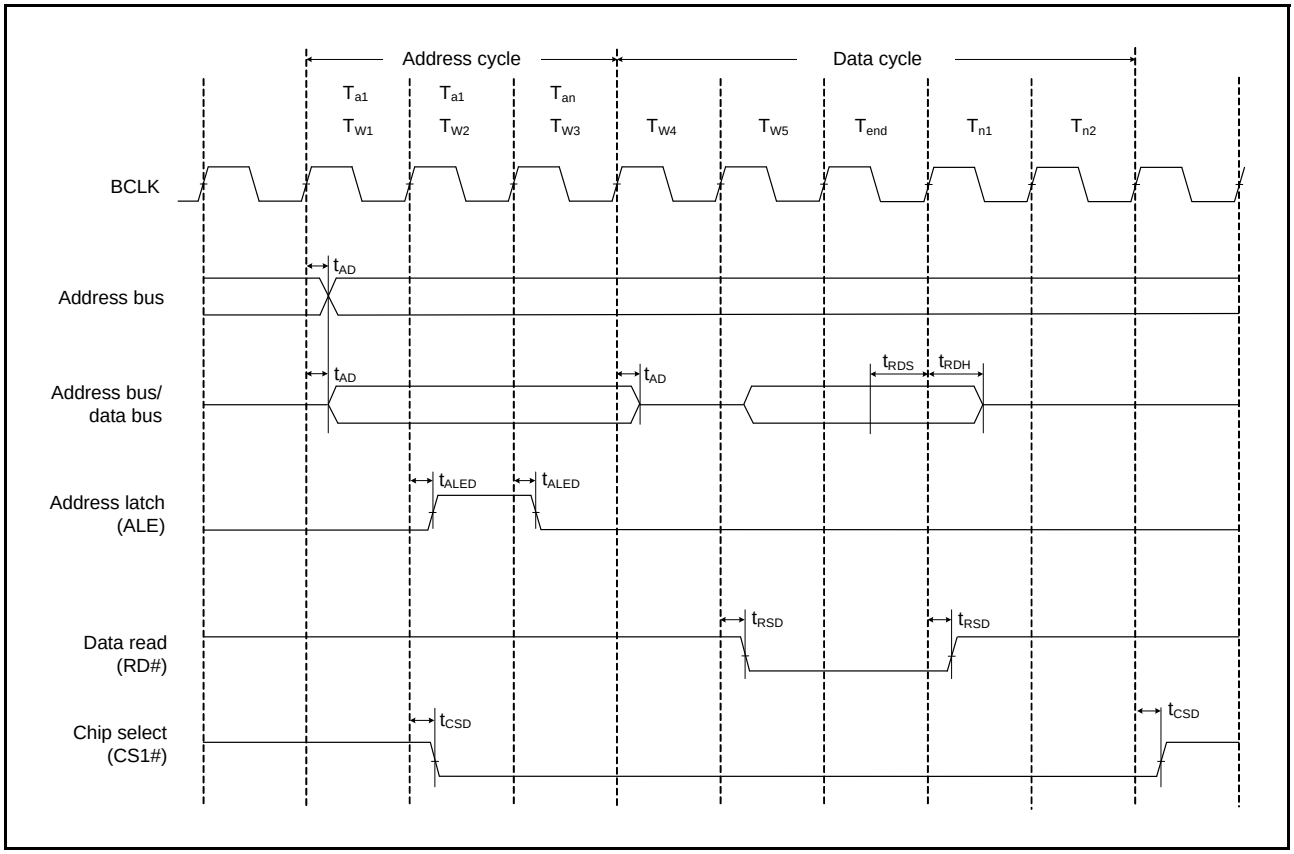


Figure 64.16 Address/Data Multiplexed Bus Read Access Timing

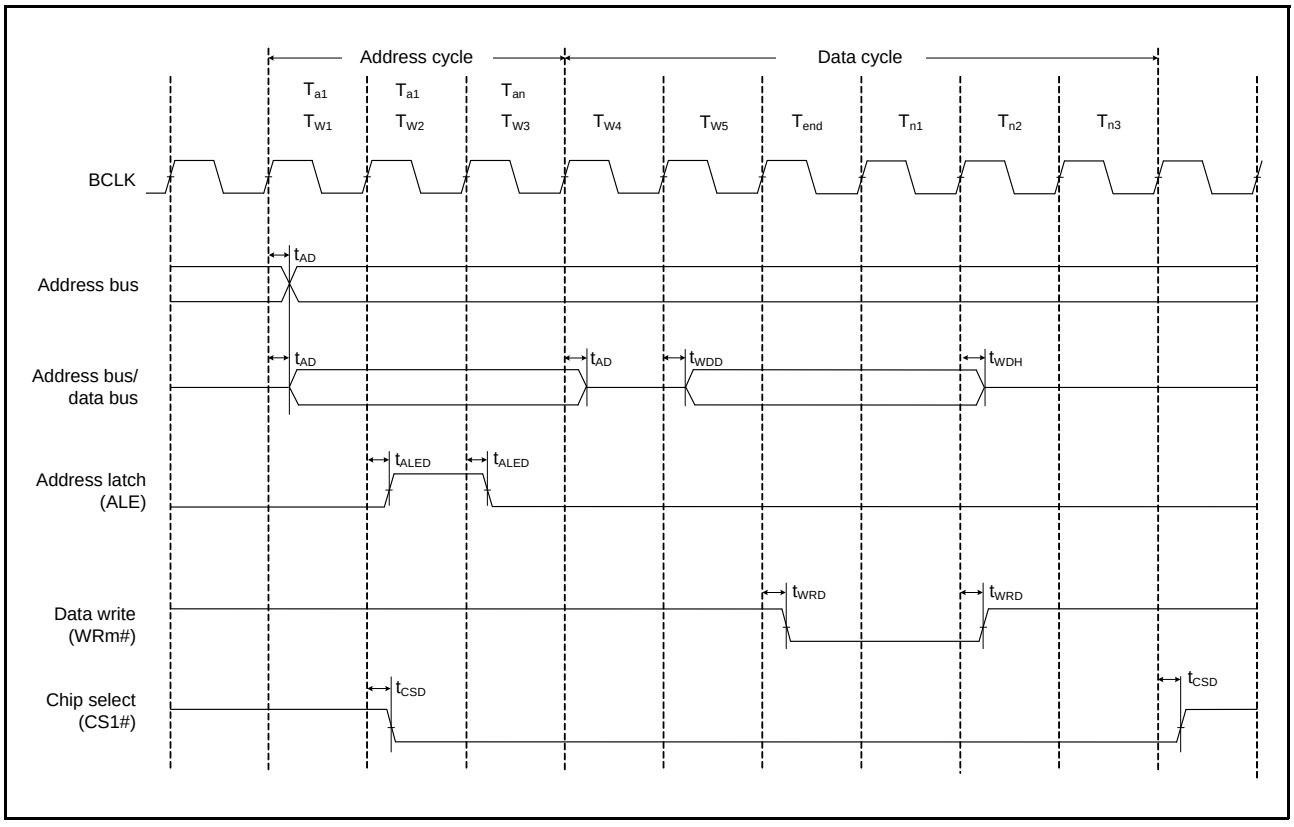


Figure 64.17 Address/Data Multiplexed Bus Write Access Timing

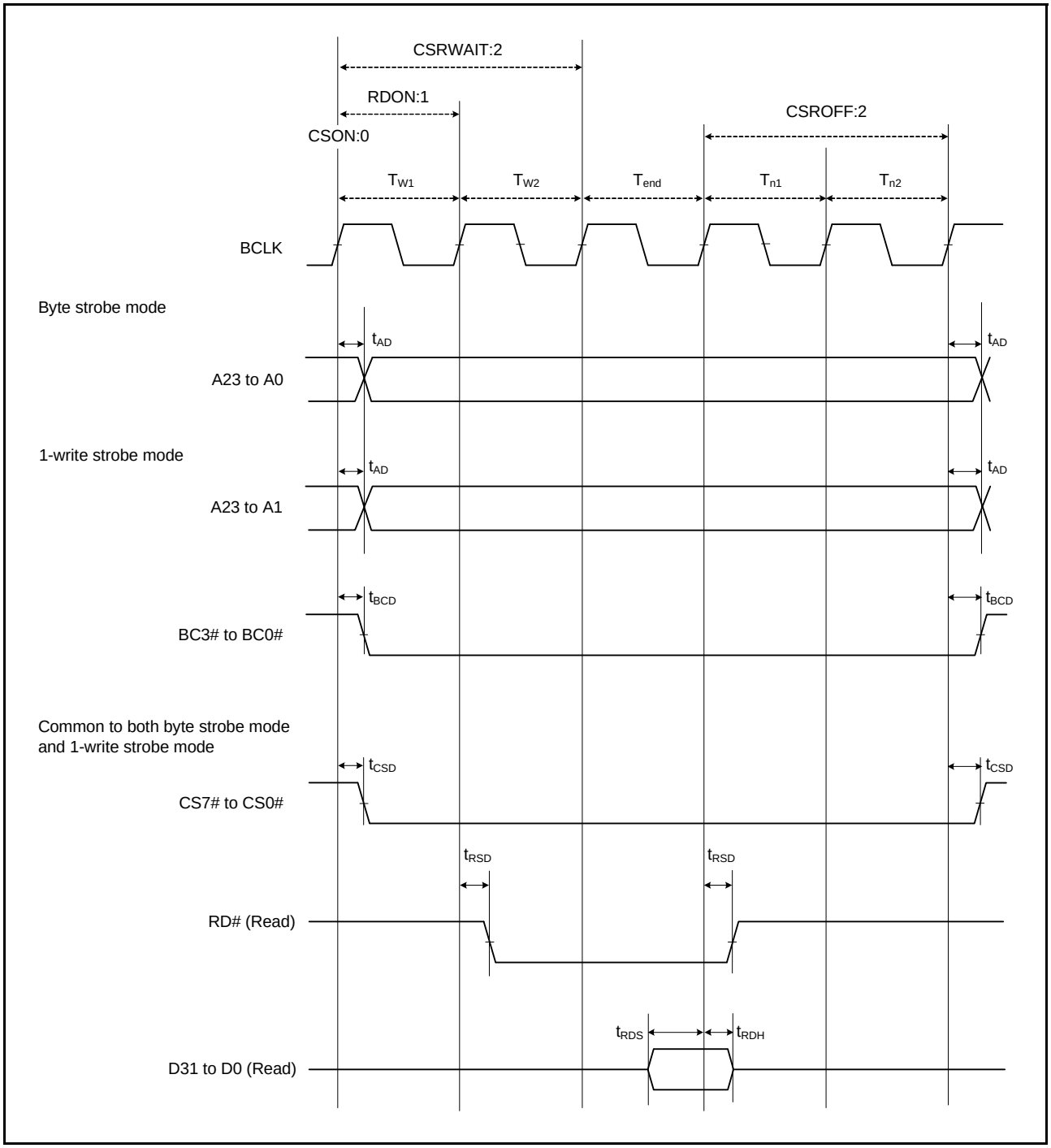
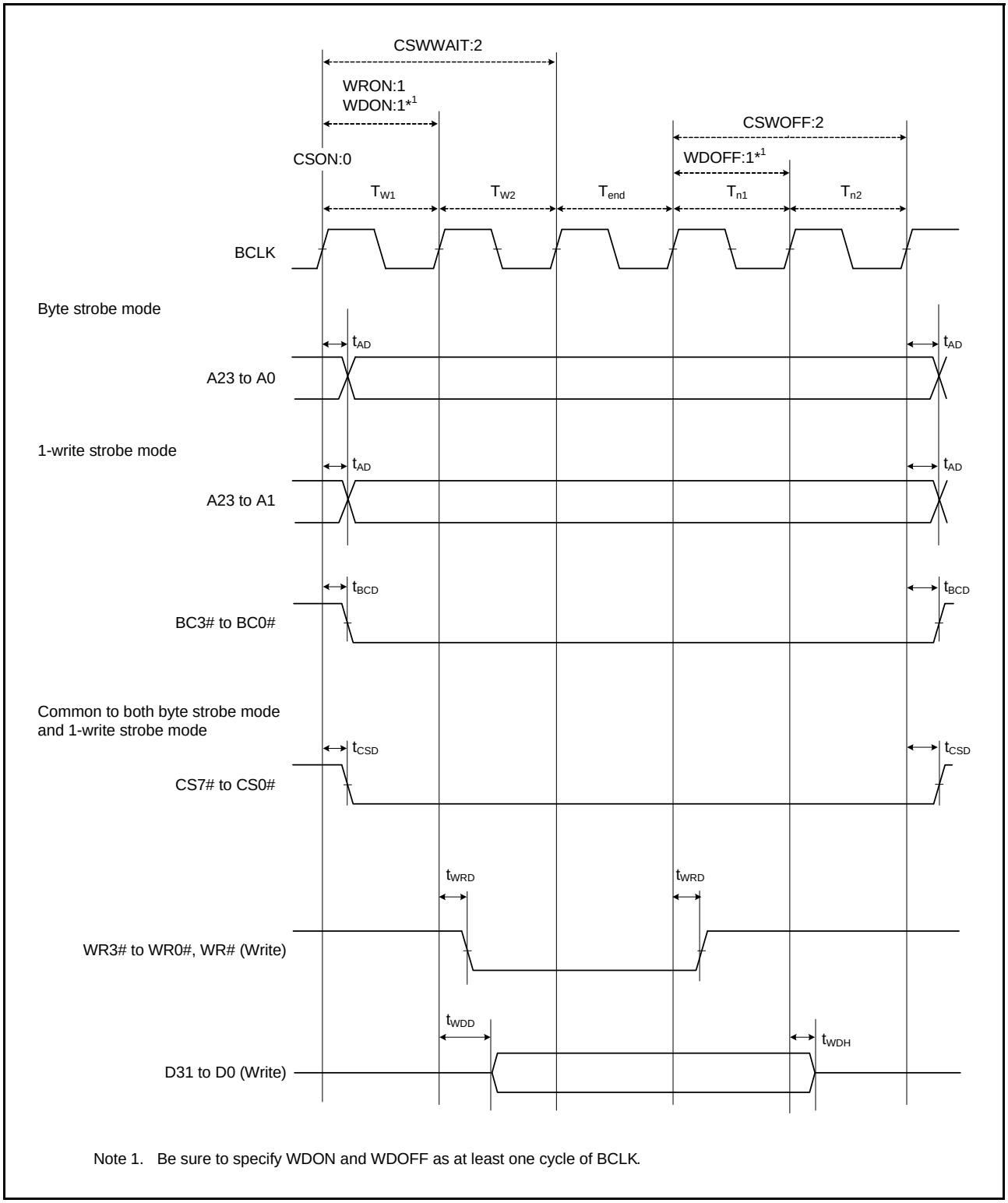


Figure 64.18 External Bus Timing/Normal Read Cycle (Bus Clock Synchronized)



Note 1. Be sure to specify WDON and WDOFF as at least one cycle of BCLK.

Figure 64.19 External Bus Timing/Normal Write Cycle (Bus Clock Synchronized)

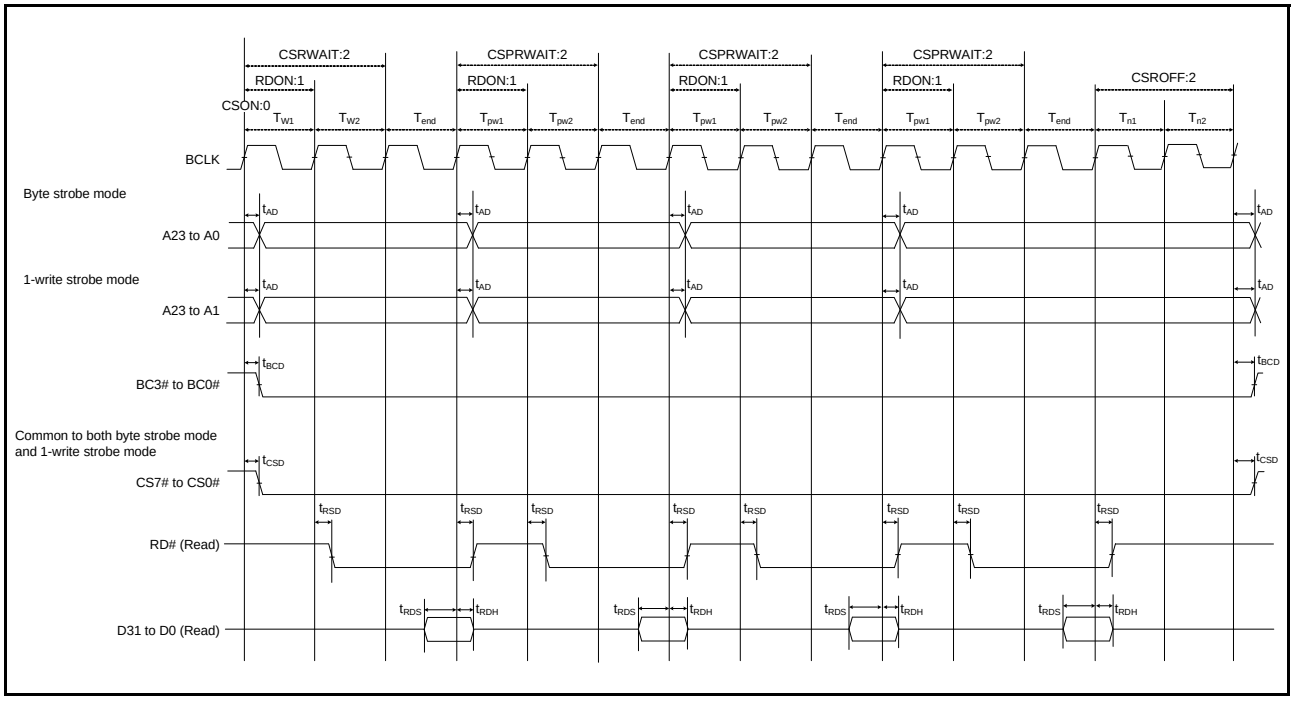
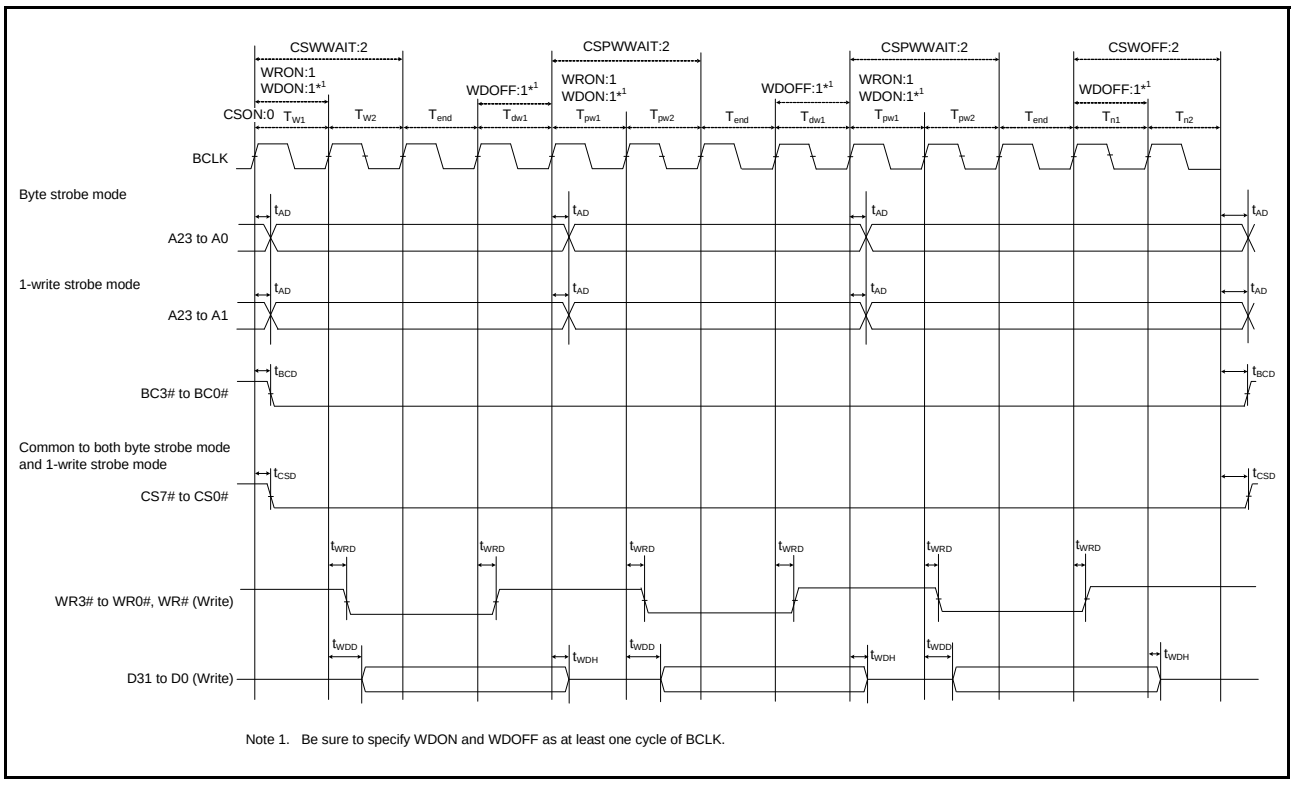


Figure 64.20 External Bus Timing/Page Read Cycle (Bus Clock Synchronized)



Note 1. Be sure to specify WDON and WDOFF as at least one cycle of BCLK.

Figure 64.21 External Bus Timing/Page Write Cycle (Bus Clock Synchronized)

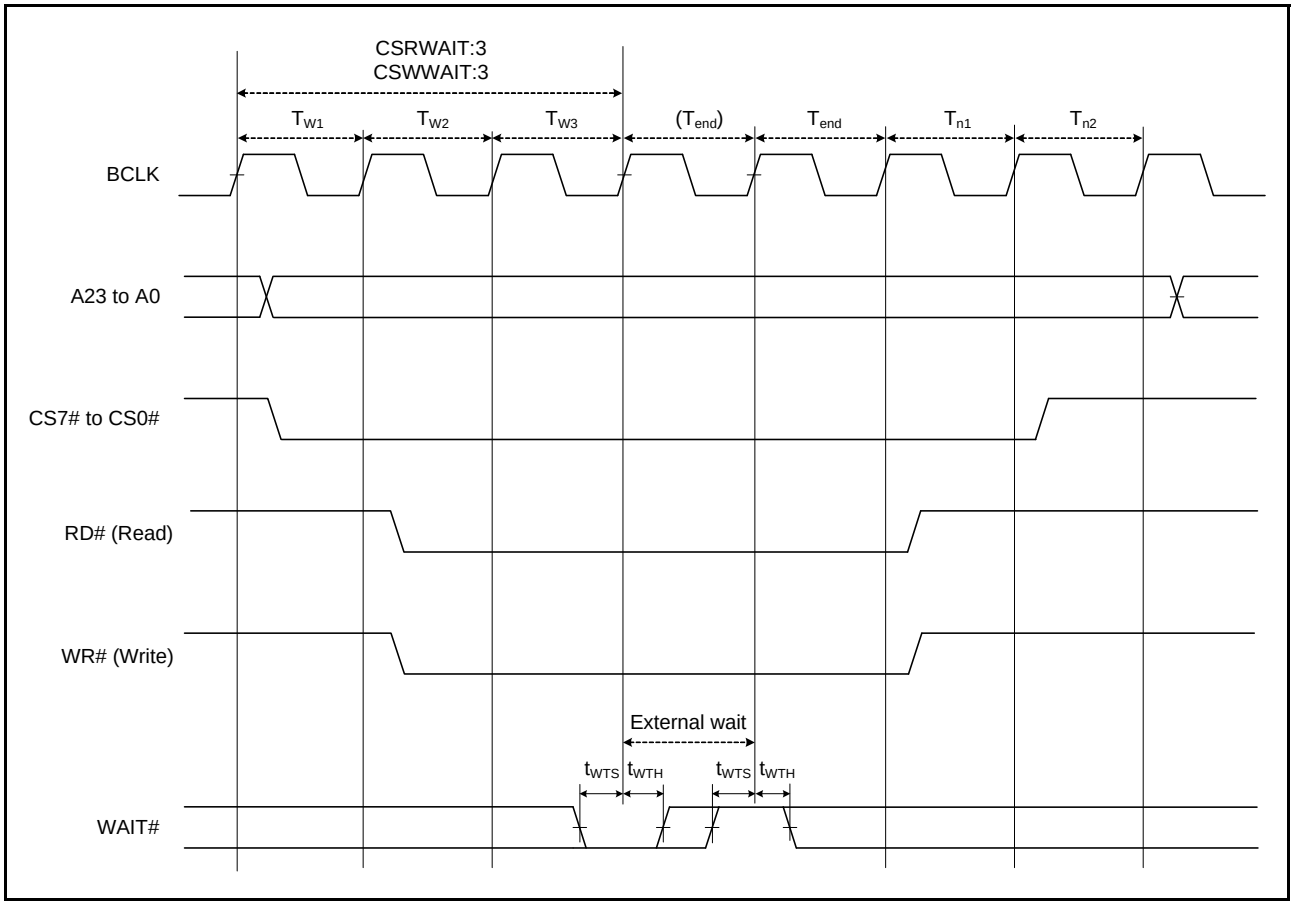
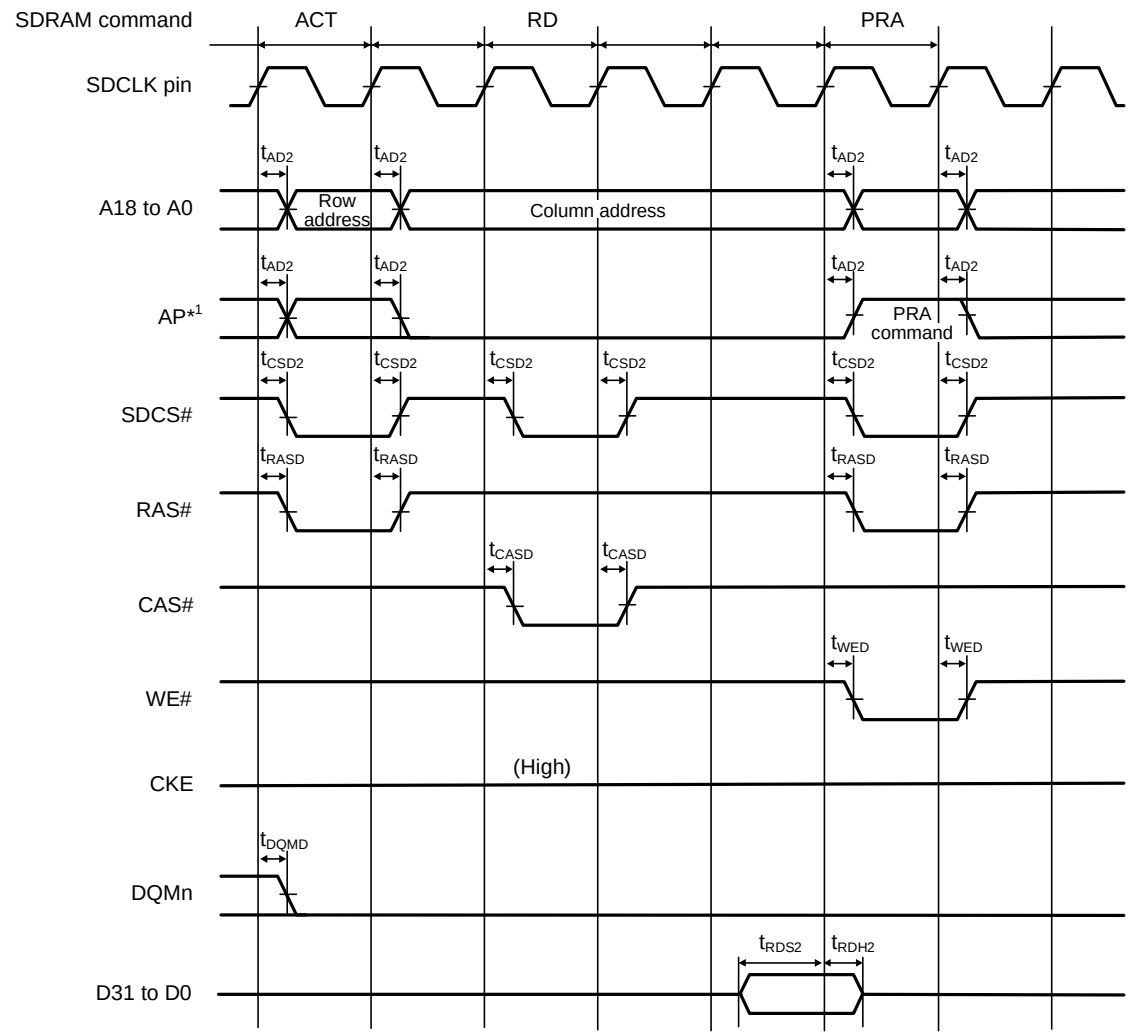
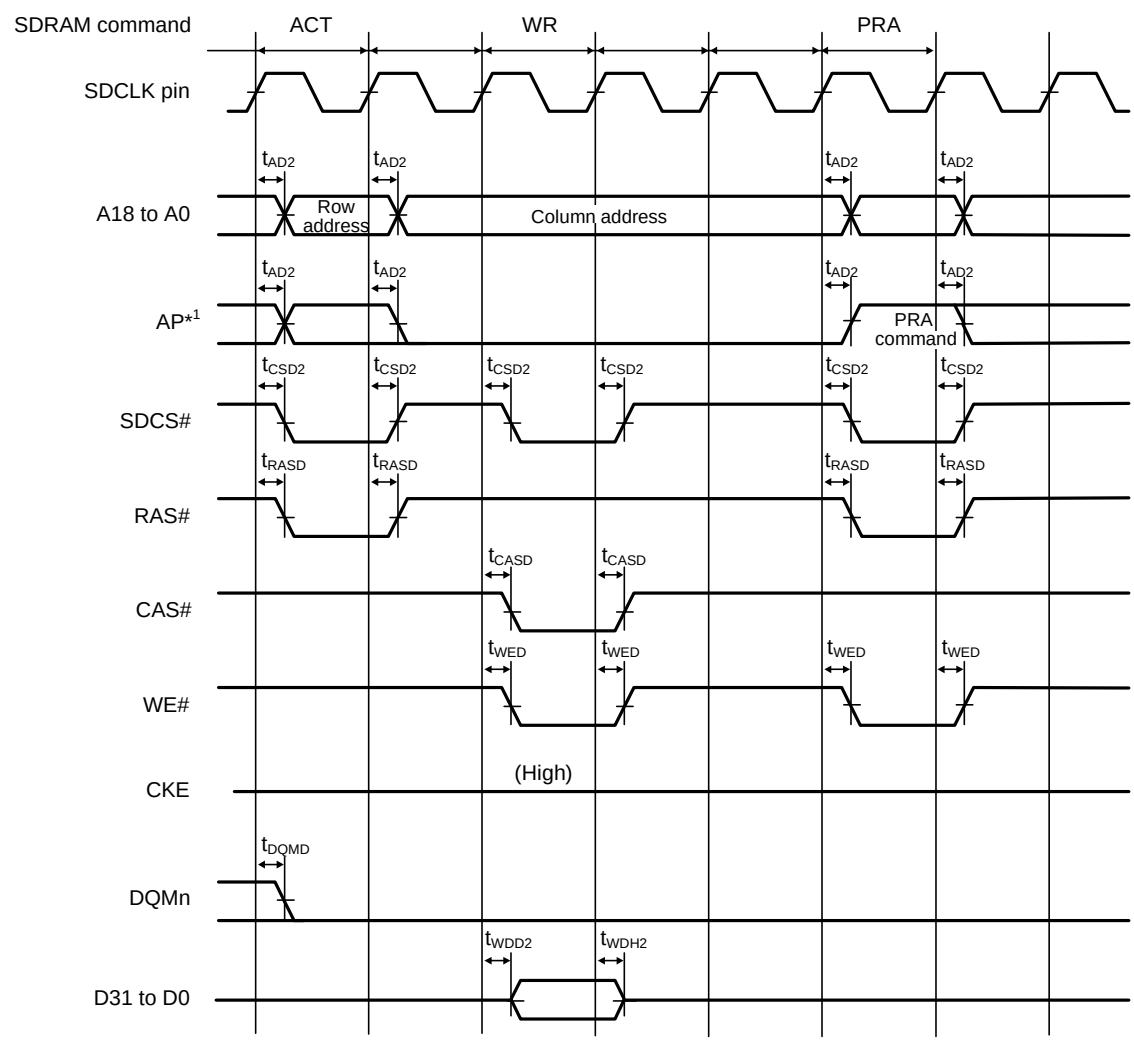


Figure 64.22 External Bus Timing/External Wait Control



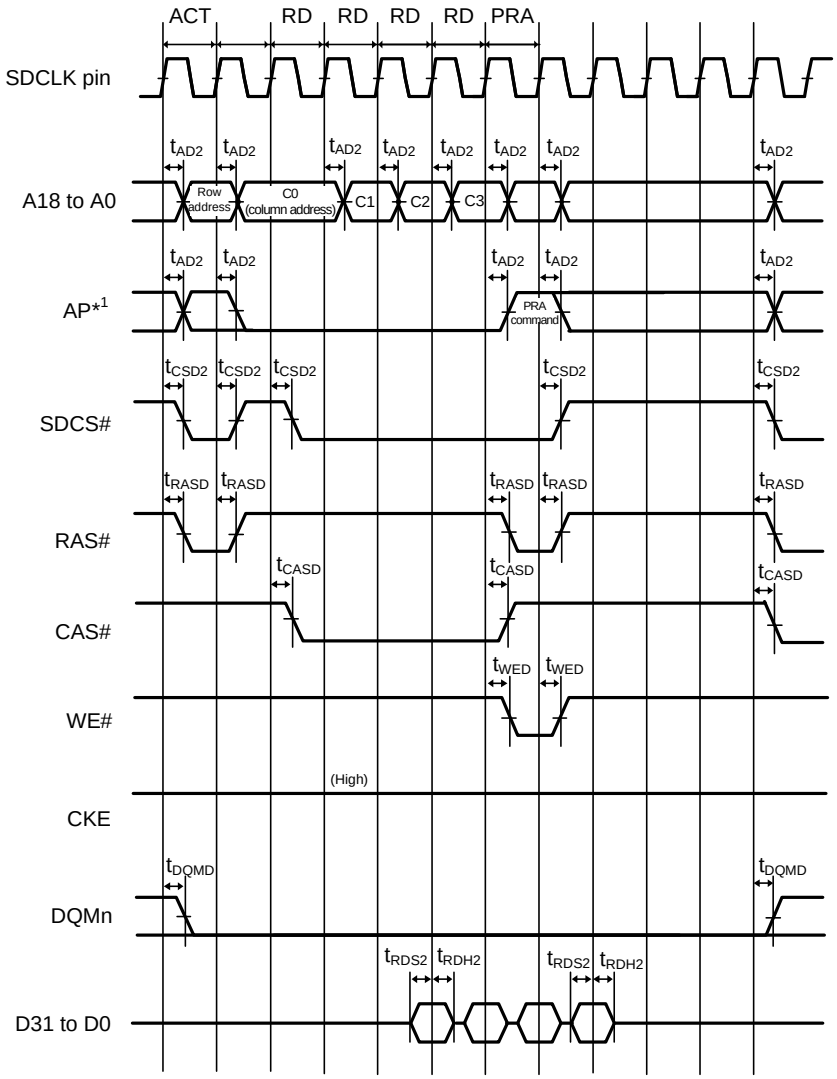
Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 64.23 SDRAM Space Single Read Bus Timing



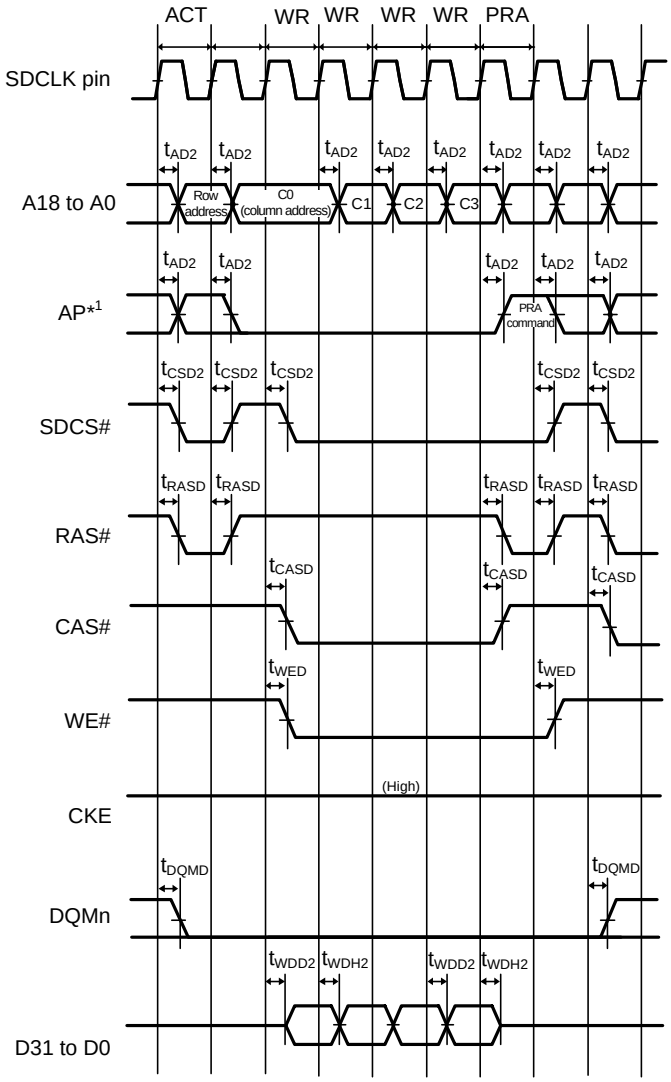
Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 64.24 SDRAM Space Single Write Bus Timing



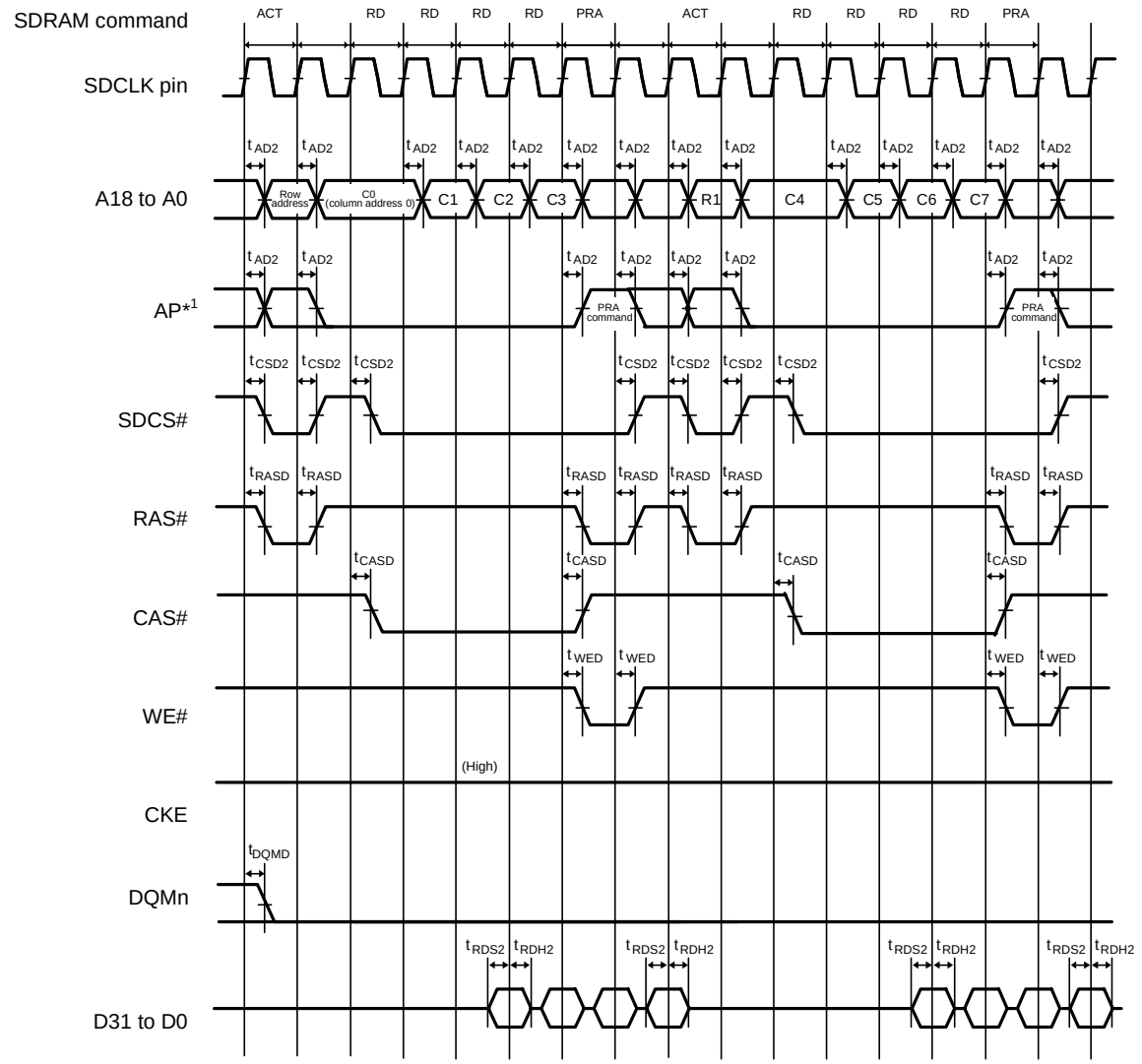
Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 64.25 SDRAM Space Multiple Read Bus Timing



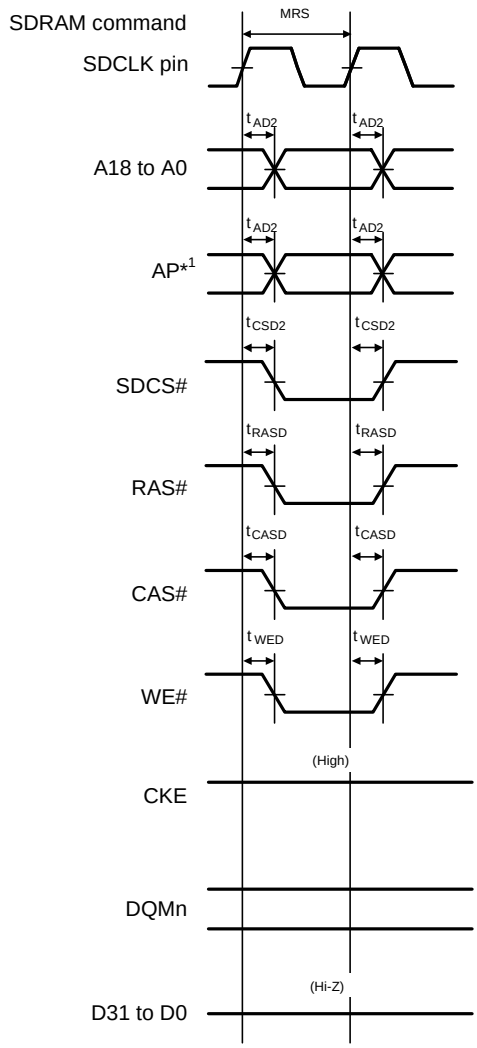
Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 64.26 SDRAM Space Multiple Write Bus Timing



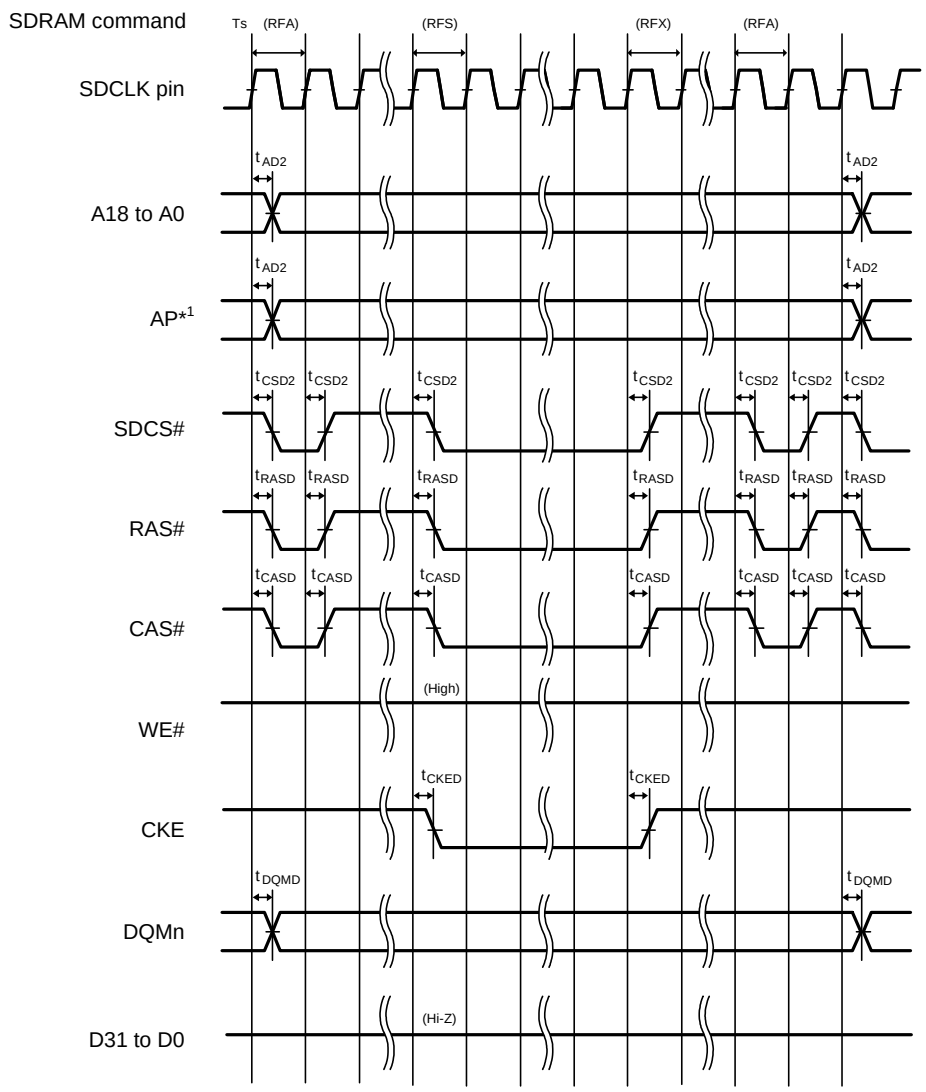
Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 64.27 SDRAM Space Multiple Read Line Stride Bus Timing



Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 64.28 SDRAM Space Mode Register Set Bus Timing



Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 64.29 SDRAM Space Self-Refresh Bus Timing

64.3.6 EXDMAC Timing

Table 64.22 EXDMAC Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
ICLK = PCLKA = 8 to 120 MHz, PCLKB = BCLK = SDCLK = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item		Symbol	Min.	Max.	Unit	Test Conditions
EXDMAC	EDREQ setup time	t_{EDRQS}	13	—	ns	Figure 64.30
	EDREQ hold time	t_{EDRQH}	2	—	ns	
	EDACK delay time	t_{EDACD}	—	13	ns	Figure 64.31, Figure 64.32

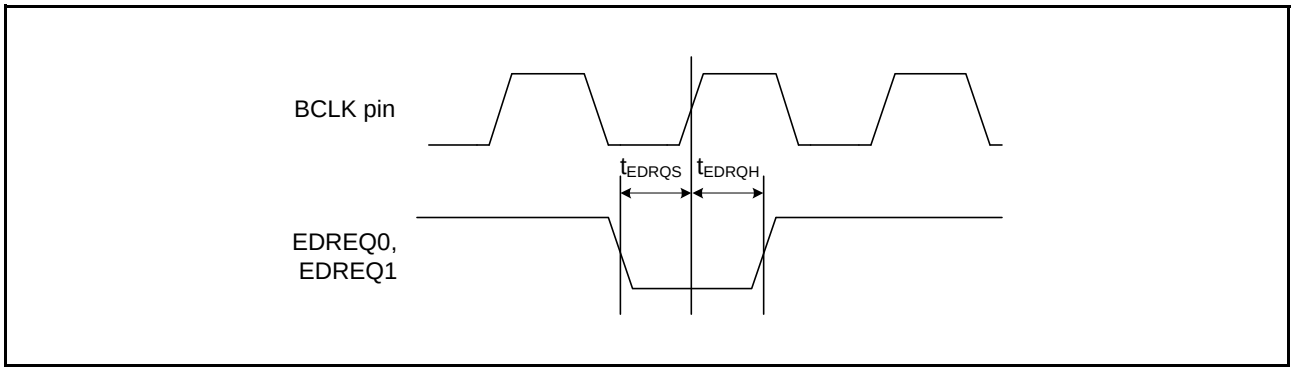


Figure 64.30 EDREQ0 and EDREQ1 Input Timing

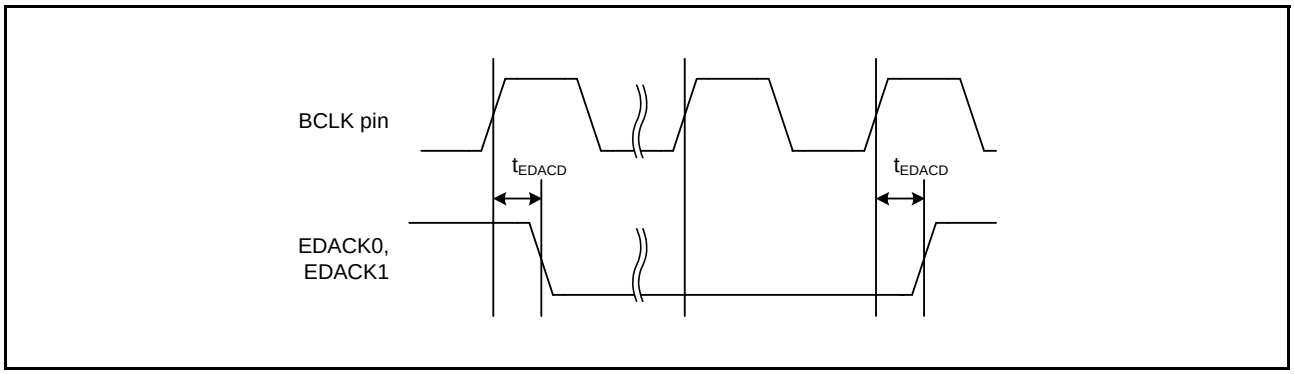


Figure 64.31 EDACK0 and EDACK1 Single-Address Transfer Timing (for a CS Area)

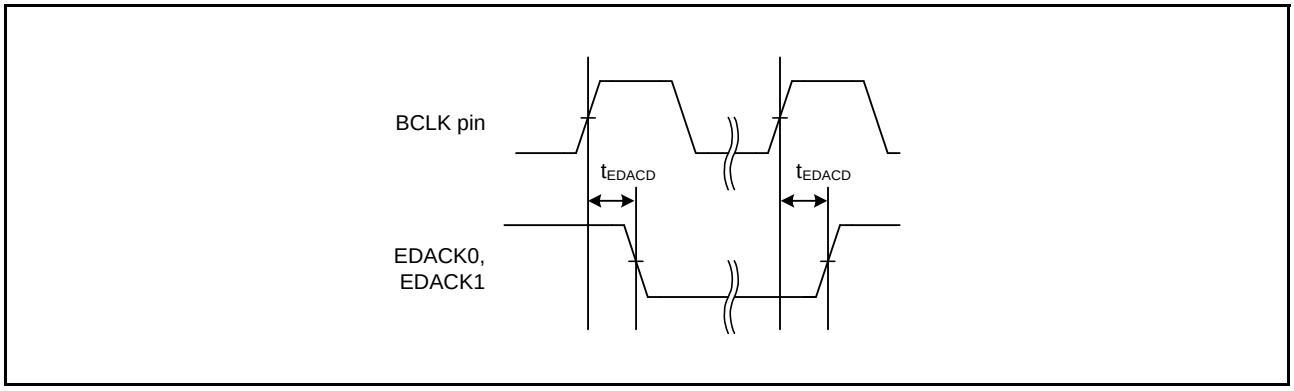


Figure 64.32 EDACK0 and EDACK1 Single-Address Transfer Timing (for SDRAM)

64.3.7 Timing of On-Chip Peripheral Modules

Table 64.23 I/O Port Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item		Symbol	Min.	Max.	Unit*1	Test Conditions
I/O ports	Input data pulse width	t_{PRW}	1.5	—	t_{PBcyc}	Figure 64.33

Note 1. t_{PBcyc}: PCLKB cycle

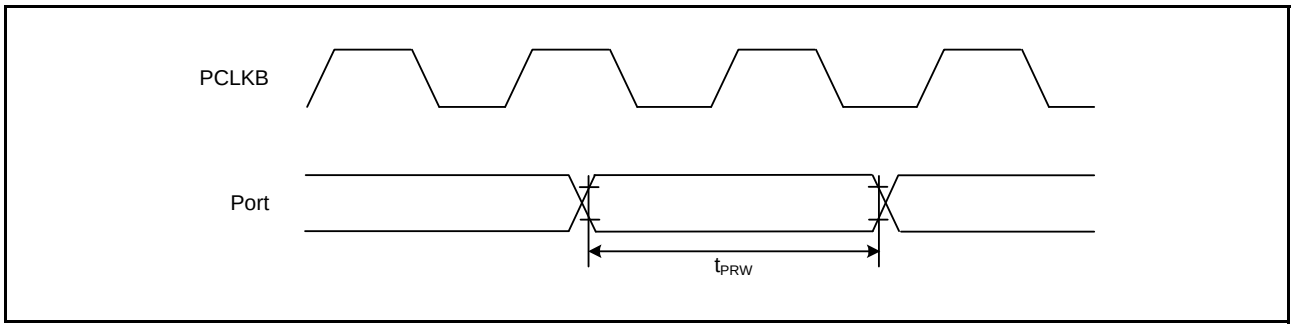


Figure 64.33 I/O Port Input Timing

Table 64.24 TPU Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
TPU	Input capture input pulse width	Single-edge setting	t_{TICW}	1.5	—	t_{PBcyc}	Figure 64.34
		Both-edge setting		2.5	—		
	Timer clock pulse width	Single-edge setting	t_{TCKWH}, t_{TCKWL}	1.5	—	t_{PBcyc}	Figure 64.35
		Both-edge setting		2.5	—		
		Phase counting mode		2.5	—		

Note 1. t_{PBcyc}: PCLKB cycle

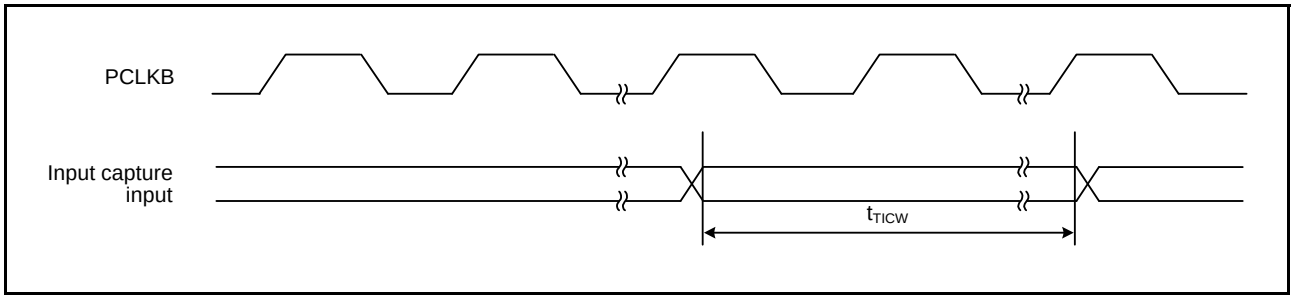


Figure 64.34 TPU Input Capture Input Timing

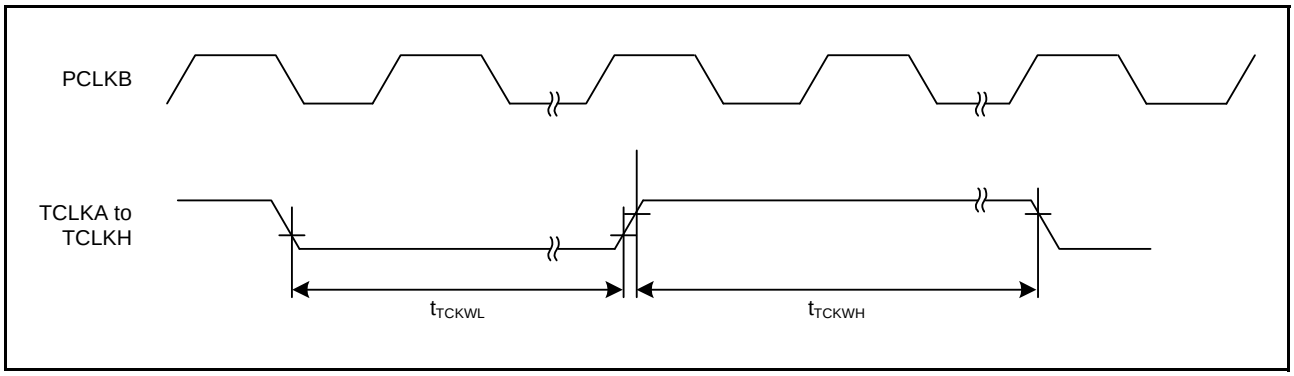


Figure 64.35 TPU Clock Input Timing

Table 64.25 TMR Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
TMR	Timer clock pulse width	Single-edge setting	t_{TMCWH}, t_{TMCWL}	1.5	—	t_{PBcyc}	Figure 64.36
		Both-edge setting		2.5	—		

Note 1. t_{PBcyc}: PCLKB cycle

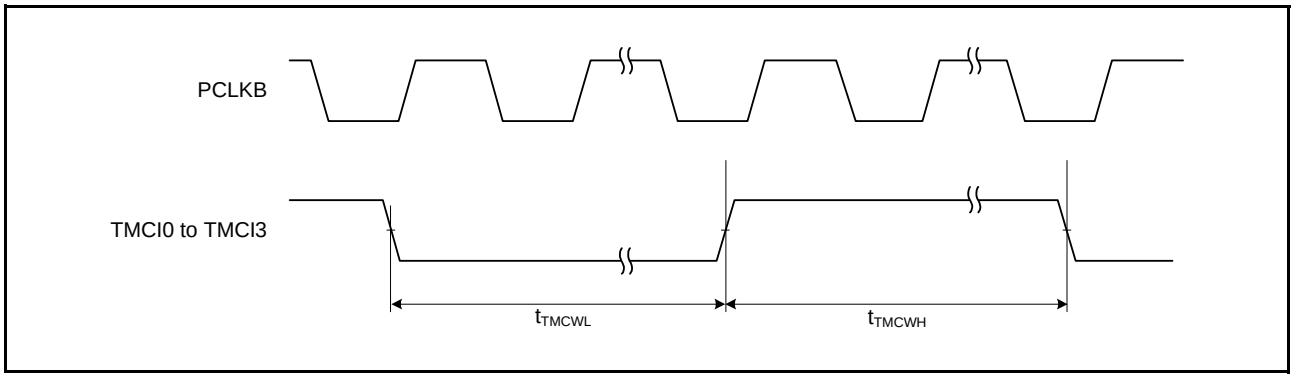


Figure 64.36 TMR Clock Input Timing

Table 64.26 CMTW Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
CMTW	Input capture input pulse width	Single-edge setting	$t_{CMTWTICW}$	1.5	—	t_{PBcyc}	Figure 64.37
		Both-edge setting		2.5	—		

Note 1. t_{PBcyc}: PCLKB cycle

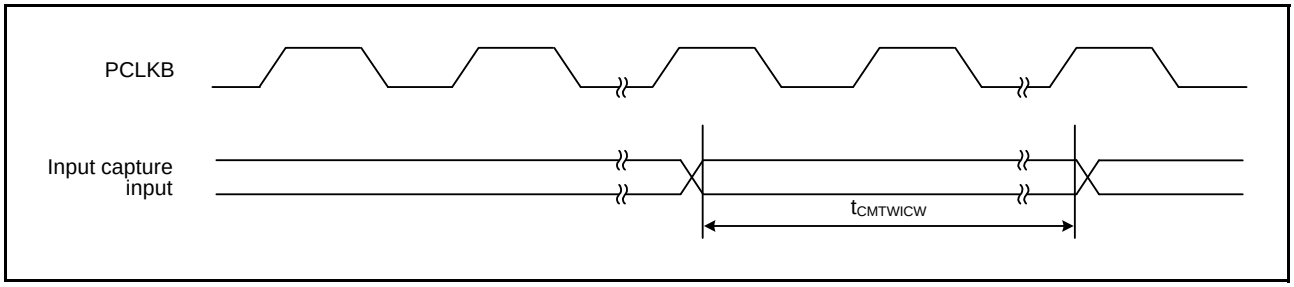


Figure 64.37 CMTW Input Capture Input Timing

Table 64.27 MTU3 Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
MTU3	Input capture input pulse width	Single-edge setting	t_{MTICW}	1.5	—	t_{PAcyc}	Figure 64.38
		Both-edge setting		2.5	—		
	Timer clock pulse width	Single-edge setting	t_{MTCKWH}, t_{MTCKWL}	1.5	—	t_{PAcyc}	Figure 64.39
		Both-edge setting		2.5	—		
		Phase counting mode		2.5	—		

Note 1. t_{PAcyc}: PCLKA cycle

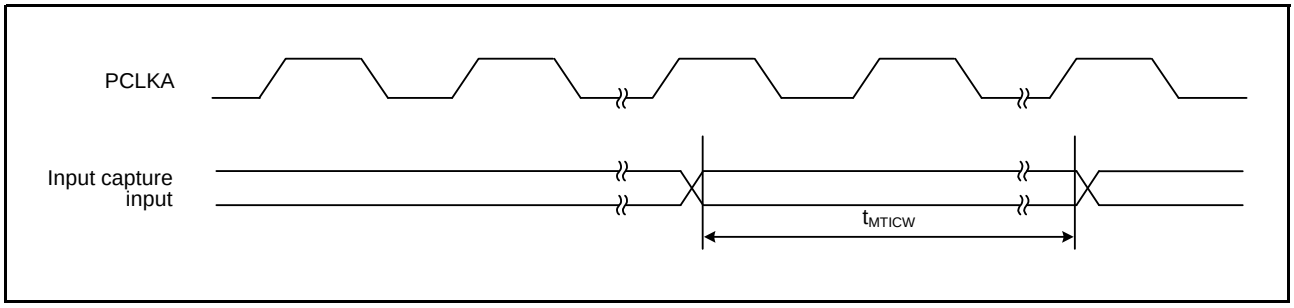


Figure 64.38 MTU3 Input Capture Input Timing

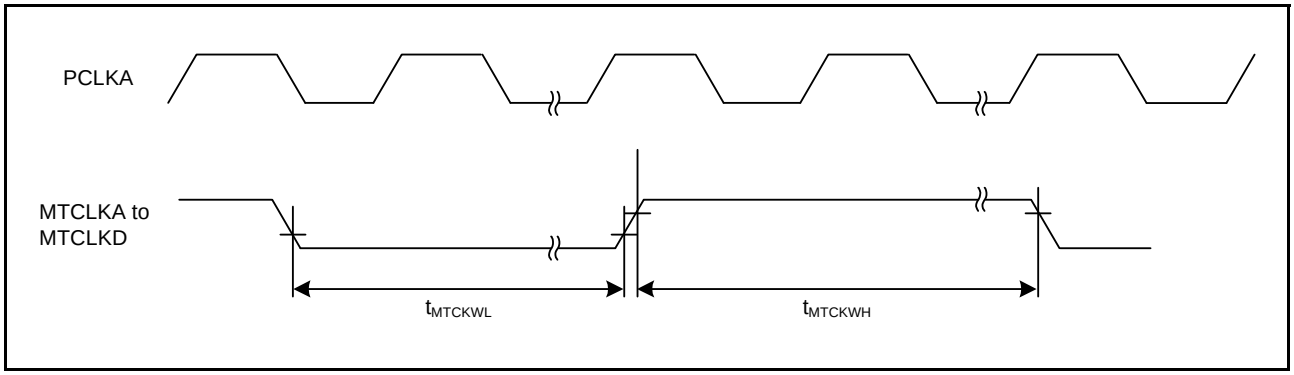


Figure 64.39 MTU3 Clock Input Timing

Table 64.28 POE3 Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item		Symbol	Min.	Max.	Unit*1	Test Conditions
POE	POE# input pulse width	t_{POEW}	1.5	—	t_{PBcyc}	Figure 64.40

Note 1. t_{PBcyc}: PCLKB cycle

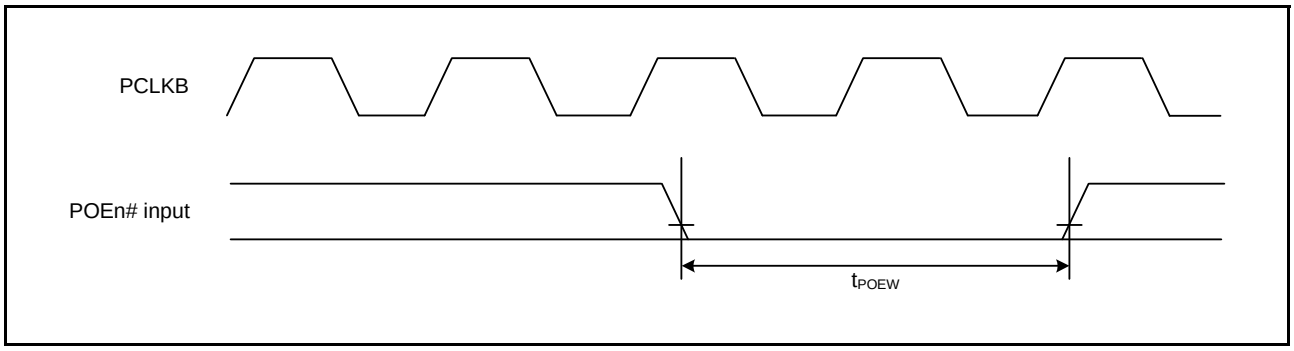


Figure 64.40 POE# Input Timing

Table 64.29 GPT Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
GPT	Input capture input pulse width	Single-edge setting	t_{GTICW}	3	—	t_{PAcyc}	Figure 64.41
		Both-edge setting		5	—		
	External trigger input pulse width	Single-edge setting	t_{OTETW}	1.5	—	t_{PAcyc}	Figure 64.42
		Both-edge setting		2.5	—		

Note 1. t_{PAcyc}: PCLKA cycle

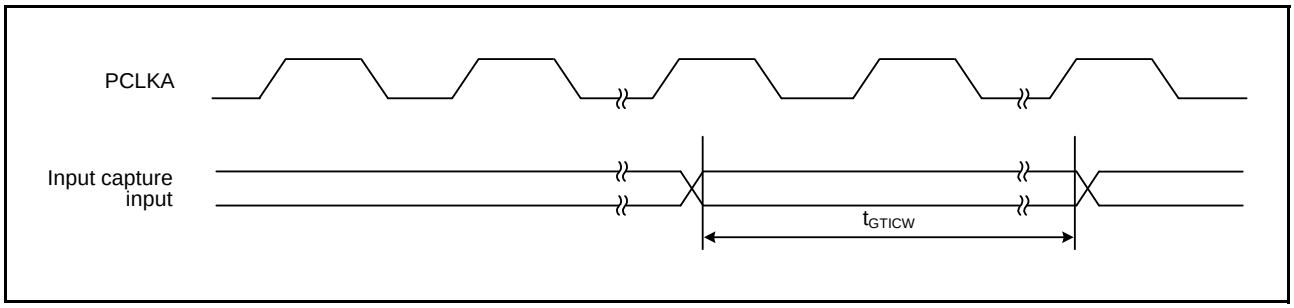


Figure 64.41 GPT Input Capture Input Timing

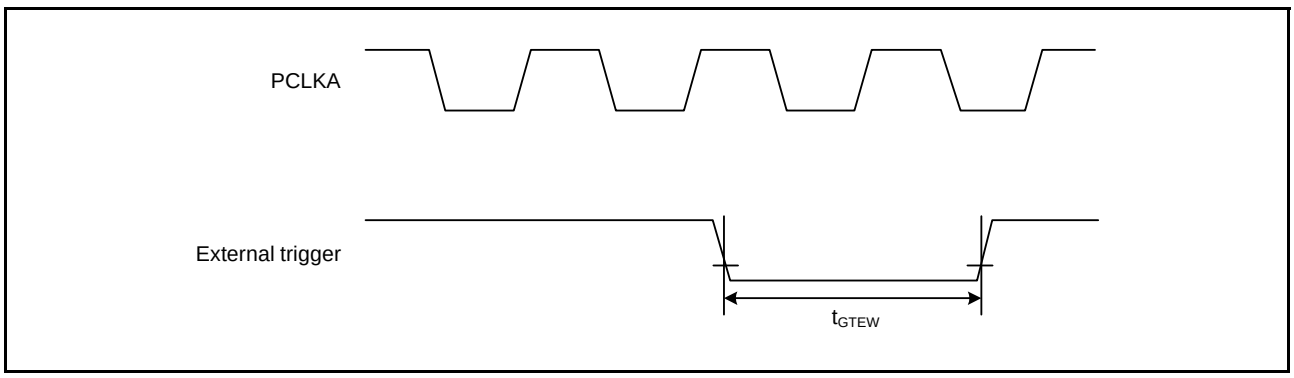


Figure 64.42 GPT External Trigger Input Timing

Table 64.30 A/D Converter Trigger Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item		Symbol	Min.	Max.	Unit*1	Test Conditions
A/D converter	A/D converter trigger input pulse width	t_{TRGW}	1.5	—	t_{PBcyc}	Figure 64.43

Note 1. t_{PBcyc}: PCLKB cycle

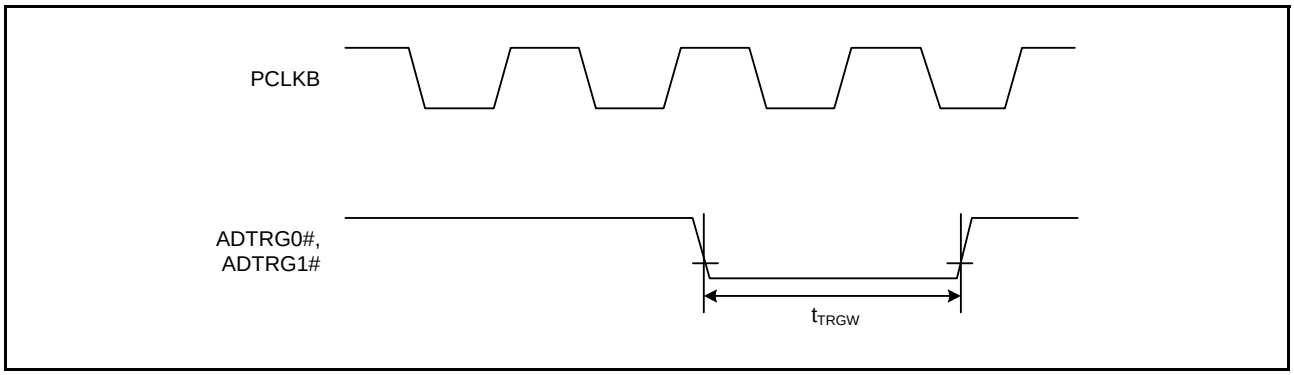


Figure 64.43 A/D Converter Trigger Input Timing

Table 64.31 CAC Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item*1, *2			Symbol	Min.*1	Max.	Unit*1	Test Conditions
CAC	CACREF input pulse width	$t_{PBcyc} \le t_{cac}$	t_{CACREF}	4.5 t_{cac} + 3 t_{PBcyc}	—	ns	
		$t_{PBcyc} > t_{cac}$		5 t_{cac} + 6.5 t_{PBcyc}	—		

Note 1. t_{PBcyc}: PCLKB cycle
Note 2. t_{CAC}: CAC count clock source cycle

Table 64.32 SCI and SCIF Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item			Symbol	Min.*1	Max.*1	Unit*1	Test Conditions
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{PBcyc}	Figure 64.44
		Clock synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	5	ns	
	Input clock fall time		t_{SCKf}	—	5	ns	
	Output clock cycle	Asynchronous*2	t_{Scyc}	8	—	t_{PBcyc}	
		Clock synchronous		4	—		
	Output clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time		t_{SCKr}	—	5	ns	
	Output clock fall time		t_{SCKf}	—	5	ns	
	Transmit data delay time	Clock synchronous	t_{TXD}	—	28	ns	Figure 64.45
	Receive data setup time	Clock synchronous	t_{RXS}	15	—	ns	
	Receive data hold time	Clock synchronous	t_{RXH}	5	—	ns	
SCIF	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{PAcyc}	Figure 64.44
		Clock synchronous		12	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	5	ns	
	Input clock fall time		t_{SCKf}	—	5	ns	
	Output clock cycle	Asynchronous*3	t_{Scyc}	8	—	t_{PAcyc}	
		Clock synchronous		4	—		
	Output clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time		t_{SCKr}	—	5	ns	
	Output clock fall time		t_{SCKf}	—	5	ns	
	Transmit data delay time	Master	t_{TXD}	—	10	ns	Figure 64.45
		Slave		—	$4 \times t_{PAcyc} + 20$		
	Receive data setup time	Master	t_{RXS}	$3 \times t_{PAcyc} + 20$	—	ns	
		Slave		$t_{PAcyc} + 10$	—		
	Receive data hold time	Master	t_{RXH}	$-3 \times t_{PAcyc} + 5$	—	ns	
		Slave		$2 \times t_{PAcyc} + 10$	—		

Note 1. t_{PBcyc}: PCLKB cycle; t_{PAcyc}: PCLKA cycle
Note 2. When the SEMR.ABCS and SEMR.BGDM bits are set to 1
Note 3. When the SEMR.ABCS0 and SEMR.BGDM bits are set to 1

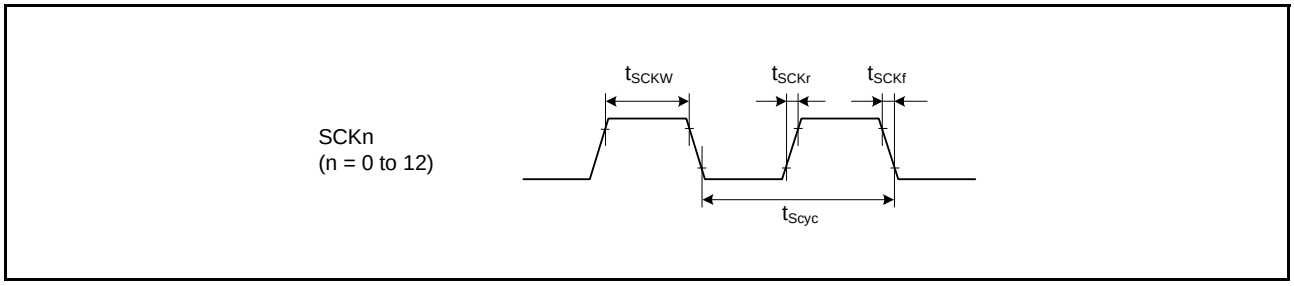


Figure 64.44 SCK Clock Input Timing

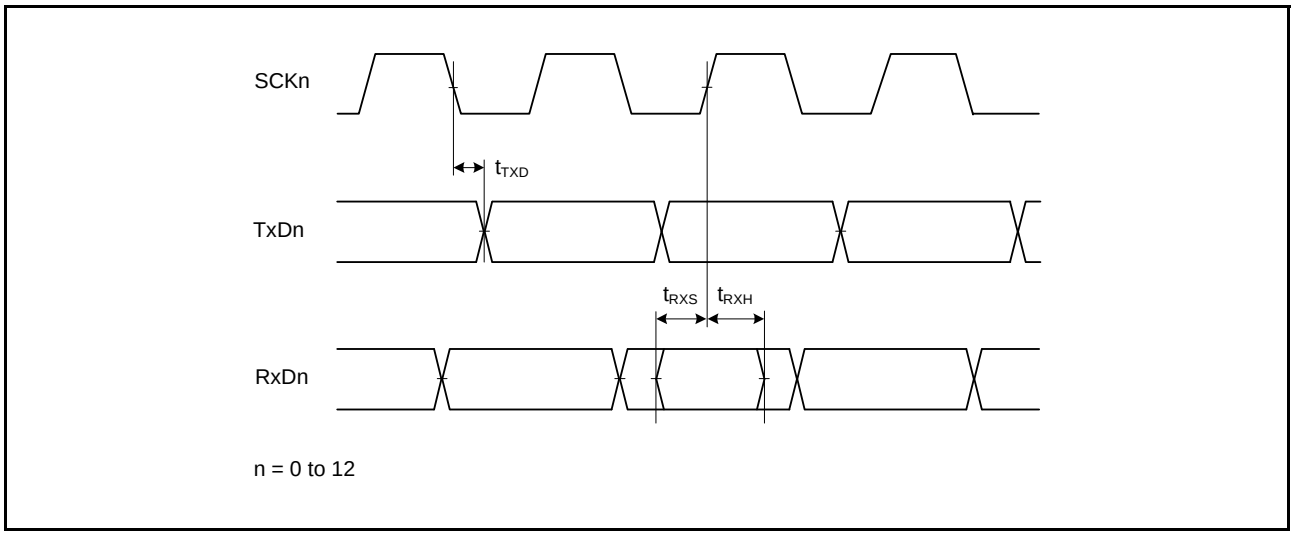


Figure 64.45 SCI Input/Output Timing: Clock Synchronous Mode

Table 64.33 RSPI Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item				Symbol	Min.*1	Max.*1	Unit*1	Test Conditions*2
RSPI	RSPCK clock cycle	Master		t_{SPcyc}	2	4096	t_{PAcyc}	Figure 64.46
		Slave			8	4096		
	RSPCK clock high pulse width	Master		t_{SPCKWH}	$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF}) / 2 - 3$	—	ns	
		Slave			$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF}) / 2$	—		
	RSPCK clock low pulse width	Master		t_{SPCKWL}	$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF}) / 2 - 3$	—	ns	
		Slave			$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF}) / 2$	—		
	RSPCK clock rise/fall time	Output		t_{SPCKr}, t_{SPCKf}	—	5	ns	
		Input			—	1	µs	
	Data input setup time	Master		t_{SU}	6	—	ns	Figure 64.47 to Figure 64.52
		Slave			$8.3 - t_{PAcyc}$	—		
	Data input hold time	Master	PCLKA division ratio set to 1/2	t_{HF}	0	—	ns	
			PCLKA division ratio set to a value other than 1/2	t_H	t_{PAcyc}	—		
		Slave			$8.3 + 2 \times t_{PAcyc}$	—		
	SSL setup time	Master		t_{LEAD}	1	8	t_{SPcyc}	
		Slave			4	—	t_{PAcyc}	
	SSL hold time	Master		t_{LAG}	1	8	t_{SPcyc}	
		Slave			4	—	t_{PAcyc}	
	Data output delay time	Master		t_{OD}	—	6.3	ns	
		Slave			—	$3 \times t_{PAcyc} + 20$		
	Data output hold time	Master		t_{OH}	0	—	ns	
		Slave			0	—		
	Successive transmission delay time	Master		t_{TD}	$t_{SPcyc} + 2 \times t_{PAcyc}$	$8 \times t_{SPcyc} + 2 \times t_{PAcyc}$	ns	
		Slave			$4 \times t_{PAcyc}$	—		
	MOSI and MISO rise/fall time	Output		t_{Dr}, t_{Df}	—	5	ns	
		Input			—	1	µs	
	SSL rise/fall time	Output		t_{SSLr}, t_{SSLf}	—	5	ns	
		Input			—	1	µs	
	Slave access time			t_{SA}	—	4	t_{PAcyc}	Figure 64.51, Figure 64.52
	Slave output release time			t_{REL}	—	3	t_{PAcyc}	

Note 1. t_{PAcyc}: PCLKA cycle
Note 2. We recommend using pins that have a letter (“-A”, “-B”, etc.) to indicate group membership appended to their names as groups. For the RSPI interface, the AC portion of the electrical characteristics is measured for each group.

Table 64.34 Simple SPI Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item		Symbol	Min.	Max.	Unit*1	Test Conditions
Simple SPI	SCK clock cycle output (master)	t_{SPcyc}	4	65536	t_{PBcyc}	Figure 64.46
	SCK clock cycle input (slave)		8	65536		
	SCK clock high pulse width	t_{SPCKWH}	0.4	0.6	t_{SPcyc}	
	SCK clock low pulse width	t_{SPCKWL}	0.4	0.6	t_{SPcyc}	
	SCK clock rise/fall time	t_{SPCKr}, t_{SPCKf}	—	20	ns	
	Data input setup time	t_{SU}	33.3	—	ns	Figure 64.47 to Figure 64.52
	Data input hold time	t_H	33.3	—	ns	
	SS input setup time	t_{LEAD}	1	—	t_{SPcyc}	
	SS input hold time	t_{LAG}	1	—	t_{SPcyc}	
	Data output delay time	t_{OD}	—	33.3	ns	
	Data output hold time	t_{OH}	–10	—	ns	
	Data rise/fall time	t_{Dr}, t_{Df}	—	16.6	ns	
	SS input rise/fall time	t_{SSLr}, t_{SSLf}	—	16.6	ns	
	Slave access time	t_{SA}	—	5	t_{PBcyc}	Figure 64.51, Figure 64.52
	Slave output release time	t_{REL}	—	5	t_{PBcyc}	

Note 1. t_{PBcyc}: PCLKB cycle

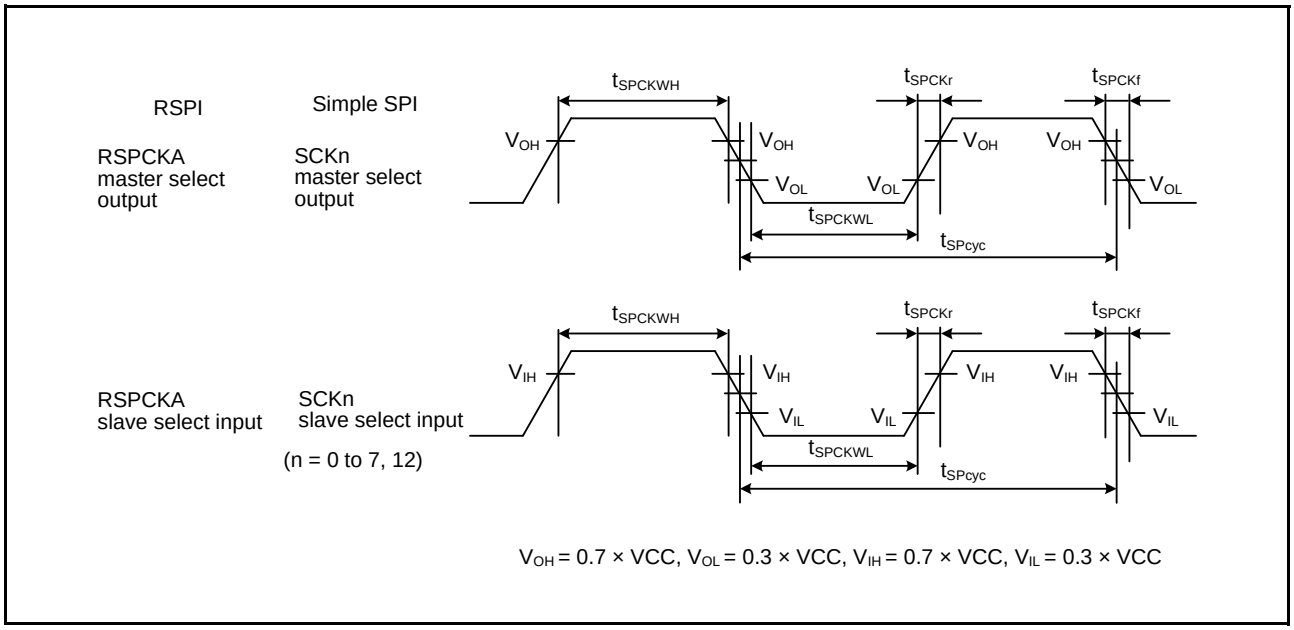


Figure 64.46 RSPI Clock Timing and Simple SPI Clock Timing

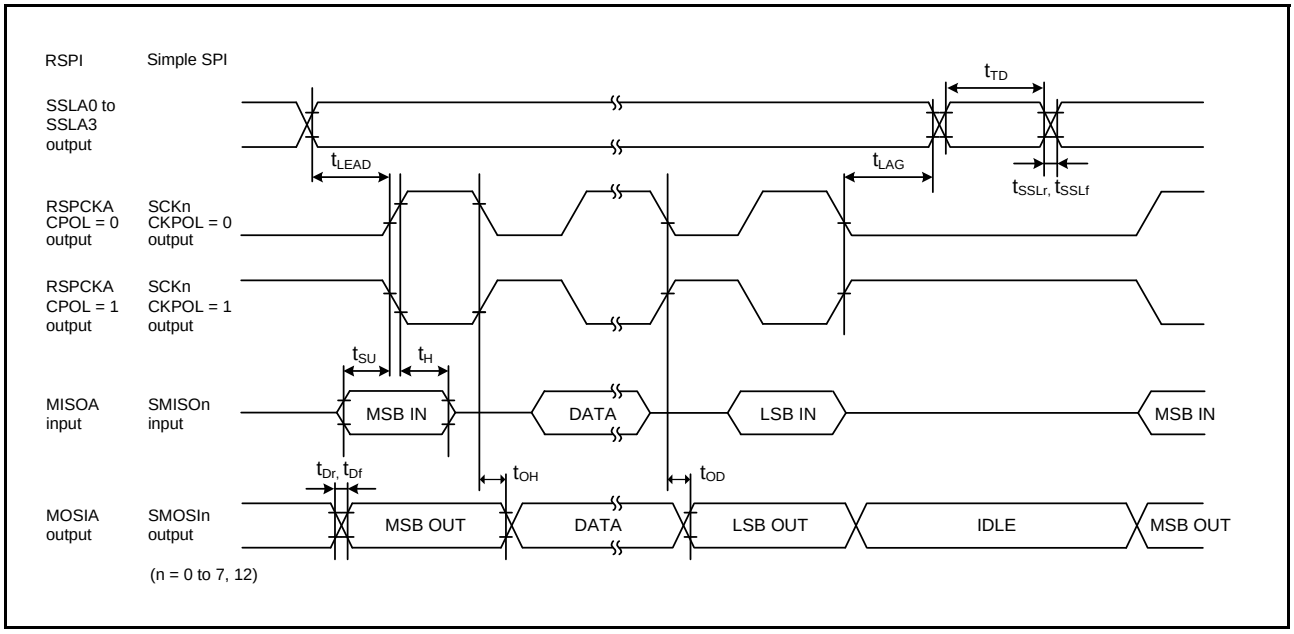


Figure 64.47 RSPI Timing (Master, CPHA = 0) (Bit Rate: PCLKB Division Ratio Set to a Value Other Than 1/2) and Simple SPI Timing (Master, CKPH = 1)

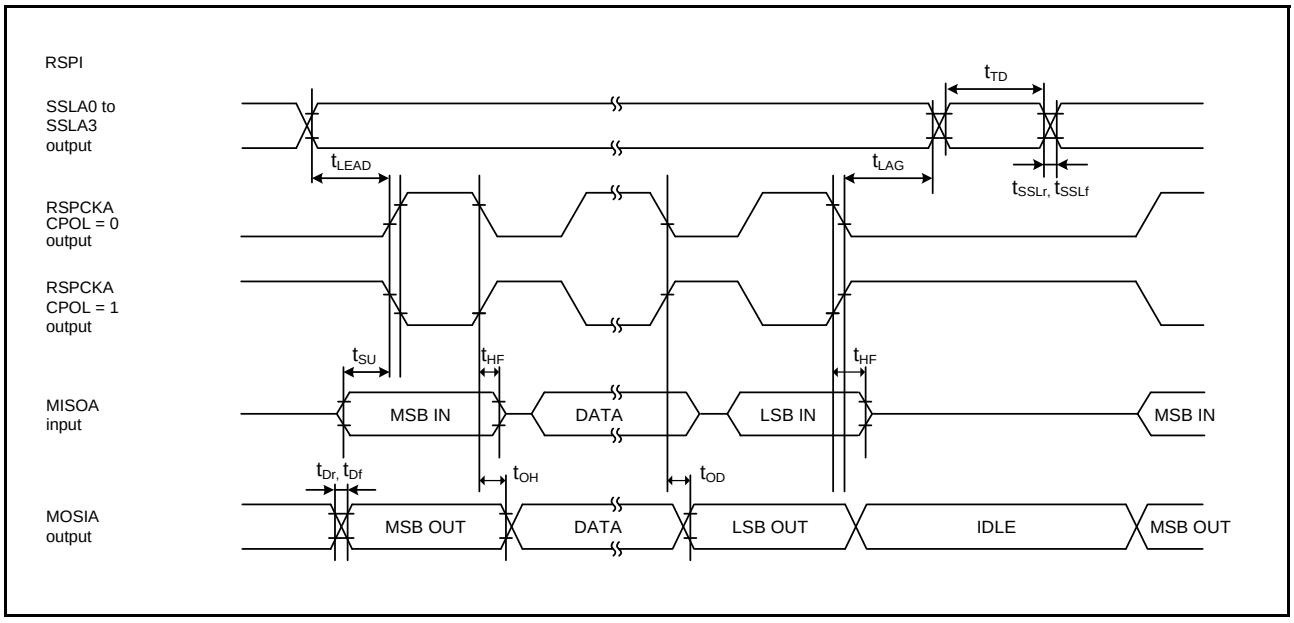


Figure 64.48 RSPI Timing (Master, CPHA = 0) (Bit Rate: PCLKB Division Ratio Set to 1/2)

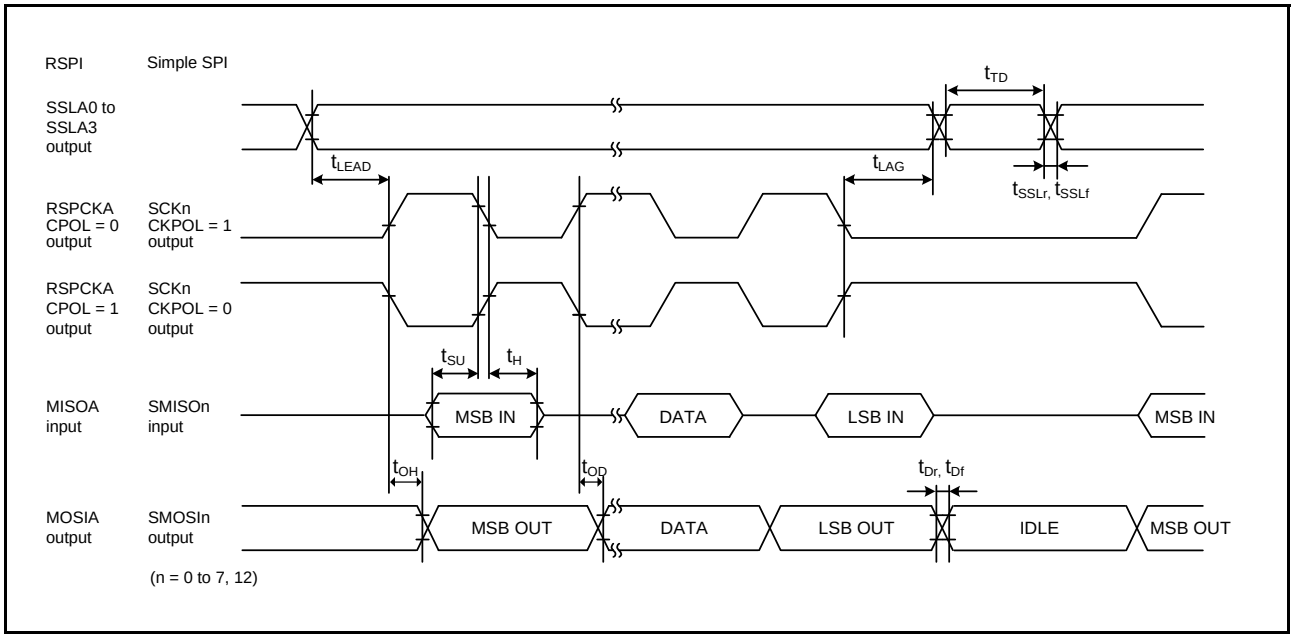


Figure 64.49 RSPI Timing (Master, CPHA = 1) (Bit Rate: PCLKB Division Ratio Set to a Value Other Than 1/2) and Simple SPI Timing (Master, CKPH = 0)

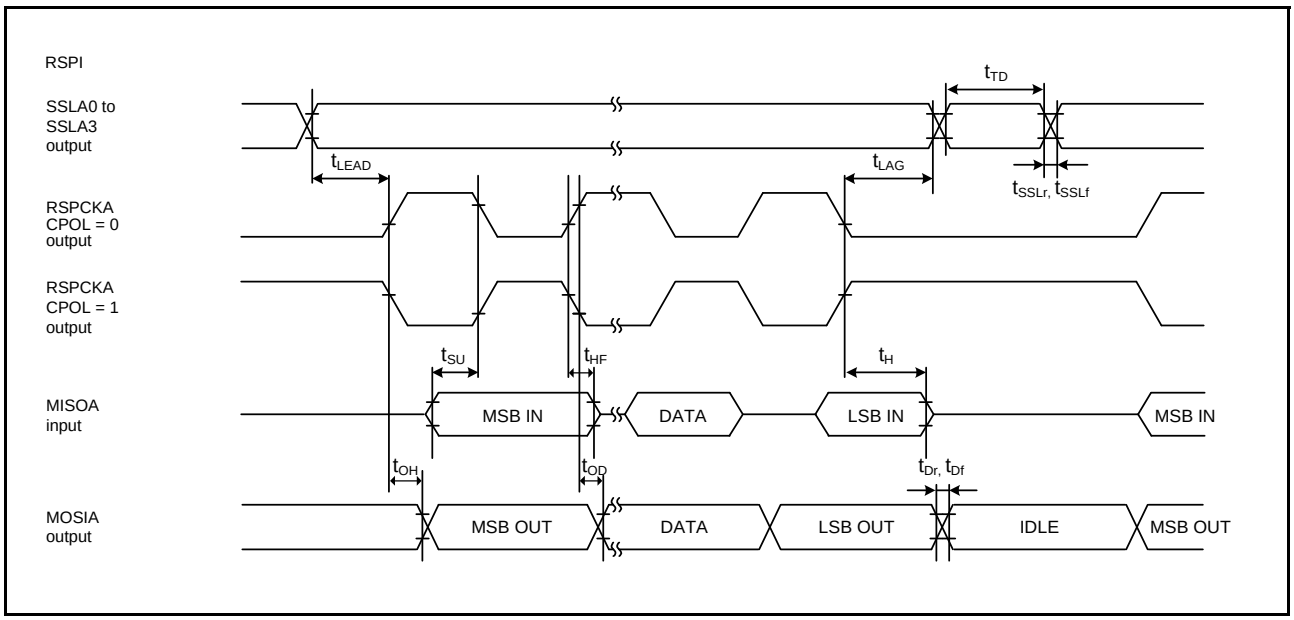


Figure 64.50 RSPI Timing (Master, CPHA = 1) (Bit Rate: PCLKB Division Ratio Set to 1/2)

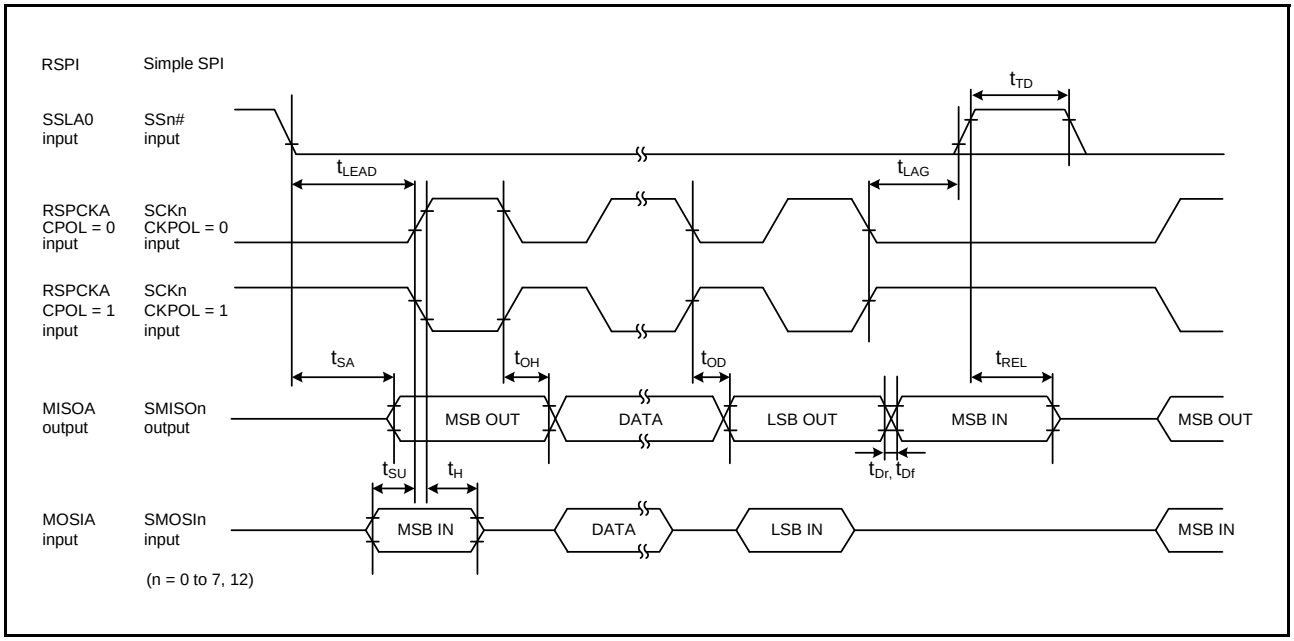


Figure 64.51 RSPI Timing (Slave, CPHA = 0) and Simple SPI Timing (Slave, CKPH = 1)

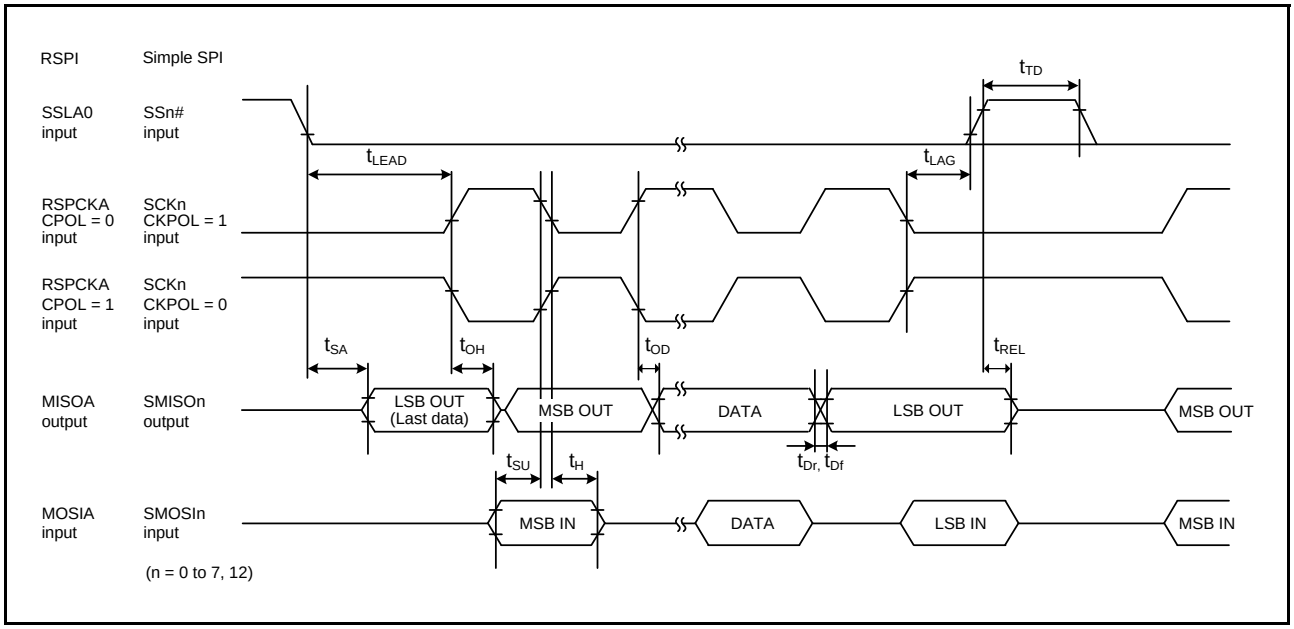


Figure 64.52 RSPI Timing (Slave, CPHA = 1) and Simple SPI Timing (Slave, CKPH = 0)

Table 64.35 QSPI Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item		Symbol	Min.	Max.	Unit*1	Test Conditions*2
QSPI	QSPCLK clock cycle	t_{QScyc}	2	4080	t_{PBcyc}	Figure 64.53
	Data input setup time	t_{Su}	6.5	—	ns	Figure 64.54, Figure 64.55
	Data input hold time	t_{IH}	5	—	ns	
	SS setup time	t_{LEAD}	1.5	8.5	t_{QScyc}	
	SS hold time	t_{LAG}	1	8	t_{QScyc}	
	Data output delay time	t_{OD}	—	10.0	ns	
	Data output hold time	t_{OH}	–5	—	ns	
	Successive transmission delay time	t_{TD}	1	8	t_{QScyc}	

Note 1. t_{PBcyc}: PCLKB cycle
Note 2. We recommend using pins that have a letter (“-A”, “-B”, etc.) to indicate group membership appended to their names as groups. For the QSPI interface, the AC portion of the electrical characteristics is measured for each group.

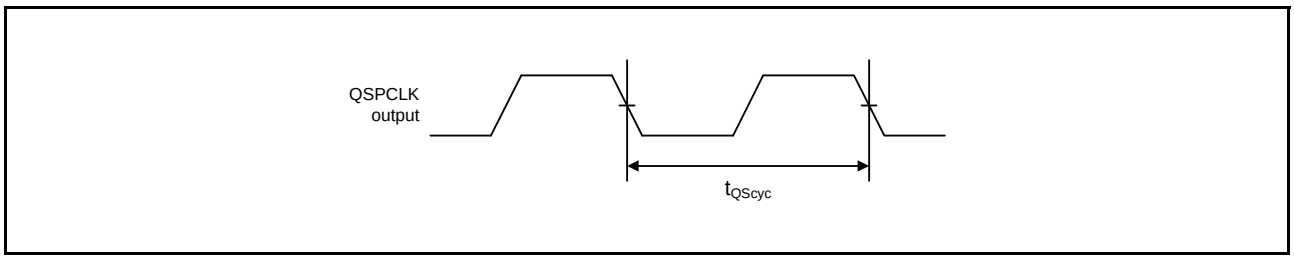


Figure 64.53 QSPI Clock Timing

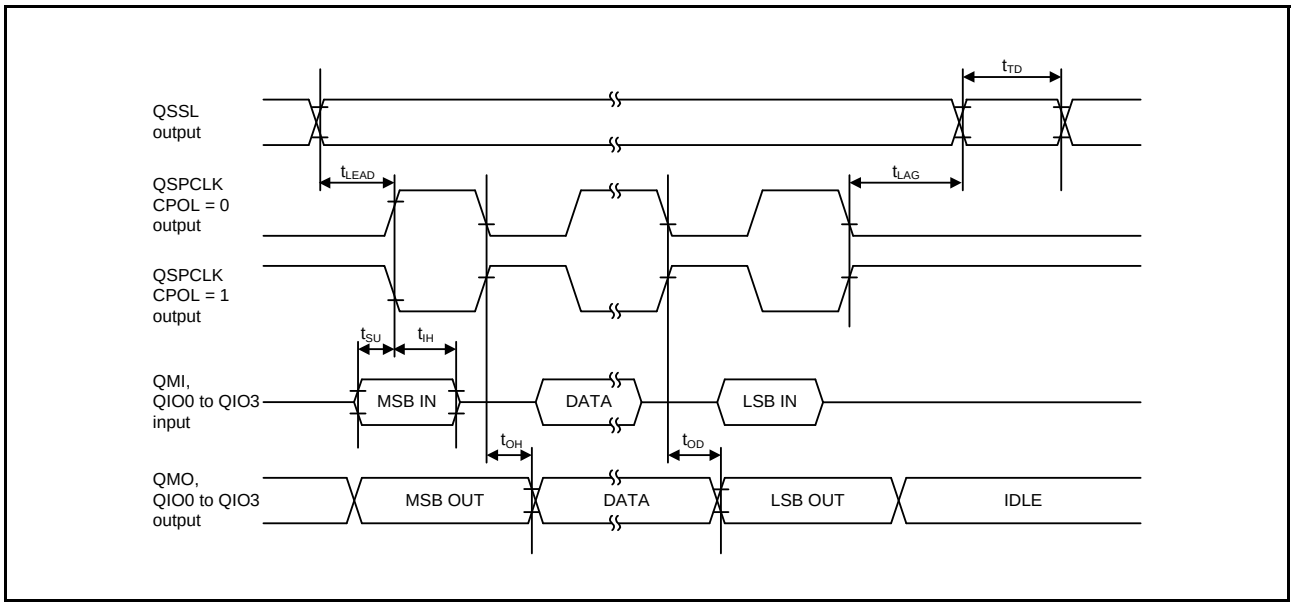


Figure 64.54 Transmit/Receive Timing (CPHA = 0)

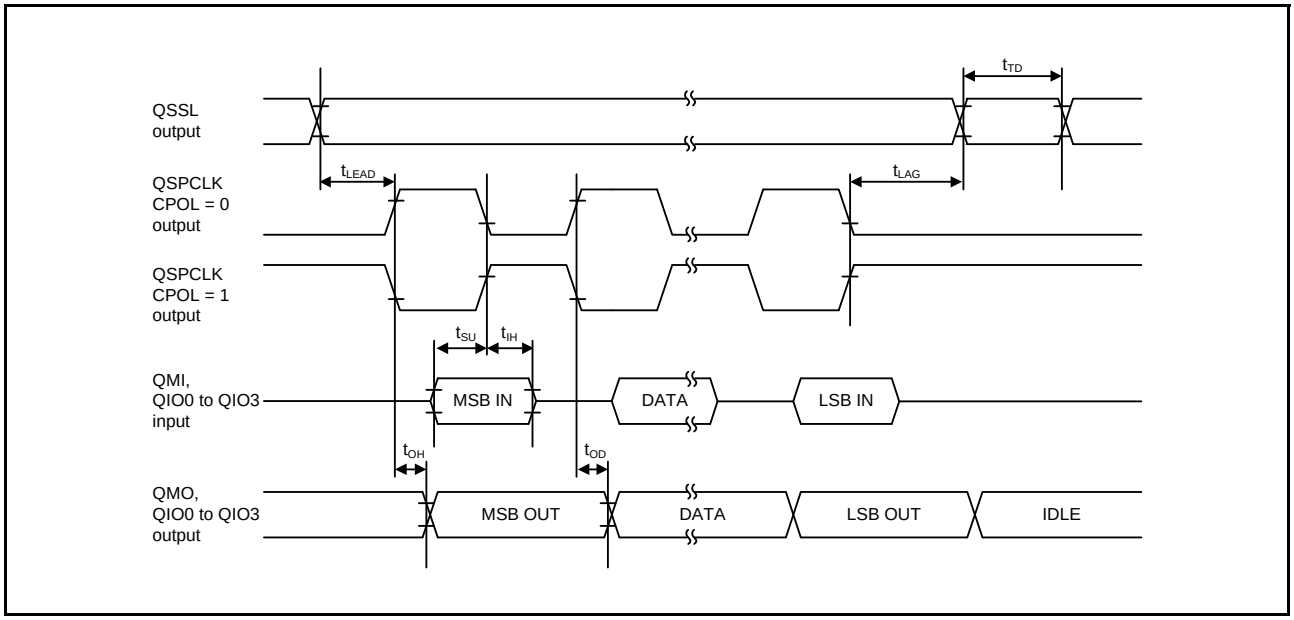


Figure 64.55 Transmit/Receive Timing (CPHA = 1)

Table 64.36 RIIC Timing (1)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
High-drive output is selected by the driving ability control register.

Item		Symbol	Min.*1, *2	Max.	Unit	Test Conditions
RIIC (Standard-mode, SMBus) ICFER.FMPE = 0	SCL input cycle time	t_{SCL}	6(12) × t_{IICcyc} + 1300	—	ns	Figure 64.56
	SCL input high pulse width	t_{SCLH}	3(6) × t_{IICcyc} + 300	—	ns	
	SCL input low pulse width	t_{SCLL}	3(6) × t_{IICcyc} + 300	—	ns	
	SCL, SDA input rise time	t_{Sr}	—	1000	ns	
	SCL, SDA input fall time	t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1(4) × t_{IICcyc}	ns	
	SDA input bus free time	t_{BUF}	3(6) × t_{IICcyc} + 300	—	ns	
	Start condition input hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Restart condition input setup time	t_{STAS}	1000	—	ns	
	Stop condition input setup time	t_{STOS}	1000	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	
RIIC (Fast-mode) ICFER.FMPE = 0	SCL input cycle time	t_{SCL}	6(12) × t_{IICcyc} + 600	—	ns	
	SCL input high pulse width	t_{SCLH}	3(6) × t_{IICcyc} + 300	—	ns	
	SCL input low pulse width	t_{SCLL}	3(6) × t_{IICcyc} + 300	—	ns	
	SCL, SDA input rise time	t_{Sr}	20 × (External pull-up voltage/5.5V)	300	ns	
	SCL, SDA input fall time	t_{Sf}	20 × (External pull-up voltage/5.5V)	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1(4) × t_{IICcyc}	ns	
	SDA input bus free time	t_{BUF}	3(6) × t_{IICcyc} + 300	—	ns	
	Start condition input hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Restart condition input setup time	t_{STAS}	300	—	ns	
	Stop condition input setup time	t_{STOS}	300	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	

Note: t_{IICcyc}: RIIC internal reference clock (IICϕ) cycle
Note 1. The value within parentheses is applicable when the value of the ICMR3.NF[1:0] bits is 11b while the digital filter is enabled by the setting ICFER.NFE = 1.
Note 2. Cb is the total capacitance of the bus lines.

Table 64.37 RIIC Timing (2)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
High-drive output is selected by the driving ability control register.

Item		Symbol	Min.*1, *2	Max.	Unit	Test Conditions
RIIC (Fast-mode+) ICFER.FMPE = 1	SCL input cycle time	t_{SCL}	6(12) × t_{IICcyc} + 240	—	ns	Figure 64.56
	SCL input high pulse width	t_{SCLH}	3(6) × t_{IICcyc} + 120	—	ns	
	SCL input low pulse width	t_{SCLL}	3(6) × t_{IICcyc} + 120	—	ns	
	SCL, SDA input rise time	t_{Sr}	—	120	ns	
	SCL, SDA input fall time	t_{Sf}	—	120	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1(4) × t_{IICcyc}	ns	
	SDA input bus free time	t_{BUF}	3(6) × t_{IICcyc} + 120	—	ns	
	Start condition input hold time	t_{STAH}	t_{IICcyc} + 120	—	ns	
	Restart condition input setup time	t_{STAS}	120	—	ns	
	Stop condition input setup time	t_{STOS}	120	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 20	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	550	pF	
Simple IIC (Standard-mode)	SDA input rise time	t_{Sr}	—	1000	ns	
	SDA input fall time	t_{Sf}	—	300	ns	
	SDA input spike pulse removal time	t_{SP}	0	4 × t_{PBcyc}	ns	
	Data input setup time	t_{SDAS}	250	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	
Simple IIC (Fast-mode)	SCL, SDA input rise time	t_{Sr}	—	300	ns	
	SCL, SDA input fall time	t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	4 × t_{PBcyc}	ns	
	Data input setup time	t_{SDAS}	100	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	

Note: t_{IICcyc}: RIIC internal reference clock (IICϕ) cycle, t_{PBcyc}: PCLKB cycle
Note 1. The value within parentheses is applicable when the value of the ICMR3.NF[1:0] bits is 11b while the digital filter is enabled by the setting ICFER.NFE = 1.
Note 2. Cb is the total capacitance of the bus lines.

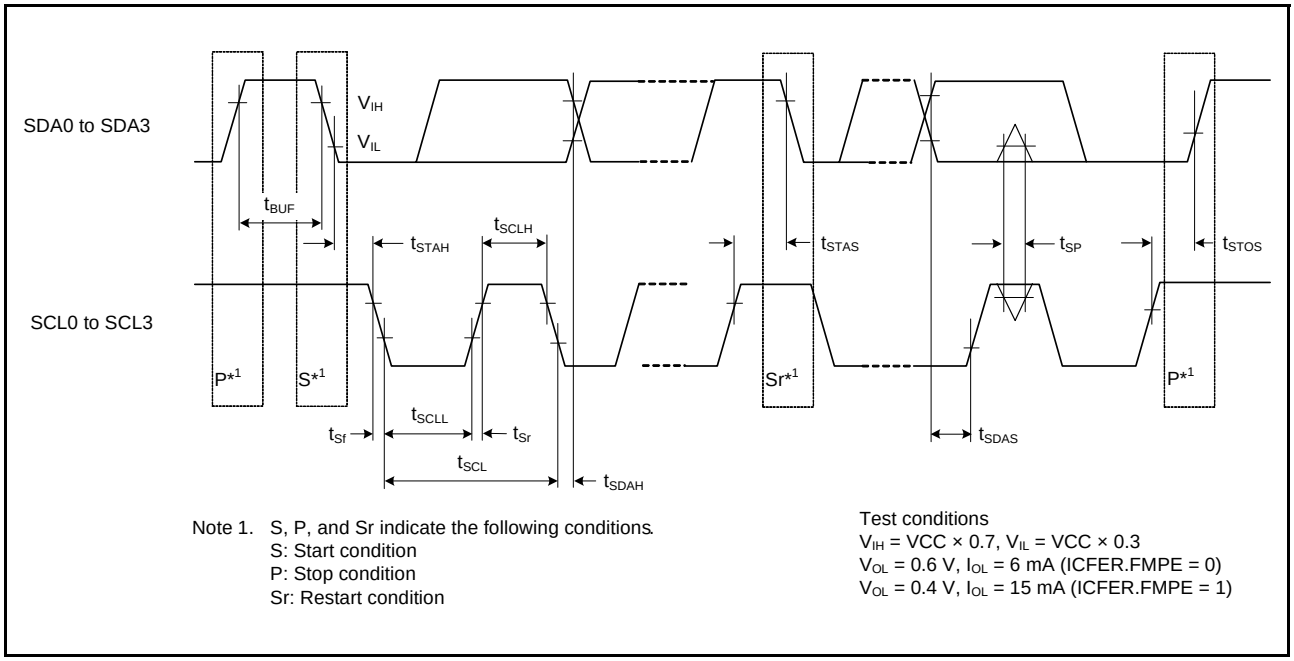


Figure 64.56 RIIC Bus Interface Input/Output Timing and Simple IIC Bus Interface Input/Output Timing

Table 64.38 Serial Sound Interface Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item		Symbol	Min.	Max.	Unit	Test Conditions
SSI	AUDIO_CLK input frequency	t_{AUDIO}	—	50	MHz	
	Output clock cycle	t_O	150	64000	ns	Figure 64.57
	Input clock cycle	t_I	150	64000	ns	
	Clock high level	t_{HC}	60	—	ns	
	Clock low level	t_{LC}	60	—	ns	
	Clock rising time	t_{RC}	—	25	ns	
	Data delay time	t_{DTR}	–5	25	ns	Figure 64.58, Figure 64.59
	Setup time	t_{SR}	25	—	ns	
	Hold time	t_{HTR}	25	—	ns	
	WS change edge SSIDATA output delay	t_{DTRW}	—	25	ns	Figure 64.60

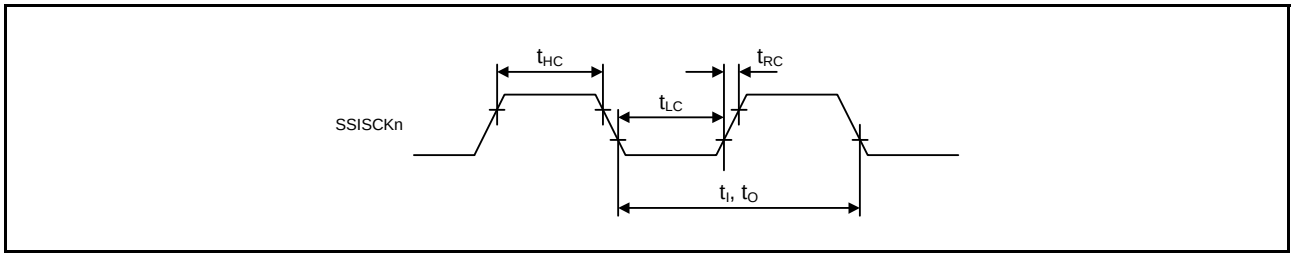


Figure 64.57 Clock Input/Output Timing

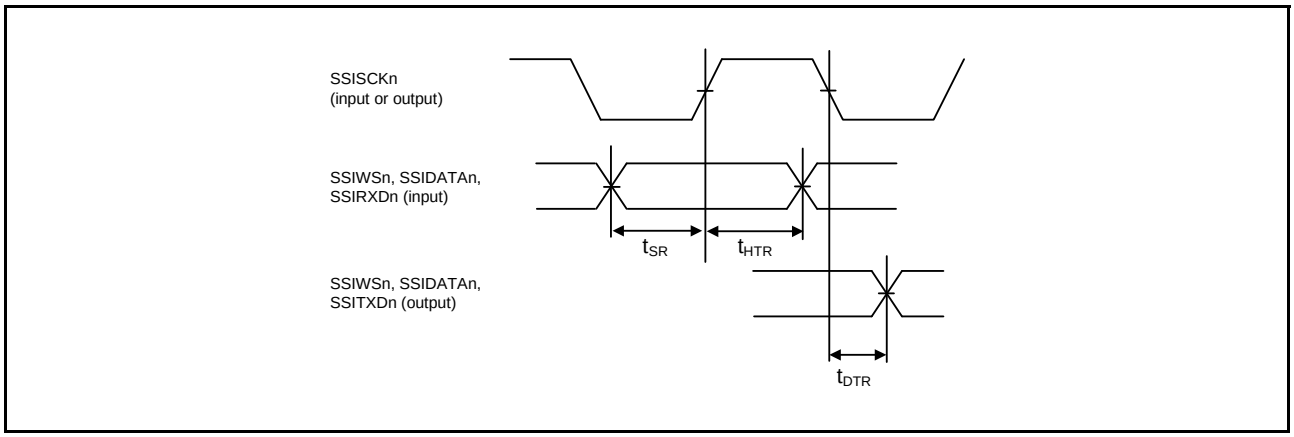


Figure 64.58 Transmit/Receive Timing (SSISCKn Rising Synchronous)

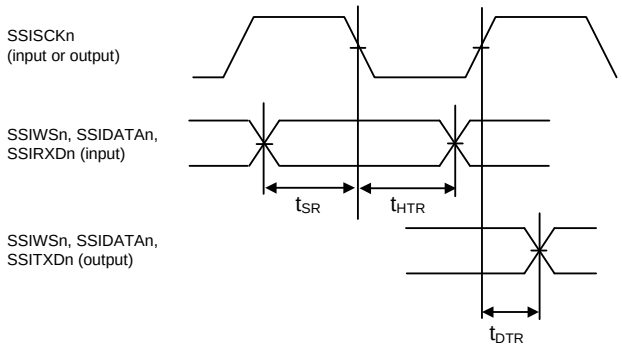
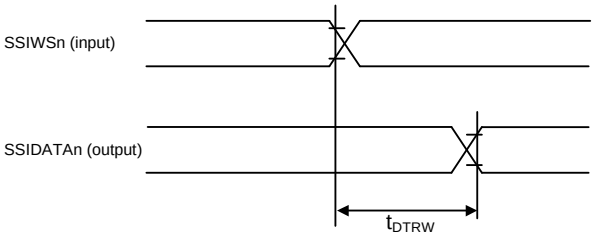


Figure 64.59 Transmit/Receive Timing (SSISCKn Falling Synchronous)



MSB bit output timing in slave transmission from SSIWSn with the settings of DEL = 1, SDTA = 0, or DEL = 1, SDTA = 1, SWL[2:0] = DWL[2:0]

Figure 64.60 SSIDATA Output Delay from SSIWSn Change Edge

Table 64.39 MMC Host Interface Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item		Symbol	Min.*1	Max.	Unit	Test Conditions*2
MMCIF	MMC_CLK clock cycle	t_{MMCPP}	2 × t_{PBcyc}	—	ns	Figure 64.61
	MMC_CLK clock high level width	t_{MMCWH}	6.5	—	ns	
	MMC_CLK clock low level width	t_{MMCWL}	6.5	—	ns	
	MMC_CLK clock rising time	t_{MMCLH}	—	5	ns	
	MMC_CLK clock falling time	t_{MMCHL}	—	5	ns	
	MMC_CMD, MMC_D7 to MMC_D0 output data delay (data transfer mode)	$t_{MMCODLY}$	–6.5	6.5	ns	
	MMC_CMD, MMC_D7 to MMC_D0 input data setup	t_{MMCISU}	8	—	ns	
	MMC_CMD, MMC_D7 to MMC_D0 input data hold	t_{MMCIH}	2	—	ns	

Note 1. t_{PBcyc}: PCLKB cycle
Note 2. We recommend using pins that have a letter (“-A”, “-B”, etc.) to indicate group membership appended to their names as groups. For the MMC interface, the AC portion of the electrical characteristics is measured for each group.

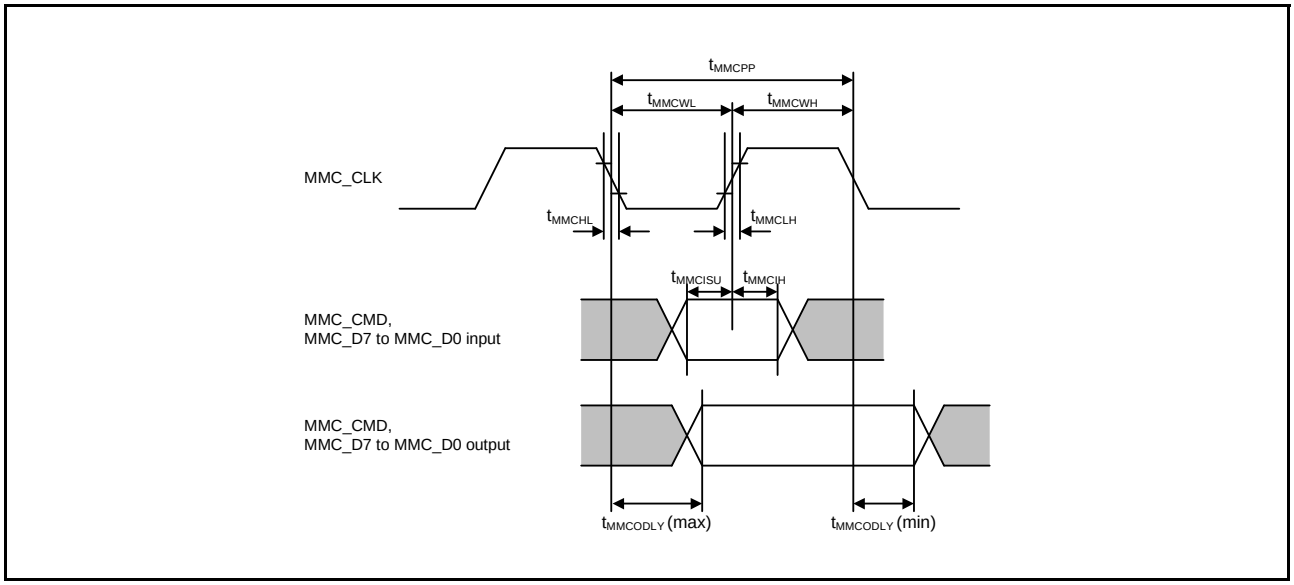


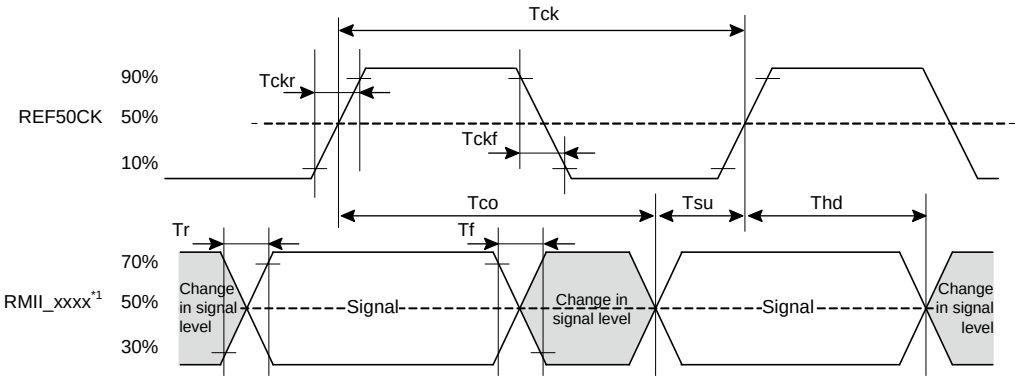
Figure 64.61 MMC Interface

Table 64.40 ETHERC Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item		Symbol	Min.	Max.	Unit	Test Conditions
ETHERC (RMII)	REF50CK cycle time	T_{ck}	20	—	ns	Figure 64.62 to Figure 64.64
	REF50CK frequency Typ. 50 MHz	—	—	50 + 100 ppm	MHz	
	REF50CK duty	—	35	65	%	
	REF50CK rise/fall time	$T_{ckr/ckf}$	0.5	3.5	ns	
	RMII_xxxx*1 output delay time	T_{co}	2.5	15.0	ns	
	RMII_xxxx*2 setup time	T_{su}	3	—	ns	
	RMII_xxxx*2 hold time	T_{hd}	1	—	ns	
	RMII_xxxx*1, *2 rise/fall time	T_r/T_f	0.5	5	ns	
	ET_WOL output delay time	t_{WOLd}	1	23.5	ns	Figure 64.66
ETHERC (MII)	ET_TX_CLK cycle time	t_{Tcyc}	40	—	ns	—
	ET_TX_EN output delay time	t_{TENd}	1	20	ns	Figure 64.67
	ET_ETXD0 to ET_ETXD3 output delay time	t_{MTDd}	1	20	ns	
	ET_CRS setup time	t_{CRSs}	10	—	ns	
	ET_CRS hold time	t_{CRSh}	10	—	ns	
	ET_COL setup time	t_{COLs}	10	—	ns	Figure 64.68
	ET_COL hold time	t_{COLh}	10	—	ns	
	ET_RX_CLK cycle time	t_{TRcyc}	40	—	ns	—
	ET_RX_DV setup time	t_{RDVs}	10	—	ns	Figure 64.69
	ET_RX_DV hold time	t_{RDVh}	10	—	ns	
	ET_ERXD0 to ET_ERXD3 setup time	t_{MRDs}	10	—	ns	
	ET_ERXD0 to ET_ERXD3 hold time	t_{MRDh}	10	—	ns	
	ET_RX_ER setup time	t_{RERs}	10	—	ns	Figure 64.70
	ET_RX_ER hold time	t_{RESh}	10	—	ns	
	ET_WOL output delay time	t_{WOLd}	1	23.5	ns	Figure 64.71

Note 1. RMII_TXD_EN, RMII_TXD1, RMII_TXD0
Note 2. RMII_CRS_DV, RMII_RXD1, RMII_RXD0, RMII_RX_ER



Note 1. RMII_TXD_EN, RMII_TXD1, RMII_TXD0, RMII_CRS_DV, RMII_RXD1, RMII_RXD0, RMII_RX_ER

Figure 64.62 Timing with the REF50CK and RMII Signals

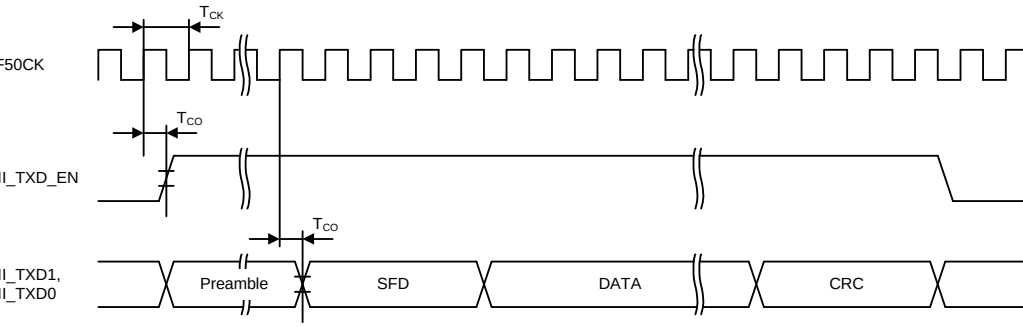


Figure 64.63 RMII Transmission Timing

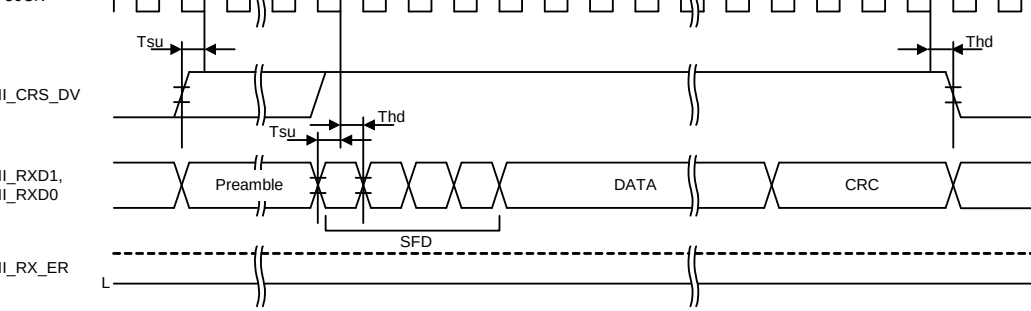


Figure 64.64 RMII Reception Timing (Normal Operation)

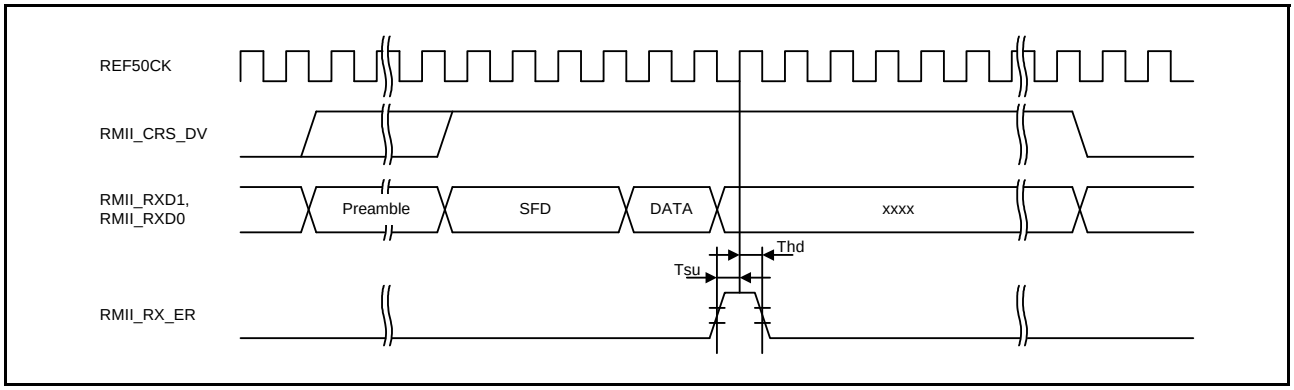


Figure 64.65 RMII Reception Timing (Error Occurrence)

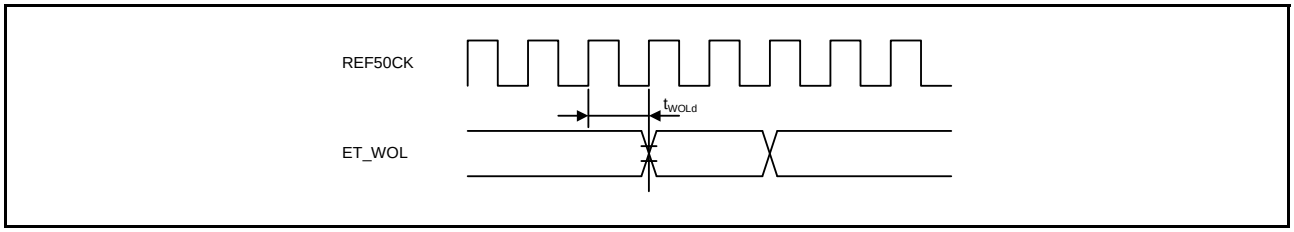


Figure 64.66 WOL Output Timing (RMII)

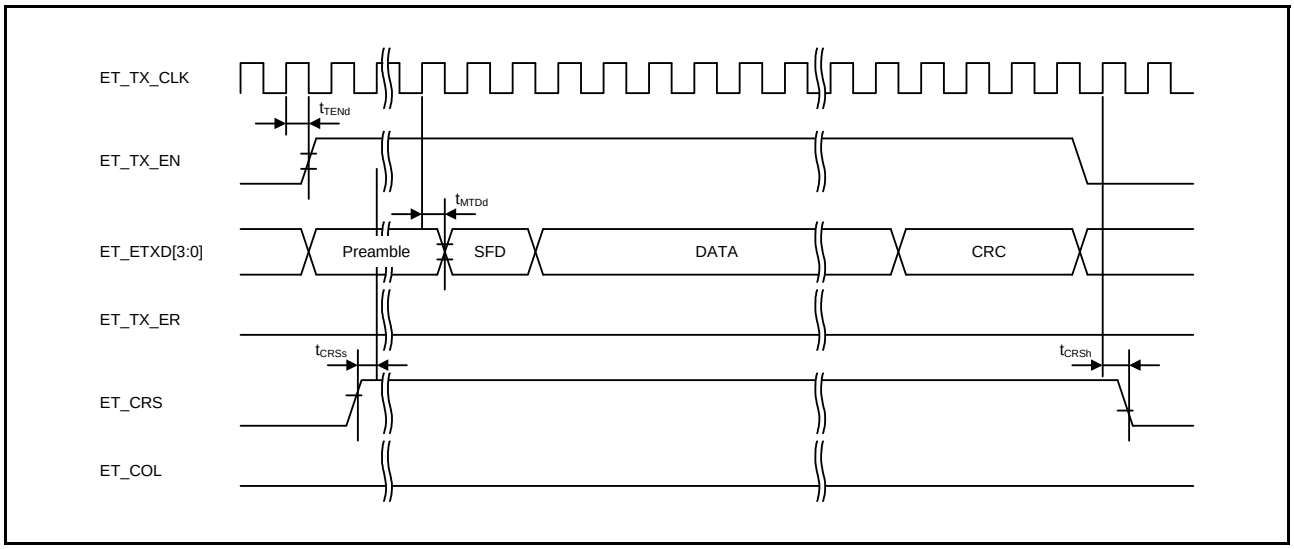


Figure 64.67 MII Transmission Timing (Normal Operation)

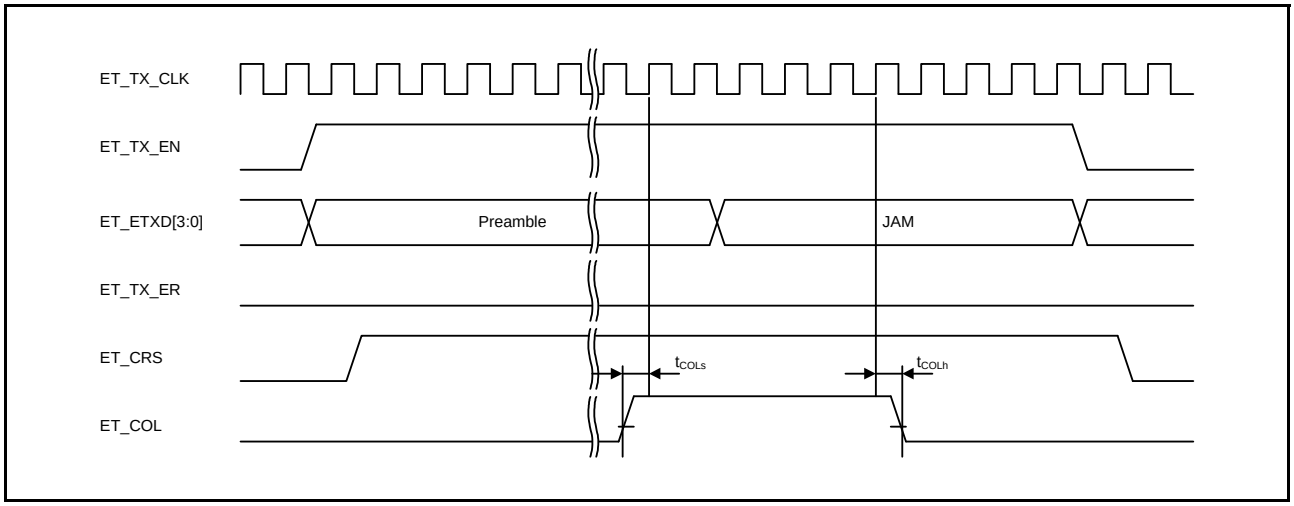


Figure 64.68 MII Transmission Timing (Conflict Occurrence)

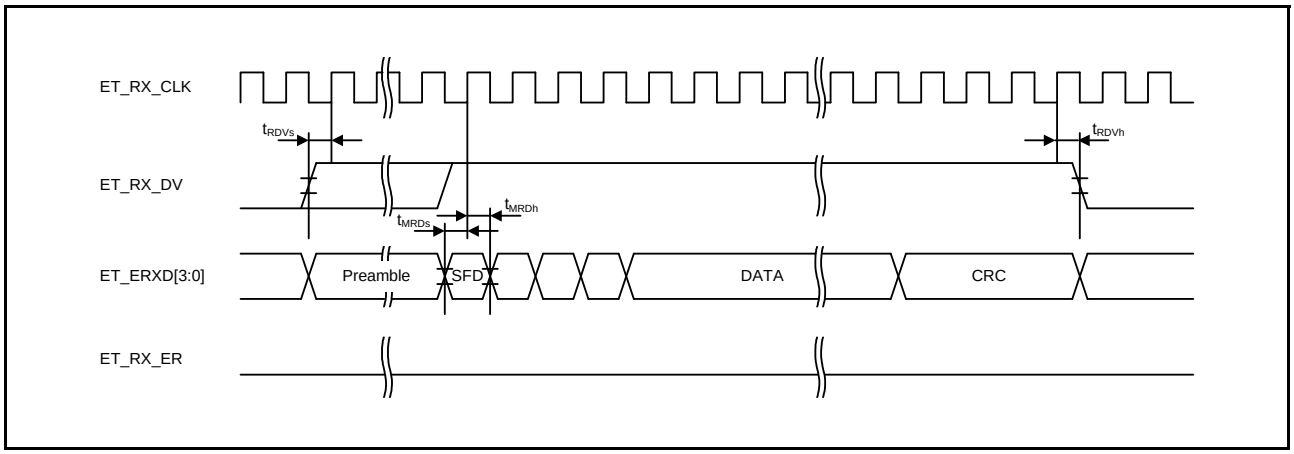


Figure 64.69 MII Reception Timing (Normal Operation)

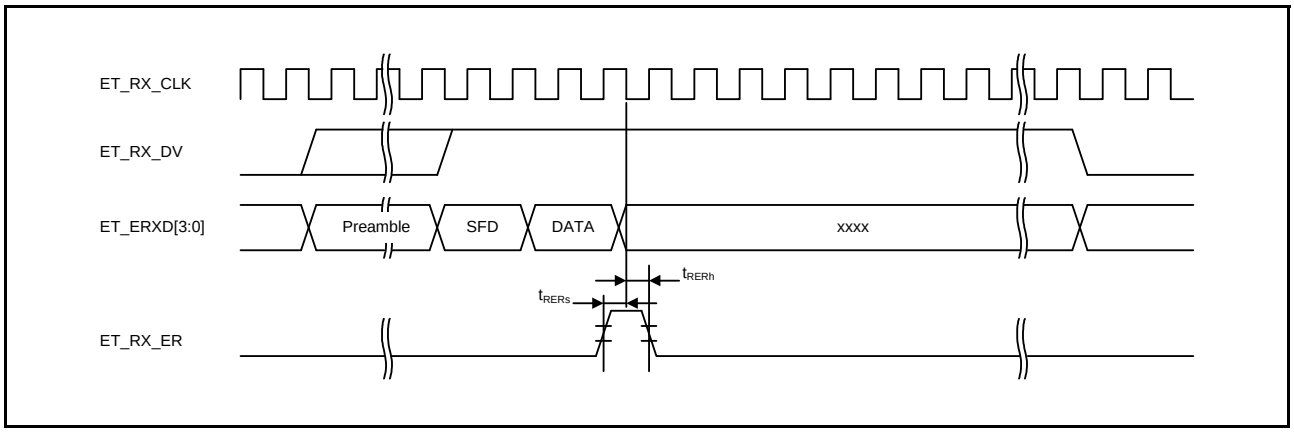


Figure 64.70 MII Reception Timing (Error Occurrence)

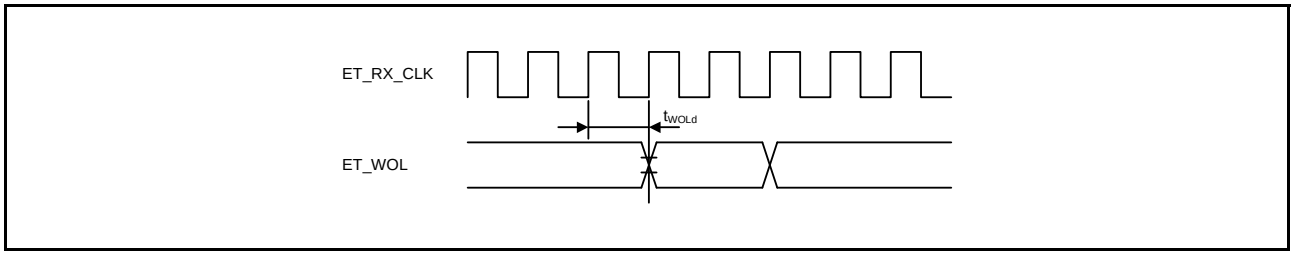


Figure 64.71 WOL Output Timing (MII)

Table 64.41 PDC Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item		Symbol	Min.*1	Max.	Unit	Test Conditions
PDC	PIXCLK input cycle time	t_{PIXcyc}	37	—	ns	Figure 64.72
	PIXCLK input high pulse width	t_{PIXH}	10	—	ns	
	PIXCLK input low pulse width	t_{PIXL}	10	—	ns	
	PIXCLK rising time	t_{PIXr}	—	5	ns	
	PIXCLK falling time	t_{PIXf}	—	5	ns	
	PCKO output cycle time	t_{PCKcyc}	2 × t_{PBcyc}	—	ns	Figure 64.73
	PCKO output high pulse width	t_{PCKH}	(t_{PCKcyc} – t_{PCKr} – t_{PCKf})/2 – 3	—	ns	
	PCKO output low pulse width	t_{PCKL}	(t_{PCKcyc} – t_{PCKr} – t_{PCKf})/2 – 3	—	ns	
	PCKO rising time	t_{PCKr}	—	5	ns	
	PCKO falling time	t_{PCKf}	—	5	ns	
	VSYNV/HSYNC input setup time	t_{SYNCS}	10	—	ns	Figure 64.74
	VSYNV/HSYNC input hold time	t_{SYNCH}	5	—	ns	
	PIXD input setup time	t_{PIXDS}	10	—	ns	
	PIXD input hold time	t_{PIXDH}	5	—	ns	

Note 1. t_{PBcyc}: PCLKB cycle

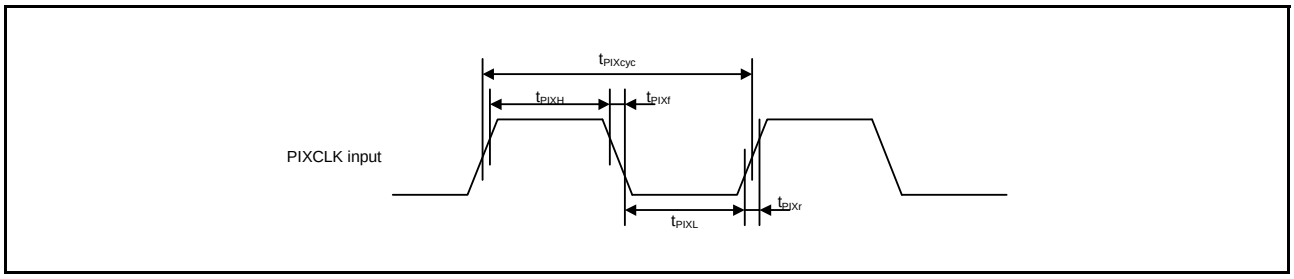


Figure 64.72 PDC Input Clock Timing

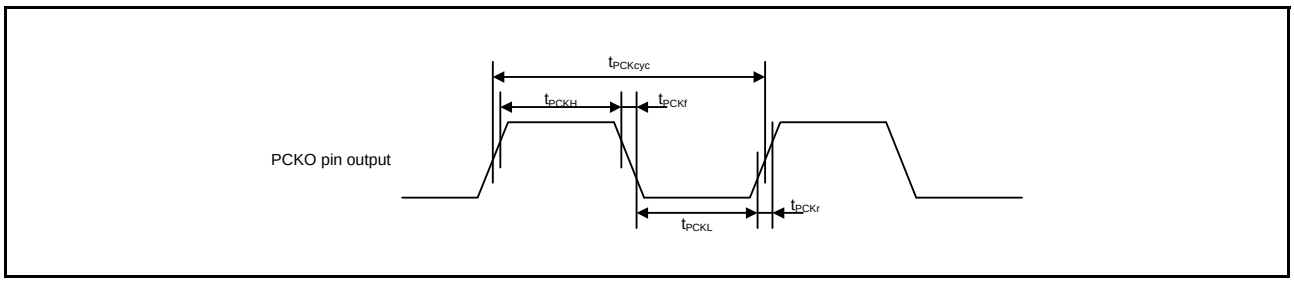


Figure 64.73 PDC Output Clock Timing

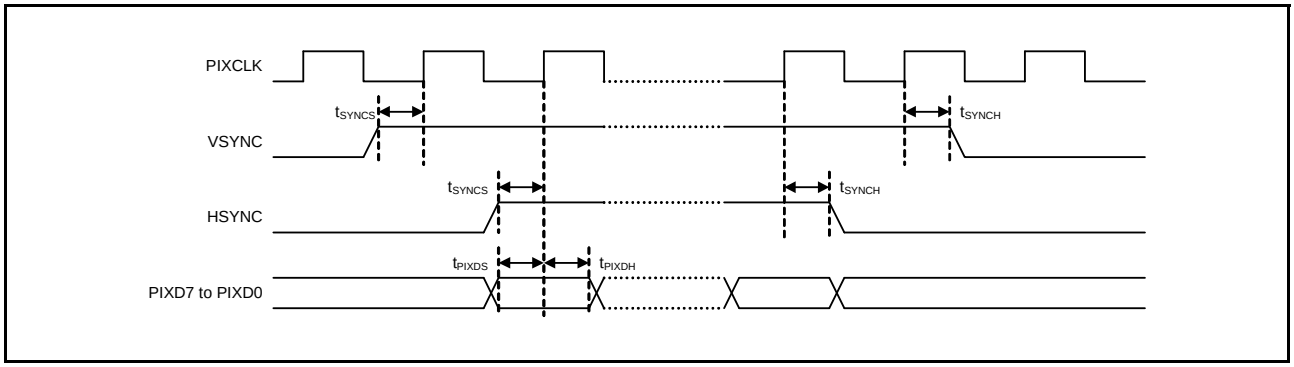


Figure 64.74 PDC AC Timing

64.4 USB Characteristics

Table 64.42 On-Chip USB Low Speed (Host Only) Characteristics (DP and DM Pin Characteristics)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 3.0 to 3.6 V, 3.0 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
USBA_RREF = 2.2 kΩ ±1%, USBMCLK = 20/24 MHz, UCLK = 48 MHz,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input characteristics	Input high level voltage	V_{IH}	2.0	—	—	V	
	Input low level voltage	V_{IL}	—	—	0.8	V	
	Differential input sensitivity	V_{DI}	0.2	—	—	V	| DP – DM |
	Differential common mode range	V_{CM}	0.8	—	2.5	V	
Output characteristics	Output high level voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 µA
	Output low level voltage	V_{OL}	0.0	—	0.3	V	I_{OL} = 2 mA
	Cross-over voltage	V_{CRS}	1.3	—	2.0	V	Figure 64.75
	Rise time	t_{LR}	75	—	300	ns	
	Fall time	t_{LF}	75	—	300	ns	
	Rise/fall time ratio	t_{LR} / t_{LF}	80	—	125	%	t_{LR}/ t_{LF}
Pull-down characteristics	DP/DM pull-down resistance (when the host controller function is selected)	R_{pd}	14.25	—	24.80	kΩ	

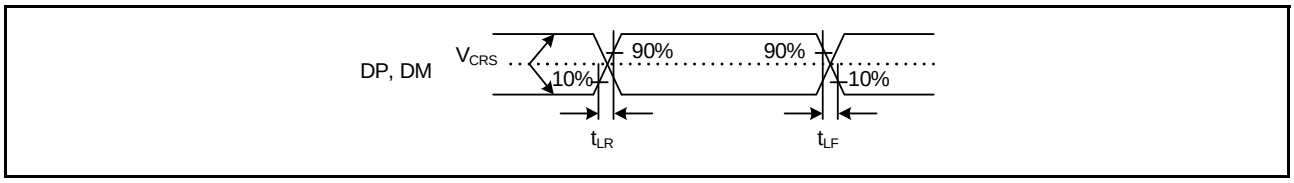


Figure 64.75 DP and DM Output Timing (Low Speed)

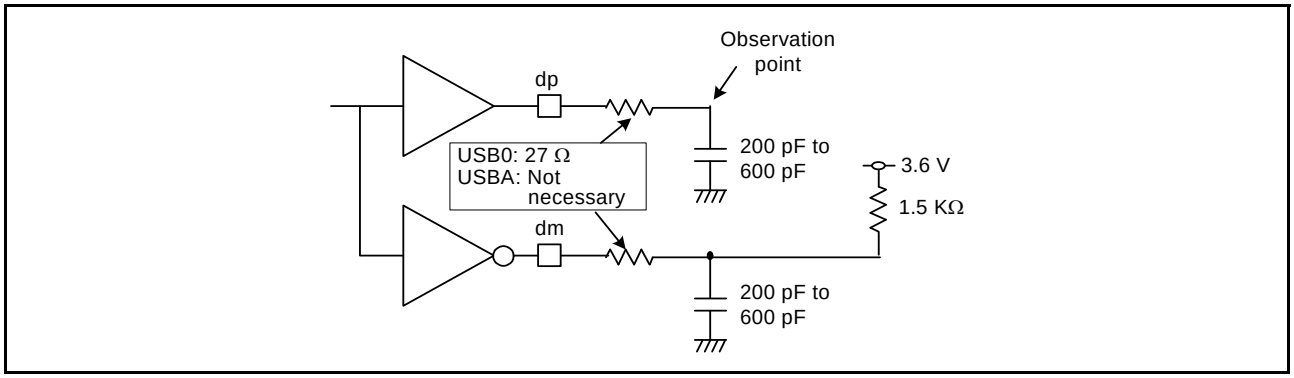


Figure 64.76 Test Circuit (Low Speed)

Table 64.43 On-Chip USB Full-Speed Characteristics (DP and DM Pin Characteristics)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 3.0 to 3.6 V, 3.0 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
USBA_RREF = 2.2 kΩ ±1%, USBMCLK = 20/24 MHz, UCLK = 48 MHz,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input characteristics	Input high level voltage	V_{IH}	2.0	—	—	V	
	Input low level voltage	V_{IL}	—	—	0.8	V	
	Differential input sensitivity	V_{DI}	0.2	—	—	V	| DP – DM |
	Differential common mode range	V_{CM}	0.8	—	2.5	V	
Output characteristics	Output high level voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 µA
	Output low level voltage	V_{OL}	0.0	—	0.3	V	I_{OL} = 2 mA
	Cross-over voltage	V_{CRS}	1.3	—	2.0	V	Figure 64.77
	Rise time	t_{FR}	4	—	20	ns	
	Fall time	t_{FF}	4	—	20	ns	
	Rise/fall time ratio	t_{FR} / t_{FF}	90	—	111.11	%	t_{FR}/ t_{FF}
	Output resistance	Z_{DRV}	28	—	44	Ω	USBFS: Rs = 27 Ω included
			40.5	—	49.5	Ω	USBA: Rs not necessary (PHYSET.REPSEL[1:0] = 01b and PHYSET.HSEB = 0)
Pull-up and pull-down characteristics	DP pull-up resistance (when the function controller function is selected)	R_{pu}	0.900	—	1.575	kΩ	Idle state
			1.425	—	3.090	kΩ	At transmission and reception
	DP/DM pull-down resistance (when the host controller function is selected)	R_{pd}	14.25	—	24.80	kΩ	

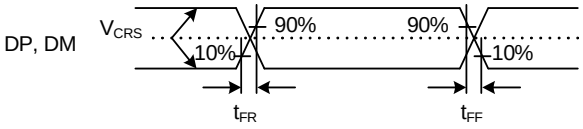


Figure 64.77 DP and DM Output Timing (Full-Speed)

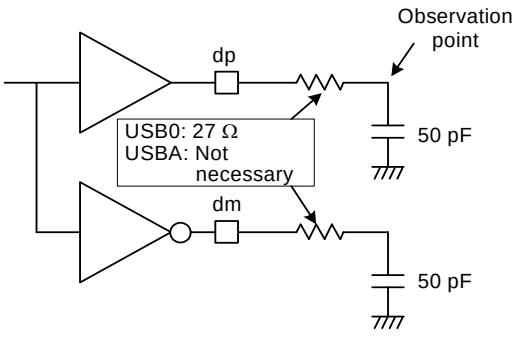


Figure 64.78 Test Circuit (Full-Speed)

Table 64.44 Battery Charge Characteristics (USBA only)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V, USBA_RREF = 2.2 kΩ ±1%, USBMCLK = 20/24 MHz, PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$

Item	Symbol	Min.	Max.	Unit	Test Conditions
D+ sink current	I_{DP_SINK}	25	175	μA	
D- sink current	I_{DM_SINK}	25	175	μA	
DCD source current	I_{DP_SRC}	7	13	μA	
Data detection voltage	V_{DAT_REF}	0.25	0.4	V	
D+ source voltage	V_{DP_SRC}	0.5	0.7	V	Output current = 250 μA
D- source voltage	V_{DM_SRC}	0.5	0.7	V	Output current = 250 μA

64.5 A/D Conversion Characteristics

Table 64.45 12-Bit A/D (Unit 0) Conversion Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKB = PCLKC = 1 MHz to 60 MHz, $T_a = T_{opr}$

Item		Min.	Typ.	Max.	Unit	Test Conditions
Resolution		8	—	12	Bit	
Analog input capacitance		—	—	30	pF	
Channel-dedicated sample-and-hold circuits in use (AN000 to AN002)	Conversion time*1 (Operation at PCLK = 60 MHz) Permissible signal source impedance (max.) = 1.0 kΩ	1.06 (0.40 + 0.25) *2	—	—	µs	• Sampling of channel-dedicated sample-and-hold circuits in 24 states • Sampling in 15 states
	Offset error	—	±1.5	±3.5	LSB	AN000 to AN002 = 0.25 V
	Full-scale error	—	±1.5	±3.5	LSB	AN000 to AN002 = VREFH0 - 0.25 V
	Quantization error	—	±0.5	—	LSB	
	Absolute accuracy	—	±2.5	±5.5	LSB	
	DNL differential nonlinearity error	—	±1.0	±2.0	LSB	
	INL integral nonlinearity error	—	±1.5	±3.0	LSB	
	Holding characteristics of sample-and-hold circuits	—	—	20	µs	
	Dynamic range	0.25	—	VREFH0 – 0.25	V	
Channel-dedicated sample-and-hold circuits not in use (AN000 to AN007)	Conversion time*1 (Operation at PCLK = 60 MHz) Permissible signal source impedance (max.) = 1.0 kΩ	0.48 (0.267) *2	—	—	µs	Sampling in 16 states
	Offset error	—	±1.0	±2.5	LSB	
	Full-scale error	—	±1.0	±2.5	LSB	
	Quantization error	—	±0.5	—	LSB	
	Absolute accuracy	—	±2.0	±4.5	LSB	
	DNL differential nonlinearity error	—	±0.5	±1.5	LSB	
	INL integral nonlinearity error	—	±1.0	±2.5	LSB	

Note: The above specification values apply when there is no access to the external bus during A/D conversion. If access proceeds during A/D conversion, values may not fall within the above ranges.
Note 1. The conversion time includes the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.
Note 2. The value in parentheses indicates the sampling time.

Table 64.46 12-Bit A/D (Unit 1) Conversion Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKB = PCLKD = 1 MHz to 60 MHz, $T_a = T_{opr}$

Item		Min.	Typ.	Max.	Unit	Test Conditions
Resolution		8	—	12	Bit	
Conversion time*1 (Operation at PCLK = 60 MHz)	Permissible signal source impedance (max.) = 1.0 kΩ	0.88 (0.667) *2	—	—	µs	Sampling in 40 states
Analog input capacitance		—	—	30	pF	
Offset error		—	±2.0	±3.5	LSB	
Full-scale error		—	±2.0	±3.5	LSB	
Quantization error		—	±0.5	—	LSB	
Absolute accuracy		—	±4.0	±6.0	LSB	
DNL differential nonlinearity error		—	±1.5	±2.5	LSB	
INL integral nonlinearity error		—	±2.0	±3.5	LSB	

Note: The above specification values apply when there is no access to the external bus during A/D conversion. If access proceeds during A/D conversion, values may not fall within the above ranges.
Note 1. The conversion time includes the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.
Note 2. The value in parentheses indicates the sampling time.

Table 64.47 A/D Internal Reference Voltage Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
PCLKB = PCLKD = 60 MHz, $T_a = T_{opr}$

Item	Min.	Typ.	Max.	Unit	Test Conditions
A/D internal reference voltage	1.20	1.25	1.30	V	

64.6 D/A Conversion Characteristics

Table 64.48 D/A Conversion Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V,
2.7 ≤ VREFH0 ≤ AVCC0, VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item		Min.	Typ.	Max.	Unit	Test Conditions
Resolution		12	12	12	Bit	
Without AMP output	Absolute accuracy	—	—	±6.0	LSB	2-MΩ resistive load 10-bit conversion
	DNL differential nonlinearity error	—	±1.0	±2.0	LSB	2-MΩ resistive load
	RO output resistance	—	7.5	—	kΩ	
	Conversion time	—	—	3.0	µs	20-pF capacitive load
With AMP output	Resistive load	5	—	—	kΩ	
	Capacitive load	—		50	pF	
	Output voltage range	0.2	—	AVCC1 – 0.2	V	
	DNL differential nonlinearity error	—	±1.0	±2.0	LSB	
	INL integral nonlinearity error	—	±2.0	±4.0	LSB	
	Conversion time	—	—	4.0	µs	

64.7 Temperature Sensor Characteristics

Table 64.49 Temperature Sensor Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item	Min.	Typ.	Max.	Unit	Test Conditions
Relative accuracy	—	±1	—	°C	
Temperature slope	—	3.8	—	mV/°C	
Output voltage (at 25°C)	—	1.21	—	V	
Temperature sensor start time	—	—	30	µs	
Sampling time*1	4.15	—	—	µs	

Note 1. Set the S12AD1.ADSSTRT register such that the sampling time of the 12-bit A/D converter satisfies this specification.

64.8 Power-on Reset Circuit and Voltage Detection Circuit Characteristics

Table 64.50 Power-on Reset Circuit and Voltage Detection Circuit Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Voltage detection level	Power-on reset (POR)	Low power consumption function disabled*1	V_{POR}	2.5	2.6	2.7	V	Figure 64.79
		Low power consumption function enabled*2		2.0	2.35	2.7		
	Voltage detection circuit (LVD0)		V_{det0_1}	2.84	2.94	3.04		Figure 64.80
			V_{det0_2}	2.77	2.87	2.97		
			V_{det0_3}	2.70	2.80	2.90		
	Voltage detection circuit (LVD1)		V_{det1_1}	2.89	2.99	3.09		Figure 64.81
			V_{det1_2}	2.82	2.92	3.02		
			V_{det1_3}	2.75	2.85	2.95		
	Voltage detection circuit (LVD2)		V_{det2_1}	2.89	2.99	3.09		Figure 64.82
			V_{det2_2}	2.82	2.92	3.02		
			V_{det2_3}	2.75	2.85	2.95		
Internal reset time	Power-on reset time		t_{POR}	—	4.6	—	ms	Figure 64.79
	LVD0 reset time		t_{LVD0}	—	0.70	—		Figure 64.80
	LVD1 reset time		t_{LVD1}	—	0.57	—		Figure 64.81
	LVD2 reset time		t_{LVD2}	—	0.57	—		Figure 64.82
Minimum VCC down time			t_{VOFF}	200	—	—	μs	Figure 64.79, Figure 64.80
Response delay time			t_{det}	—	—	200	μs	Figure 64.79 to Figure 64.82
LVD operation stabilization time (after LVD is enabled)*3			$T_{d(E-A)}$	—	—	10	μs	Figure 64.81, Figure 64.82
Hysteresis width (LVD1 and LVD2)			V_{LVH}	—	80	—	mV	

Note: The minimum VCC down time indicates the time when VCC is below the minimum value of voltage detection levels V_{POR}, V_{det1}, and V_{det2} for the POR/ LVD.

Note 1. The low power consumption function is disabled and DEEPCUT[1:0] = 00b or 01b.

Note 2. The low power consumption function is enabled and DEEPCUT[1:0] = 11b.

Note 3. The voltage of VCC = AVCC0 = AVCC1 when LVD1 is enabled must be set to at least 80 mV above the maximum value of the voltage detection 1 level ($V_{det1\_1, 2, 3}$) selected by the LVDLVLR.LVD1LVL[3:0] bits.
Similarly, the voltage of VCC = AVCC0 = AVCC1 when LVD2 is enabled must be set to at least 80 mV above the maximum value of the voltage detection 2 level ($V_{det2\_1, 2, 3}$) selected by the LVDLVLR.LVD2LVL[3:0] bits.

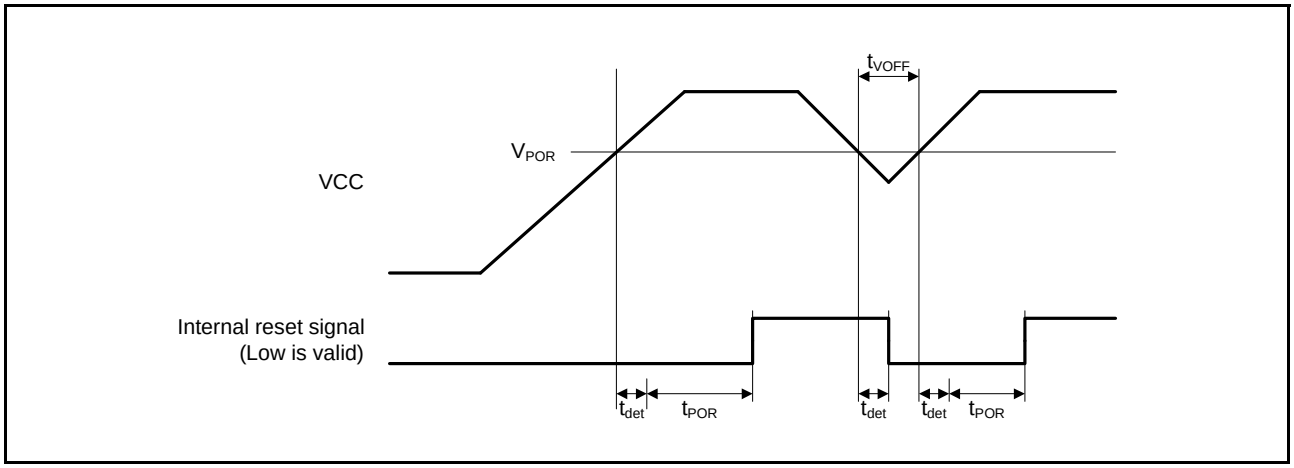


Figure 64.79 Power-on Reset Timing

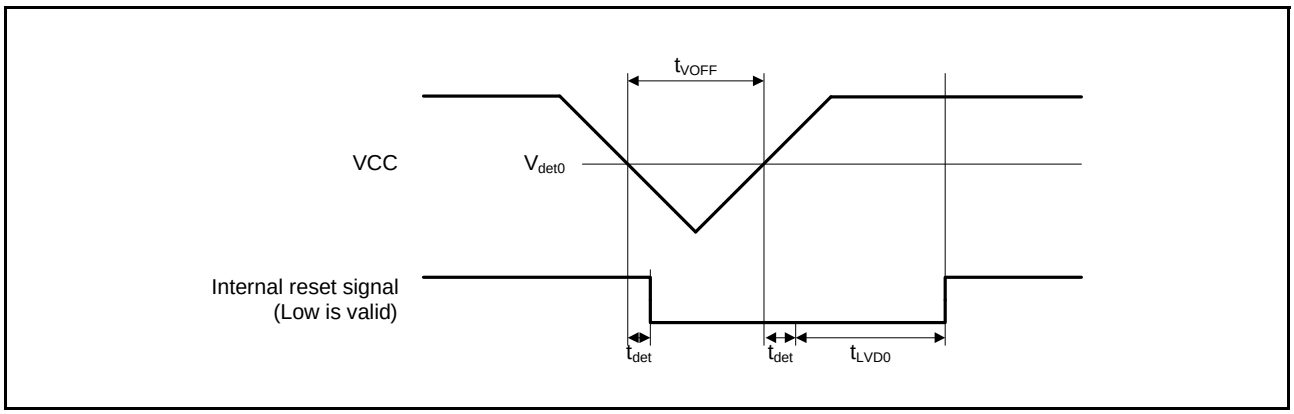


Figure 64.80 Voltage Detection Circuit Timing (V_{det0})

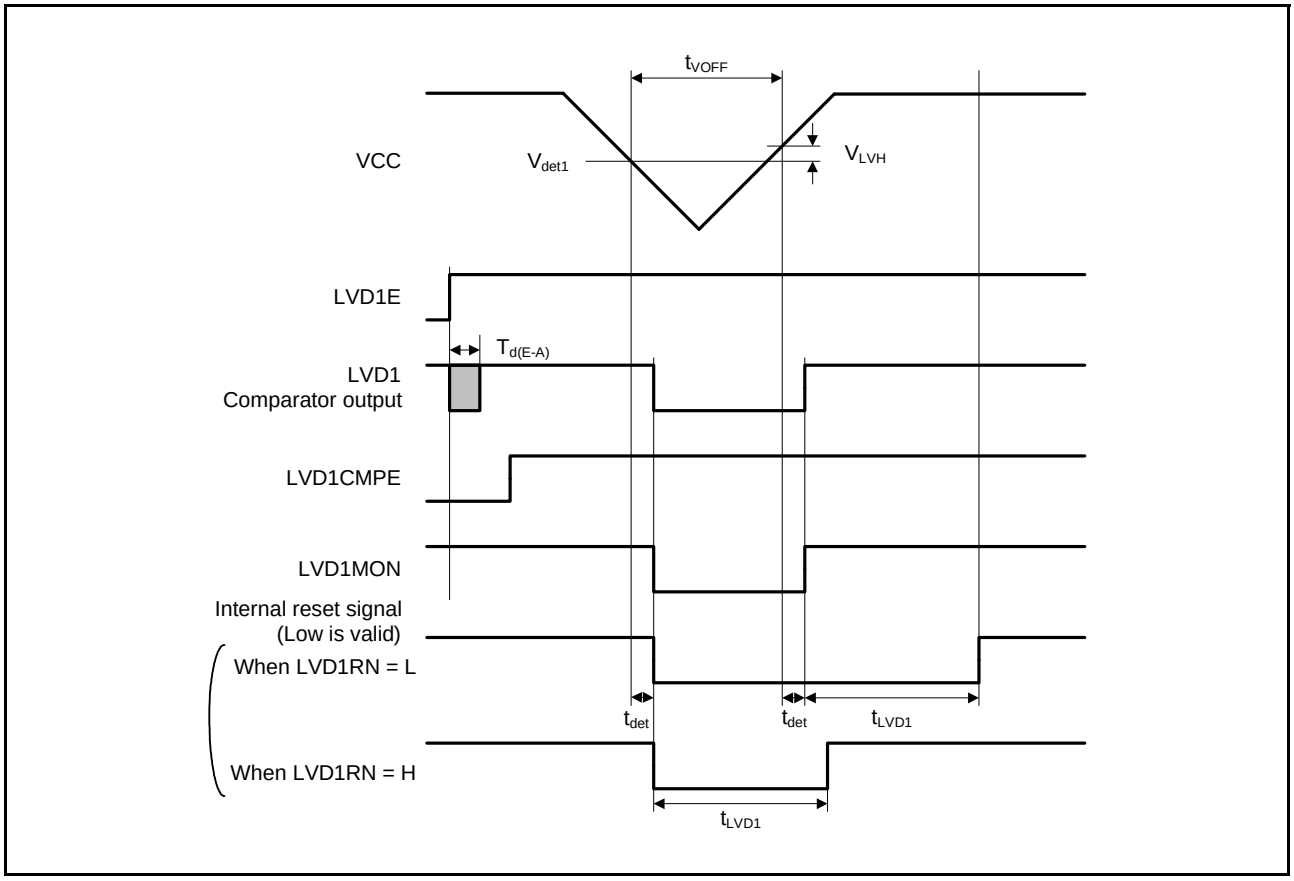


Figure 64.81 Voltage Detection Circuit Timing (V_{det1})

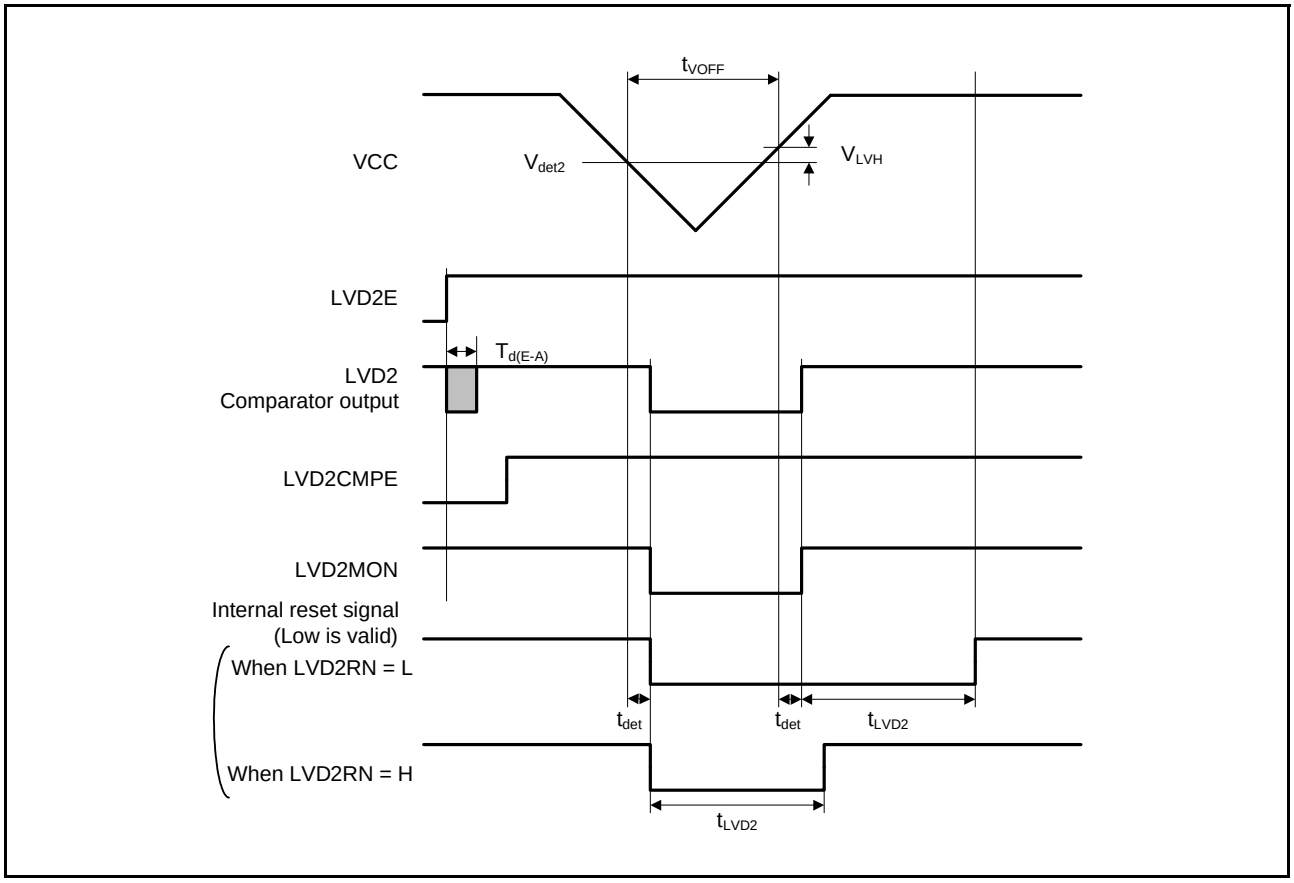


Figure 64.82 Voltage Detection Circuit Timing (V_{det2})

64.9 Oscillation Stop Detection Timing

Table 64.51 Oscillation Stop Detection Circuit Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Detection time	t_{dr}	—	—	1	ms	Figure 64.83

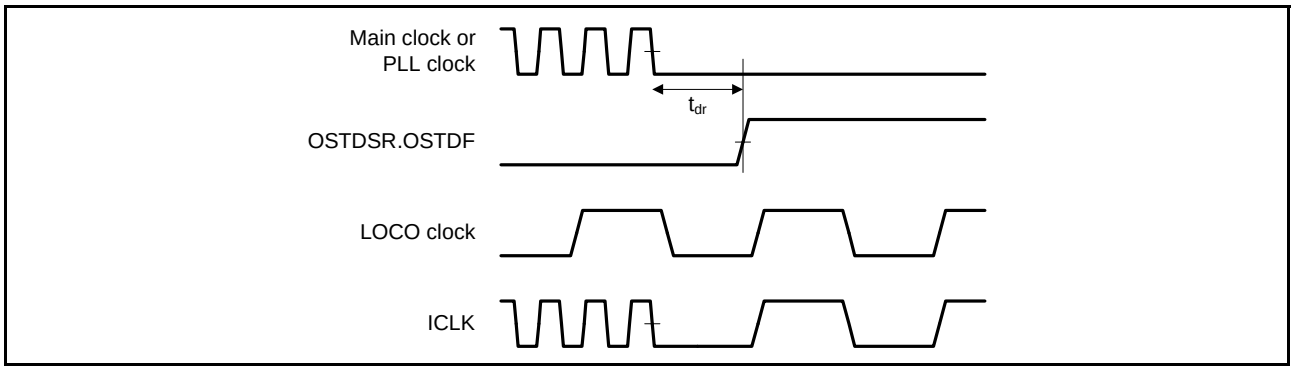


Figure 64.83 Oscillation Stop Detection Timing

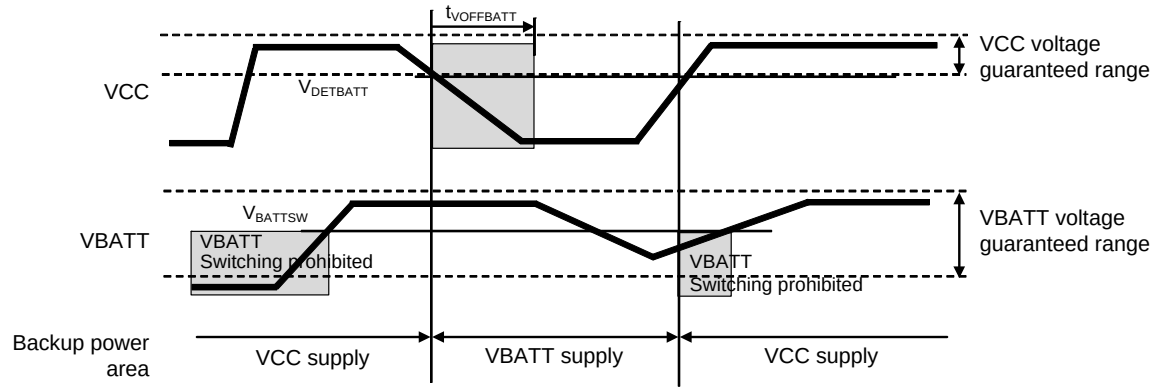
64.10 Battery Backup Function Characteristics

Table 64.52 Battery Backup Function Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
V_{BATT} = 2.0 to 3.6 V, $T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Voltage level for switching to battery backup	$V_{DETBATT}$	2.50	2.60	2.70	V	Figure 64.84
Lower-limit V_{BATT} voltage for power supply switching due to VCC voltage drop	V_{BATTSW}	2.70	—	—		
VCC-off period for starting power supply switching	$t_{VOFFBATT}$	200	—	—	µs	

Note: The VCC-off period for starting power supply switching indicates the period in which VCC is below the minimum value of the voltage level for switching to battery backup ($V_{DETBATT}$).



Note. The VBATT voltage when the supplied power source switches from Vcc to V_{BATT} should not be lower than V_{BATTSW}, the lower-limit VBATT voltage for switching between power supplies due to a drop in the VCC voltage.

Figure 64.84 Battery Backup Function Characteristics

64.11 Flash Memory Characteristics

Table 64.53 Code Flash Memory Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V
Temperature range for programming/erasure: $T_a = T_{opr}$

Item		Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time N_{PEC} ≤ 100 times	256 bytes	t_{P256}	—	0.9	13.2	—	0.4	6	ms
	8 Kbytes	t_{P8K}	—	29	176	—	13	80	ms
	32 Kbytes	t_{P32K}	—	116	704	—	52	320	ms
Programming time N_{PEC} > 100 times	256 bytes	t_{P256}	—	1.1	15.8	—	0.5	7.2	ms
	8 Kbytes	t_{P8K}	—	35	212	—	16	96	ms
	32 Kbytes	t_{P32K}	—	140	848	—	64	384	ms
Erasure time N_{PEC} ≤ 100 times	8 Kbytes	t_{E8K}	—	71	216	—	39	120	ms
	32 Kbytes	t_{E32K}	—	254	864	—	141	480	ms
Erasure time N_{PEC} > 100 times	8 Kbytes	t_{E8K}	—	85	260	—	47	144	ms
	32 Kbytes	t_{E32K}	—	304	1040	—	169	576	ms
Reprogramming/erasure cycle*1		N_{PEC}	1000*2	—	—	1000*2	—	—	Times
Suspend delay time during programming		t_{SPD}	—	—	264	—	—	120	µs
First suspend delay time during erasing (in suspend priority mode)		t_{SESD1}	—	—	216	—	—	120	µs
Second suspend delay time during erasure (in suspend priority mode)		t_{SESD2}	—	—	1.7	—	—	1.7	ms
Suspend delay time during erasure (in erasure priority mode)		t_{SEED}	—	—	1.7	—	—	1.7	ms
Forced stop command		t_{FD}	—	—	32	—	—	20	µs
Data hold time*3		t_{DRP}	10	—	—	10	—	—	Year
FCU reset time		t_{FCUR}	35	—	—	35	—	—	µs

Note 1. Definition of reprogram/erase cycle:
The reprogram/erase cycle is the number of erasing for each block. When the reprogram/erase cycle is n times (n = 1000), erasing can be performed n times for each block. For instance, when 256-byte programming is performed 32 times for different addresses in 8-Kbyte block and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address for several times as one erasing is not enabled (overwriting is prohibited).

Note 2. This is the minimum number of times to guarantee all the characteristics after reprogramming (guaranteed range is from 1 to the value of the minimum value).

Note 3. This shows the characteristics when reprogramming is performed within the specified range, including the minimum value.

Table 64.54 Data Flash Memory Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
Temperature range for programming/erasure: $T_a = T_{opr}$

Item		Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	4 bytes	t_{DP4}	—	0.36	3.8	—	0.16	1.7	ms
Erasure time	64 bytes	t_{DE64}	—	3.1	18	—	1.7	10	ms
Blank check time	4 bytes	t_{DBC4}	—	—	84	—	—	30	µs
	64 bytes	t_{DBC64}	—	—	280	—	—	100	µs
	2 Kbytes	t_{DBC2K}	—	—	6169	—	—	2200	µs
Reprogramming/erasure cycle*1		N_{DPEC}	100000 *2	—	—	100000 *2	—	—	—
Suspend delay time during programming		t_{DSPD}	—	—	264	—	—	120	µs
First suspend delay time during erasure (in suspend priority mode)		t_{DSESD1}	—	—	216	—	—	120	µs
Second suspend delay time during erasure (in suspend priority mode)		t_{DSESD2}	—	—	300	—	—	300	µs
Suspend delay time during erasing (in erasure priority mode)		t_{DSEED}	—	—	300	—	—	300	µs
Forced stop command		t_{FD}	—	—	32	—	—	20	µs
Data hold time*3		t_{DDRP}	10	—	—	10	—	—	—

Note 1. Definition of reprogram/erase cycle:
The reprogram/erase cycle is the number of erasing for each block. When the reprogram/erase cycle is n times (n = 100000), erasing can be performed n times for each block. For instance, when 4-byte programming is performed 512 times for different addresses in 2-Kbyte block and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address for several times as one erasing is not enabled (overwriting is prohibited).

Note 2. This is the minimum number of times to guarantee all the characteristics after reprogramming (guaranteed range is from 1 to the value of the minimum value).

Note 3. This shows the characteristics when reprogramming is performed within the specified range, including the minimum value.

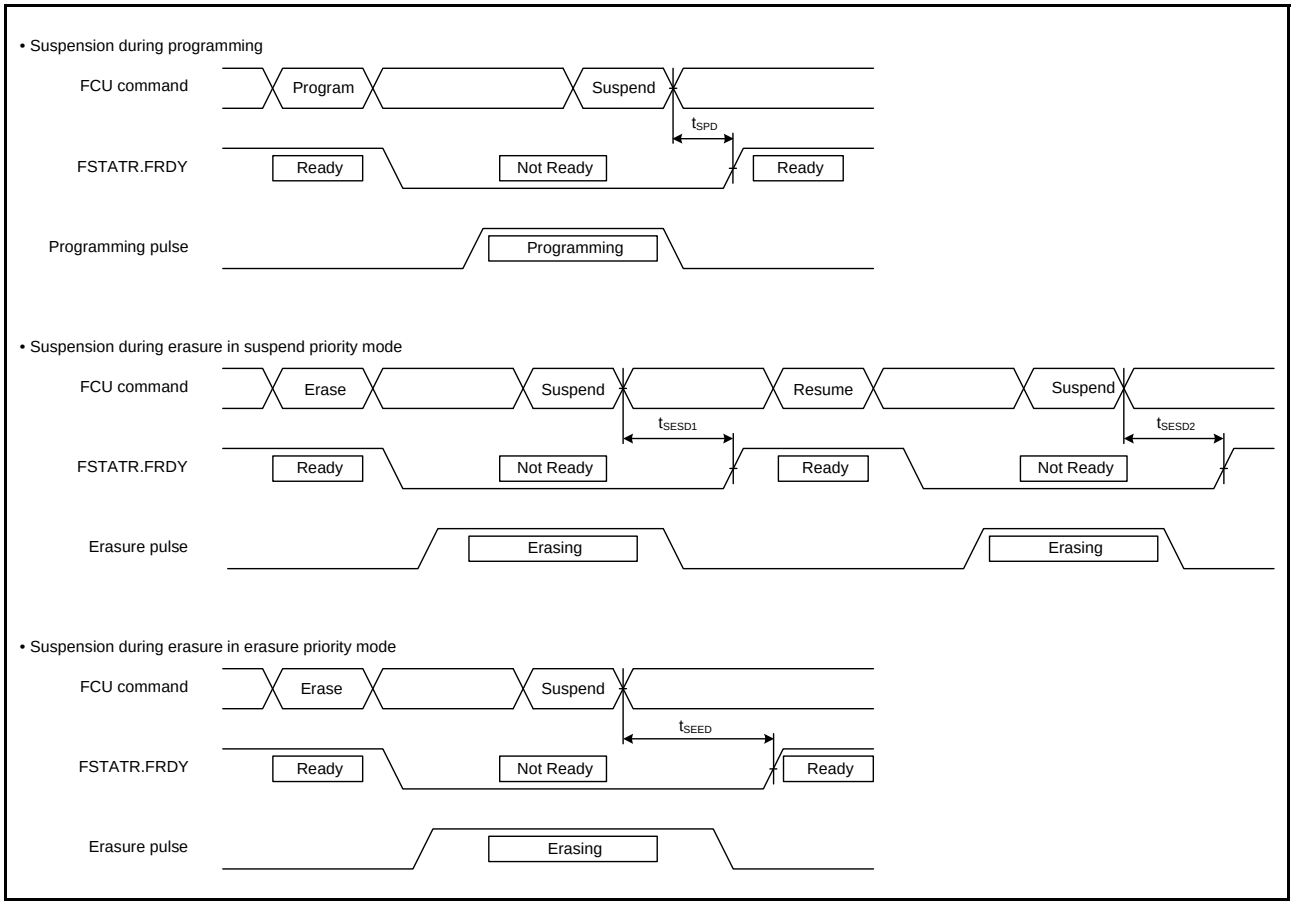


Figure 64.85 Flash Memory Programming/Erasure Suspension Timing

64.12 Boundary Scan

Table 64.55 Boundary Scan Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0,
VCC_USBA = AVCC_USBA = 3.0 to 3.6 V,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = VSS1_USBA = VSS2_USBA = PVSS_USBA = AVSS_USBA = 0 V,
$T_a = T_{opr}$
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF
High-drive output is selected by the driving ability control register.

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
TCK clock cycle time	t_{TCKcyc}	100	—	—	ns	Figure 64.86
TCK clock high pulse width	t_{TCKH}	45	—	—	ns	
TCK clock low pulse width	t_{TCKL}	45	—	—	ns	
TCK clock rise time	t_{TCKr}	—	—	5	ns	
TCK clock fall time	t_{TCKf}	—	—	5	ns	
TRST# pulse width	t_{TRSTW}	20	—	—	t_{TCKcyc}	Figure 64.87
TMS setup time	t_{TMSS}	20	—	—	ns	Figure 64.88
TMS hold time	t_{TMSH}	20	—	—	ns	
TDI setup time	t_{TDIS}	20	—	—	ns	
TDI hold time	t_{TDIH}	20	—	—	ns	
TDO data delay time	t_{TDOD}	—	—	40	ns	

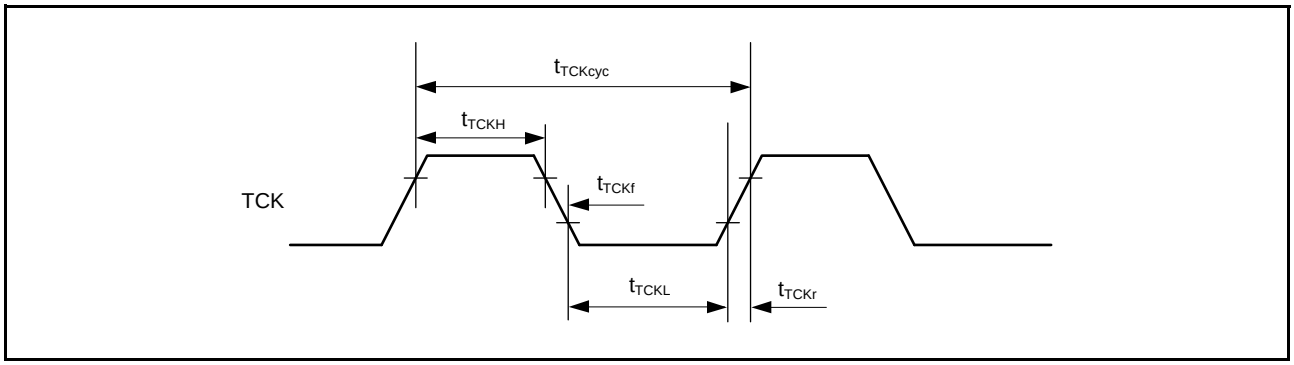


Figure 64.86 Boundary Scan TCK Timing

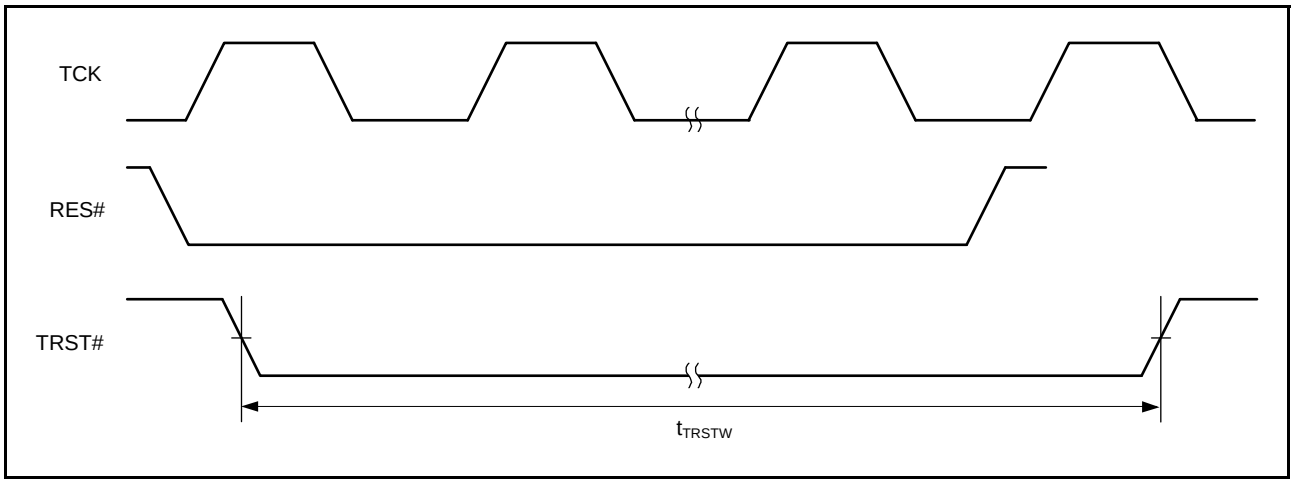


Figure 64.87 Boundary Scan TRST# Timing

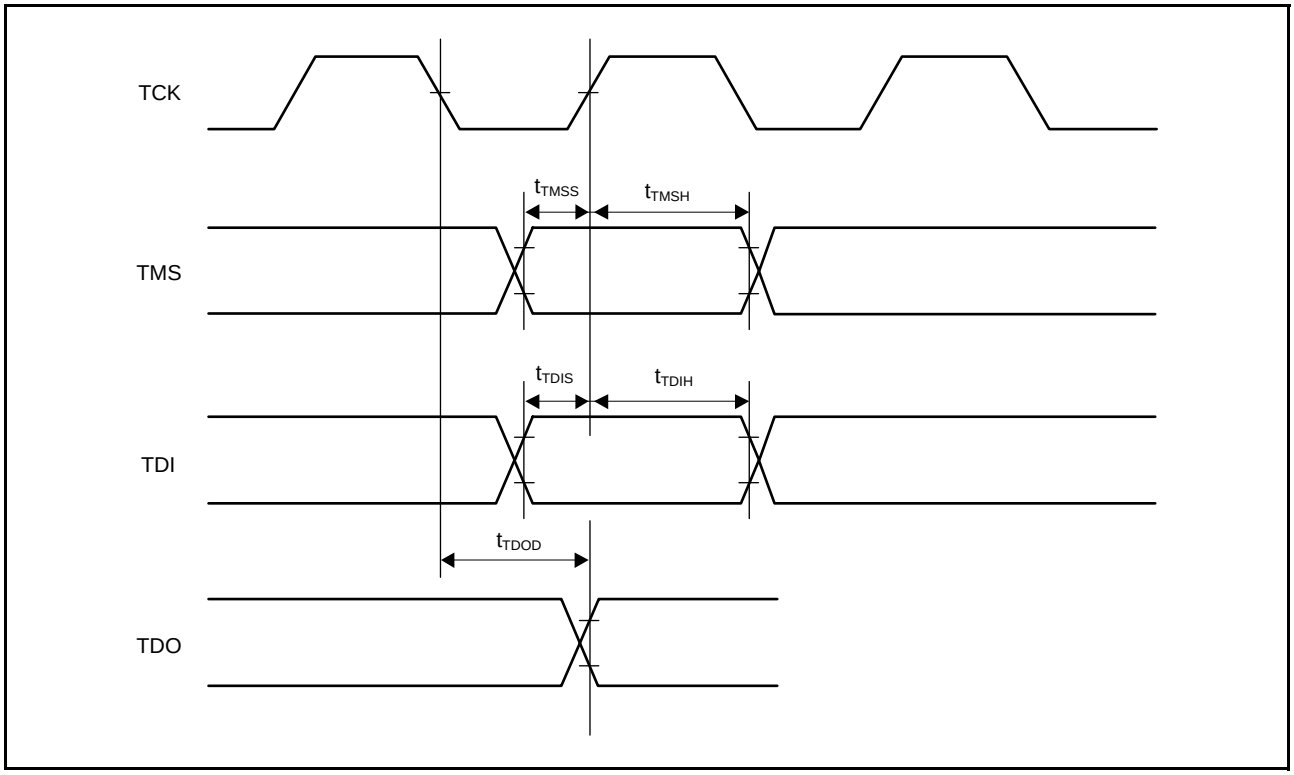


Figure 64.88 Boundary Scan Input/Output Timing

Appendix 1. Port States in Each Processing Mode

Table 1.1 Port States in Each Processing State (1 / 6)

Port Name Pin Name	Operating Mode According to Registers Setting		Reset	Software Standby Mode		Deep Software Standby Mode	After Deep Software Standby Mode is Canceled (Return to Start-up Mode)	
				OPE = 1	OPE = 0	IOKEEP = 1/0	IOKEEP = 1*1	IOKEEP = 0
P00/IRQ8, P01/IRQ9, P02/IRQ10	All		Hi-Z	Keep-O *2		Keep	Keep	Hi-Z
P03/DA0/ IRQ11	All	DA0 output (DAOE0 = 1)	Hi-Z	DA output retained		Hi-Z	Hi-Z	Hi-Z
		Other than the above (DAOE0 = 0)		Keep-O *2		Keep	Keep	
P05/DA1/ IRQ13	All	DA1 output (DAOE1 = 1)	Hi-Z	DA output retained		Hi-Z	Hi-Z	Hi-Z
		Other than the above (DAOE1 = 0)		Keep-O*2		Keep	Keep	
P07/IRQ15	All		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P10/IRQ0/ ALE/USBA_ OVRCURA	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O *2		Keep-O *3	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)			[ALE output] L [Other than the above] Keep-O *2	[ALE output] Hi-Z [Other than the above] Keep-O *2			
P11/IRQ1/ USBA_ VBUS	All		Hi-Z	Keep-O *2		Keep-O *3	Keep	Hi-Z
P12/IRQ2/ WR3#/BC3#	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O *2		Keep-O	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)			[WR3#/BC3# output] H [Other than the above] Keep-O *2	[WR3#/BC3# output] Hi-Z [Other than the above] Keep-O *2			
P13/IRQ3/ WR2#/BC2#	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O *2		Keep-O	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)			[WR2#/BC2# output] H [Other than the above] Keep-O *2	[WR2#/BC2# output] Hi-Z [Other than the above] Keep-O *2			
P14/IRQ4/ USB0_ OVRCURA, P15/IRQ5/ CRX1, P16/IRQ6/ SCL2/USB0_ VBUS/USB0_ OVRCURB, P17/IRQ7/ SDA2	All		Hi-Z	Keep-O *2		Keep-O*3	Keep	Hi-Z
P20/IRQ8, P21/IRQ9 P22/USB0_ OVRCURB/ USBA_ OVRCURB, P23	All		Hi-Z	Keep-O*2		Keep-O *3	Keep	Hi-Z

Table 1.1 Port States in Each Processing State (2 / 6)

Port Name Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode		Deep Software Standby Mode	After Deep Software Standby Mode is Canceled (Return to Start-up Mode)	
			OPE = 1	OPE = 0	IOKEEP = 1/0	IOKEEP = 1*1	IOKEEP = 0
P24/CS4#, P25/CS5#, P26/CS6#, P27/CS7#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CSn# output] H [Other than the above] Keep-O	[CSn# output] Hi-Z [Other than the above] Keep-O			
P30/ IRQ0-DS/ RTCIC0, P31/ IRQ1-DS/ RTCIC1, P32/ IRQ2-DS/ RTCIC2, P33/ IRQ3-DS, P34/ IRQ4-DS, P35/NMI	All	Hi-Z	Keep-O*2		Keep-O*3	Keep	Hi-Z
P36, P37	All	Hi-Z	Keep-O		Keep	Keep	Hi-Z
P40/ IRQ8-DS, P41/ IRQ9-DS, P42/ IRQ10-DS, P43/ IRQ11-DS, P44/ IRQ12-DS, P45/ IRQ13-DS, P46/ IRQ14-DS, P47/ IRQ15-DS	All	Hi-Z	Keep-O*2		Keep-O*3	Keep	Hi-Z
P50/WR0#/ WR#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[WR0#/WR# output] H [Other than the above] Keep-O	[WR0#/WR# output] Hi-Z [Other than the above] Keep-O			
P51/WR1#/ BC1#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[WR1#/BC1# output] H [Other than the above] Keep-O	[WR1#/BC1# output] Hi-Z [Other than the above] Keep-O			
P52/RD#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[RD# output] H [Other than the above] Keep-O	[RD# output] Hi-Z [Other than the above] Keep-O			
P53/BCLK	All	Hi-Z	[Clock output] H [Other than the above] Keep-O		Keep	Keep	Hi-Z
P54/ALE/ TRDATA2	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[ALE output] L [Other than the above] Keep-O	[ALE output] Hi-Z [Other than the above] Keep-O			
P55/ TRDATA3/ IRQ10	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P56	All	Hi-Z	Keep-O		Keep	Keep	Hi-Z

Table 1.1 Port States in Each Processing State (3 / 6)

Port Name Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode		Deep Software Standby Mode	After Deep Software Standby Mode is Canceled (Return to Start-up Mode)	
			OPE = 1	OPE = 0	IOKEEP = 1/0	IOKEEP = 1*1	IOKEEP = 0
P60/CS0#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CS0# output] H [Other than the above] Keep-O	[CS0# output] Hi-Z [Other than the above] Keep-O			
P61/CS1#/ SDCS#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CS1# output] H [Other than the above] Keep-O	[CS1# output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh disabled (SDSELF.SFEN = 0)		[SDCS# output] H [Other than the above] Keep-O	[SDCS# output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh enabled (SDSELF.SFEN = 1)		[SDCS# output] L [Other than the above] Keep-O				
P62/CS2#/ RAS#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CS2# output] H [Other than the above] Keep-O	[CS2# output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh disabled (SDSELF.SFEN = 0)		[RAS# output] H [Other than the above] Keep-O	[RAS# output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh enabled (SDSELF.SFEN = 1)		[RAS# output] L [Other than the above] Keep-O				
P63/CS3#/ CAS#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CS3# output] H [Other than the above] Keep-O	[CS3# output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh disabled (SDSELF.SFEN = 0)		[CAS# output] H [Other than the above] Keep-O	[CAS# output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh enabled (SDSELF.SFEN = 1)		[CAS# output] L [Other than the above] Keep-O				
P64/CS4#/ WE#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CS4# output] H [Other than the above] Keep-O	[CS4# output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh disabled (SDSELF.SFEN = 0)		[WE# output] H [Other than the above] Keep-O	[WE# output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh enabled (SDSELF.SFEN = 1)						

Table 1.1 Port States in Each Processing State (4 / 6)

Port Name Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode		Deep Software Standby Mode	After Deep Software Standby Mode is Canceled (Return to Start-up Mode)	
			OPE = 1	OPE = 0	IOKEEP = 1/0	IOKEEP = 1*1	IOKEEP = 0
P65/CS5#/ CKE	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CS5# output] H [Other than the above] Keep-O	[CS5# output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh disabled (SDSELF.SFEN = 0)		[CKEoutput] H [Other than the above] Keep-O	[CKEoutput] Hi-Z [Other than the above] Keep-O			
	Self-Refresh enabled (SDSELF.SFEN = 1)		[CKEoutput] L [Other than the above] Keep-O				
P66/CS6#/ DQM0	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CS6# output] H [Other than the above] Keep-O	[CS6# output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh disabled (SDSELF.SFEN = 0)		[DQM0output] DQM0 output retained [Other than the above] Keep-O	[DQM0output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh enabled (SDSELF.SFEN = 1)						
P67/CS7#/ IRQ15/DQM1	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CS7# output] H [Other than the above] Keep-O *2	[CS7# output] Hi-Z [Other than the above] Keep-O*2			
	Self-Refresh disabled (SDSELF.SFEN = 0)		[DQM1output] DQM1 output retained [Other than the above] Keep-O *2	[DQM1output] Hi-Z [Other than the above] Keep-O*2			
	Self-Refresh enabled (SDSELF.SFEN = 1)						
P70/SDCLK	All	Hi-Z	[Clock output] H [Other than the above] Keep-O		Keep	Keep	Hi-Z
P71/CS1#, P72/CS2#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CSn# output] H [Address output] Address output retained [Other than the above] Keep-O	[CSn# output] Hi-Z [Address output] Hi-Z [Other than the above] Keep-O	Keep	Keep	Hi-Z
P73/CS3#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CSn# output] H [Other than the above] Keep-O	[CSn# output] Hi-Z [Other than the above] Keep-O			

Table 1.1 Port States in Each Processing State (5 / 6)

Port Name Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode		Deep Software Standby Mode	After Deep Software Standby Mode is Canceled (Return to Start-up Mode)	
			OPE = 1	OPE = 0	IOKEEP = 1/0	IOKEEP = 1*1	IOKEEP = 0
P74/CS4#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CSn# output] H [Address output] Address output retained [Other than the above] Keep-O	[CSn# output] Hi-Z [Address output] Hi-Z [Other than the above] Keep-O			
P75/CS5#, P76/CS6#, P77/CS7#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CSn# output] H [Other than the above] Keep-O	[CSn# output] Hi-Z [Other than the above] Keep-O			
P80/ TRDATA0, P81/ TRDATA1, P82,P83, P86,P87	All	Hi-Z	Keep-O		Keep	Keep	Hi-Z
Port 9	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[Address output] Address output retained [Data output] Hi-Z [Other than the above] Keep-O	[Address output] Hi-Z [Data output] Hi-Z [Other than the above] Keep-O			
PA0/DQM2	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[Address output] Address output retained [Other than the above] Keep-O	[Address output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh disabled (SDSELF.SFEN = 0)		[DQM2 output] DQM2 output retained [Other than the above] Keep-O	[DQM2 output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh enabled (SDSELF.SFEN = 1)						
PA1/IRQ11/ DQM3	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[Address output] Address output retained [Other than the above] Keep-O*2	[Address output] Hi-Z [Other than the above] Keep-O*2			
	Self-Refresh disabled (SDSELF.SFEN = 0)		[DQM3 output] DQM3 output retained [Other than the above] Keep-O*2	[DQM3 output] Hi-Z [Other than the above] Keep-O*2			
	Self-Refresh enabled (SDSELF.SFEN = 1)						
PA2, PA3/ IRQ6-DS, PA4/IRQ5, PA5 to PA7	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O *2		Keep-O*3	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[Address output] Address output retained [Other than the above] Keep-O *2	[Address output] Hi-Z [Other than the above] Keep-O *2			

Table 1.1 Port States in Each Processing State (6 / 6)

Port Name Pin Name	Operating Mode According to Registers Setting		Reset	Software Standby Mode		Deep Software Standby Mode	After Deep Software Standby Mode is Canceled (Return to Start-up Mode)	
				OPE = 1	OPE = 0	IOKEEP = 1/0	IOKEEP = 1*1	IOKEEP = 0
PB0/IRQ12, PB1/IRQ4-DS, PB2 to PB7	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O *2		Keep-O*3	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)			[Address output] Address output retained [Other than the above] Keep-O *2	[Address output] Hi-Z [Other than the above] Keep-O*2			
PC0/IRQ14, PC1/IRQ12, PC2, PC3, PC4/CS3#, PC5/CS2#, PC6/IRQ13/CS1#, PC7/IRQ14/CS0#	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O *2		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)			[Address output] Address output retained [CSn# output] H [Other than the above] Keep-O *2	[Address output] Hi-Z [CSn# output] Hi-Z [Other than the above] Keep-O *2			
PD0/IRQ0, PD1/IRQ1, PD2/IRQ2, PD3/IRQ3, PD4/IRQ4, PD5/IRQ5, PD6/IRQ6, PD7/IRQ7	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)			[Data output] Hi-Z				
PE0, PE1, PE2/IRQ7-DS, PE3, PE4, PE5/IRQ5, PE6/IRQ6, PE7/IRQ7	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O *2		Keep-O *3	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)	8 bits in width of bus		Keep-O*2				
		16 bits in width of bus		[Data output] Hi-Z				
PF0 to PF4, PF5/IRQ4	All		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
PG0, PG1, PG2/TRDATA0, PG3/TRDATA1, PG4, PG5, PG6/TRDATA2, PG7/TRDATA3	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)			[Data output] Hi-Z [Other than the above] Keep-O				
Port J	All		Hi-Z	Keep-O		Keep	Keep	Hi-Z
USB0_DM	All		Hi-Z	Keep-O *4		Hi-Z*3	Hi-Z	
USB0_DP	All		Hi-Z	Keep-O *4		Hi-Z*3	Hi-Z	
USBA_DM	All		Hi-Z	Keep-O *4		Hi-Z*5	Hi-Z	
USBA_DP	All		Hi-Z	Keep-O *4		Hi-Z*5	Hi-Z	

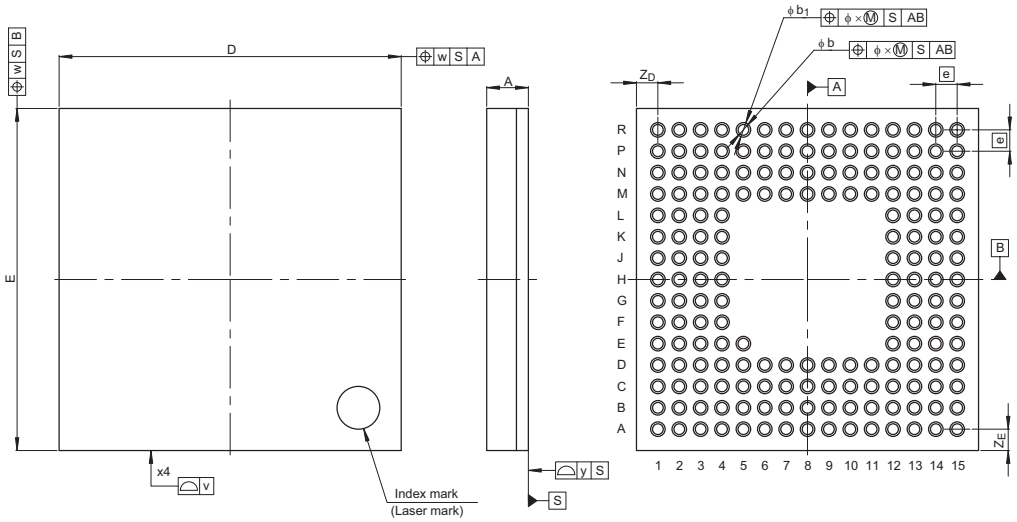
H: High-level
L: Low-level
Keep-O: Output pins retain their previous values, and input pins become high-impedance.
Keep: Pin states are retained during periods on software standby.
Hi-Z: High-impedance

Note 1. Retains the I/O port state until the DPSBYCR.IOKEEP bit is cleared to 0.
Note 2. Input is enabled if the pin is specified as the software standby cancelling source while it is used as an external interrupt pin.
Note 3. Input is enabled if the pin is specified as the deep software standby cancelling source.
Note 4. Input is enabled while the pin is used as an input pin.
Note 5. For the host operation, set the DPUSR0R.DRPD0 bit to 1 to enable the DP and DM pull-down resistors.
For the function operation, set the DPUSR0R.RPUE0 bit to 1 to enable the DP pull-up resistor.

Appendix 2. Package Dimensions

Information on the latest version of the package dimensions or mountings has been displayed in “Packages” on Renesas Electronics Corporation website.

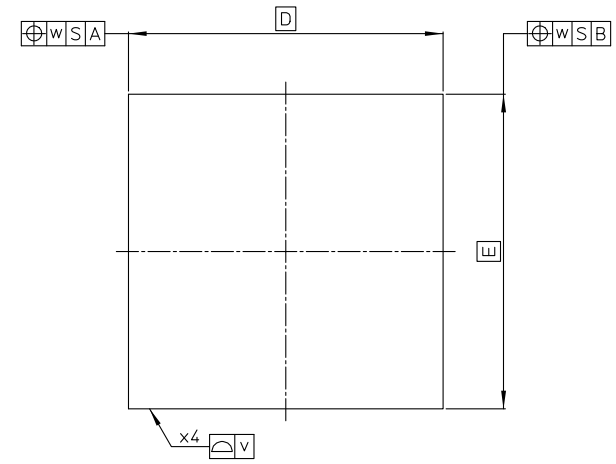
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-TFLGA177-8x8-0.50	PTLG0177KA-A	177F0E-A	0.2g

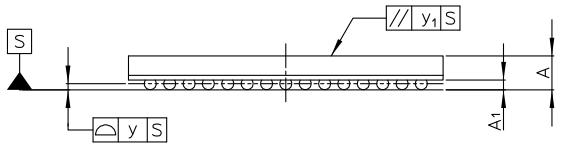


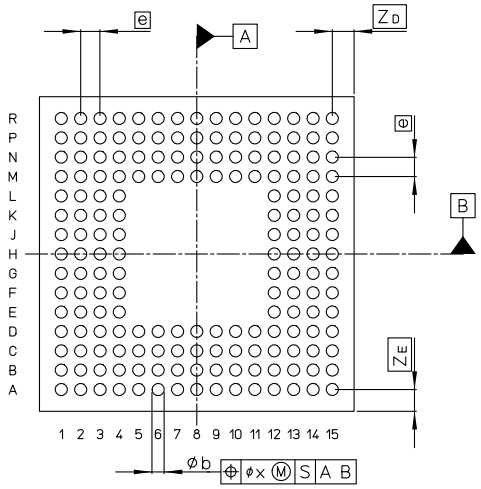
Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	8.0	—
E	—	8.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.05
e	—	0.5	—
b	0.21	0.25	0.29
b_1	0.29	0.34	0.39
x	—	—	0.08
y	—	—	0.08
Z_D	—	0.5	—
Z_E	—	0.5	—

Figure A 177-Pin TFLGA (PTLG0177KA-A)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFBGA176-13x13-0.80	PLBG0176GA-A	BP-176/BP-176V	0.45g



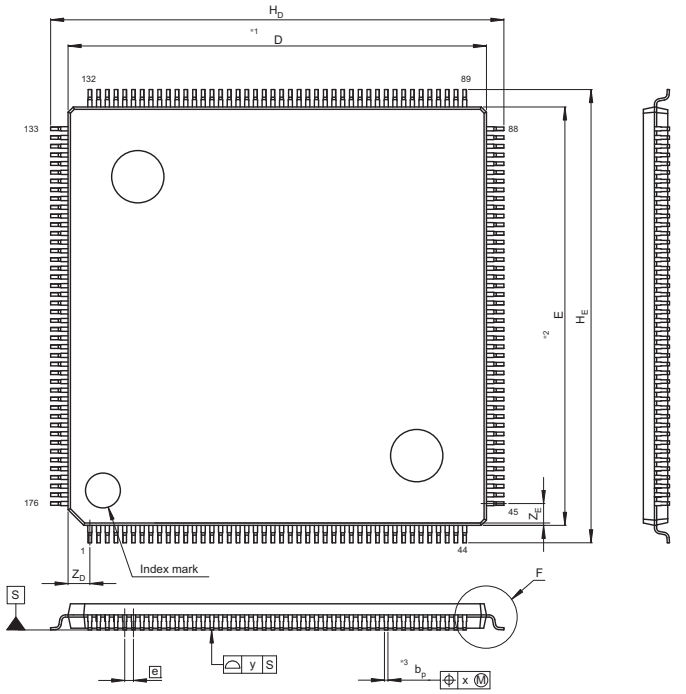


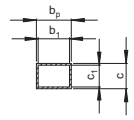


Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	13.0	—
E	—	13.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.40
A_1	0.35	0.40	0.45
e	—	0.80	—
b	0.45	0.50	0.55
x	—	—	0.08
y	—	—	0.10
y_1	—	—	0.2
S_D	—	—	—
S_E	—	—	—
Z_D	—	0.90	—
Z_E	—	0.90	—

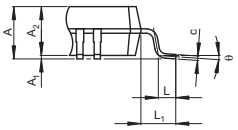
Figure B 176-Pin LFBGA (PLBG0176GA-A)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP176-24x24-0.50	PLQP0176KB-A	176P6Q-A/FP-176E/FP-176EV	1.8g





Terminal cross section



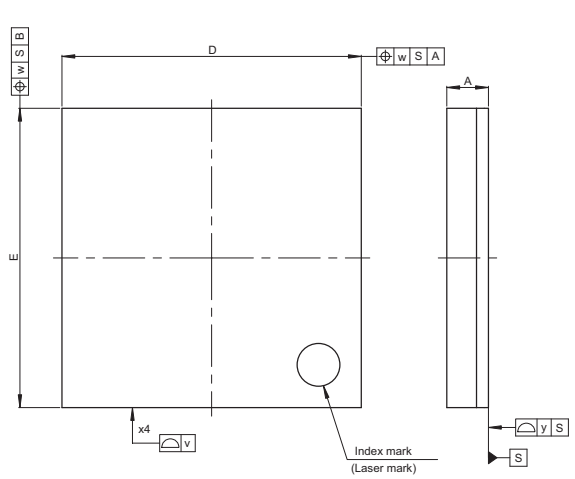
Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	23.9	24.0	24.1
E	23.9	24.0	24.1
A_2	—	1.4	—
H_D	25.8	26.0	26.2
H_E	25.8	26.0	26.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.10
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

Figure C 176-Pin LFQFP (PLQP0176KB-A)

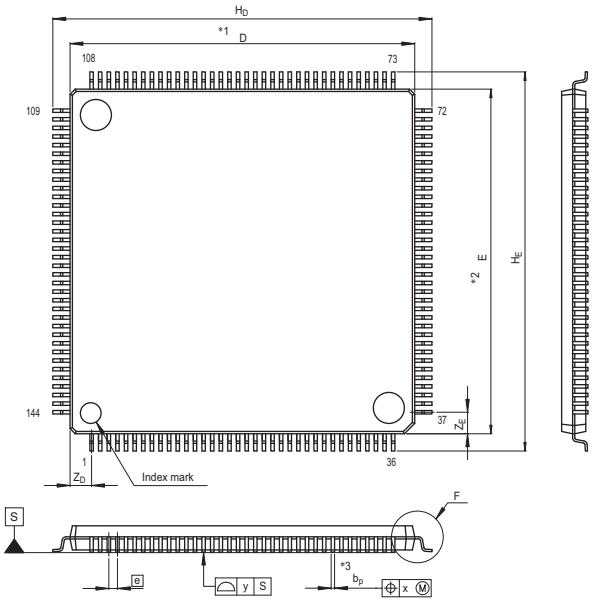
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-TFLGA145-7x7-0.50	PTLG0145KA-A	145F0G	0.1g

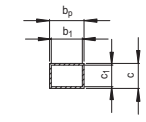


Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	7.0	—
E	—	7.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.05
e	—	0.5	—
b	0.21	0.25	0.29
b_1	0.29	0.34	0.39
x	—	—	0.08
y	—	—	0.08
Z_D	—	0.5	—
Z_E	—	0.5	—

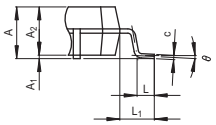
Figure D 145-Pin TFLGA (PTLG0145KA-A)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP144-20x20-0.50	PLQP0144KA-A	144P6Q-A / FP-144L / FP-144LV	1.2g





Terminal cross section



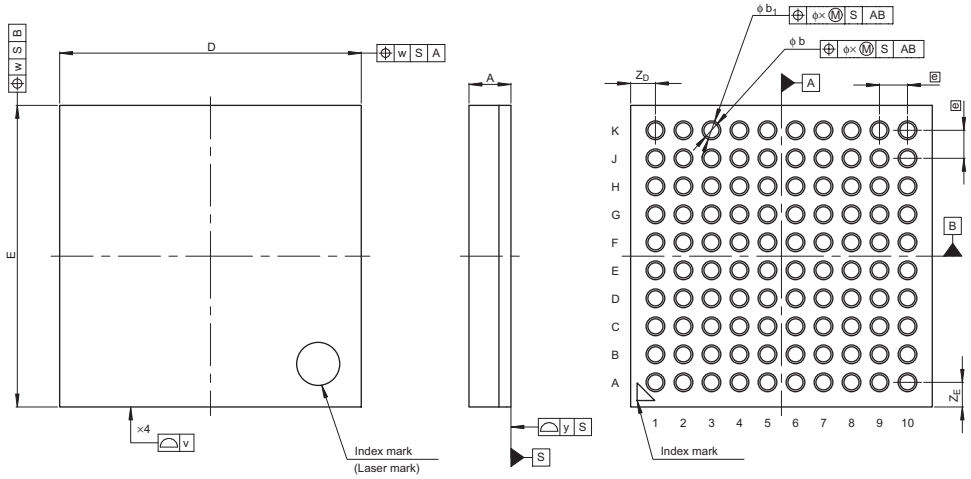
Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	19.9	20.0	20.1
E	19.9	20.0	20.1
A_2	—	1.4	—
H_D	21.8	22.0	22.2
H_E	21.8	22.0	22.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.10
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

Figure E 144-Pin LFQFP (PLQP0144KA-A)

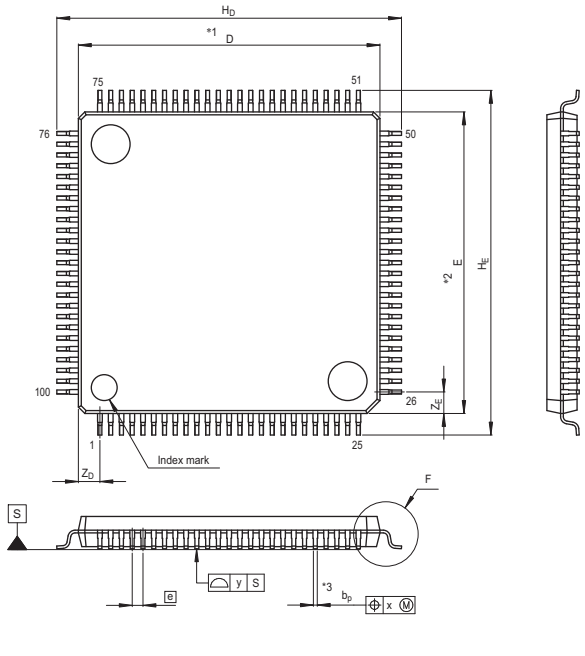
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-TFLGA100-7x7-0.65	PTLG0100JA-A	100F0G	0.1g

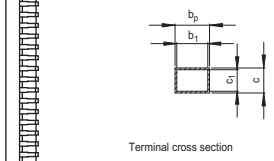


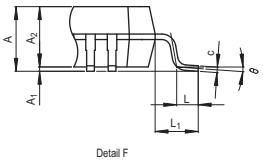
Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	7.0	—
E	—	7.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.05
e	—	0.65	—
b	0.31	0.35	0.39
b_1	0.385	0.435	0.485
x	—	—	0.08
y	—	—	0.10
Z_D	—	0.575	—
Z_E	—	0.575	—

Figure F 100-Pin TFLGA (PTLG0100JA-A)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP100-14x14-0.50	PLQP0100KB-A	100P6Q-A / FP-100U / FP-100UV	0.6g







NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A_2	—	1.4	—
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.0	—
Z_E	—	1.0	—
L	0.35	0.5	0.65
L_1	—	1.0	—

Figure G 100-Pin LFQFP (PLQP0100KB-A)

REVISION HISTORY	RX64M Group User's Manual: Hardware

Rev.	Date	Description	
		Page	Summary
0.90	Jul 24, 2013	—	First edition, issued
1.00	Jul 31, 2014	1. Overview	
		—	FINEC (Pin), deleted
		70	Table 1.1 Outline of Specifications (1/9), changed
		76	Table 1.1 Outline of Specifications (7/9), changed
		77	Table 1.1 Outline of Specifications (8/9), changed
		93	Table 1.4 Pin Functions (8/8), Note added
		7. Option Setting Memory	
		—	TM Enable Flag Register (TMEF), TM Identification Data Register (TMINF), added
		265	7.2.6 TM Enable Flag Register (TMEF), changed
		8. Voltage Detection Circuit (LVDA)	
		278	8.2.5 Voltage Monitoring Circuit Control Register (LVCMPCR), notes added
		285	Table 8.4 Procedures for Setting Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset so that Voltage Monitoring Operates, Note 4 added
		288	Table 8.6 Procedures for Setting Bits Related to the Voltage Monitoring 2 Interrupt and Voltage Monitoring 2 Reset so that Voltage Monitoring Operates, Note 4 added
		9. Clock Generation Circuit	
		296	9.2.1 System Clock Control Register (SCKCR), changed
		308	9.2.11 High-Speed On-Chip Oscillator Control Register (HOCOCR), changed
		313	9.2.15 Oscillation Stop Detection Status Register (OSTDSR), changed
		326	9.7.6 USB Clock, changed
		327	9.7.7 USBA Clock, changed
		15. Interrupt Controller (ICUA)	
		460	15.4.5.3 EXDMAC Start Request by Software Configurable Interrupts, changed
		464	15.5.3 Group Interrupts Using Edge Detection, changed
		18. DMA Controller (DMACAa)	
		618	18.2.7 DMA Address Mode Register (DMAMD), changed
		646	Figure 18.15 DMAC Interrupt Handling Routine to Resume/Terminate DMA Transfer, changed
		20. Data Transfer Controller (DTCa)	
		741, 742	20.6.2 Chain Transfer
		24. Multi-Function Timer Pulse Unit (MTU3a)	
		—	Terms unified: Counter clock→Count clock Synchronous presetting→Synchronous setting Preset→Set
		884	24.2.13 Timer Counter (TCNT), changed 24.2.14 Timer Longword Counter (TCNTLW), changed
		885	24.2.15 Timer General Register (TGR), changed
		886	24.2.16 Timer Longword General Register (TGRnLW) (n = A, B)
		972	Figure 24.48 Example of Complementary PWM Mode Setting Procedure, changed
		984, 985	Figure 24.56 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (1) to Figure 24.58 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (3), changed
		986 to 988	Figure 24.59 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (1) to Figure 24.63 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (5), changed
		27. 16-Bit Timer Pulse Unit (TPUa)	
		1325	Table 27.30 Correspondence between Channels and TSTRA.CSTn bit, changed
		1326	Table 27.31 TGR and TIOR Used for Input Capture by ELC, changed
		1328	(3) Input Capture Operation, changed
		28. Programmable Pulse Generator (PPG)	
		1352, 1353	Figure 28.6 Sample Setup Procedure for Normal Pulse Output (PPG0 Setting) and Figure 28.7 Sample Setup Procedure for Normal Pulse Output (PPG1 Setting), changed

Rev.	Date	Description	
		Page	Summary
1.00	Jul 31, 2014	1356, 1357	Figure 28.11 Sample Setup Procedure for Non-Overlapping Pulse Output (PPG0 Setting) and Figure 28.12 Sample Setup Procedure for Non-Overlapping Pulse Output (PPG1 Setting), changed
		32. Realtime Clock (RTCd)	
		1450	32.2.21 Frequency Register H/L (RFRH/RFRL), changed
		33. Watchdog Timer (WDTA)	
		1481	33.2.3 WDT Status Register (WDTSR), changed
		35. Ethernet Controller (ETHERC)	
		1510	Table 35.2 ETHERC I/O Pins (n = 0, 1), changed
		1518	35.2.7 Random Number Generation Counter Limit Setting Register (RDMLR), changed
		1531, 1532	Figure 35.6 MII Frame Transmit Timing During Normal Transmission, changed Figure 35.7 MII Frame Transmit Timing When Collision Occurs, changed Figure 35.8 MII Frame Transmit Timing When Transmit Error Occurs, changed Figure 35.9 MII Frame Receive Timing During Normal Reception, changed Figure 35.10 MII Frame Receive Timing for Receive Error Notification, changed Figure 35.11 MII Fame Receive Timing for False Carrier Notification, changed
		1533	Figure 35.12 RMII Frame Transmit Timing During Normal Transmission, changed Figure 35.13 RMII Frame Receive Timing During Normal Reception, changed Figure 35.14 RMII Frame Receive Timing When False Carrier Is Detected, changed
		36. PTP Module for the Ethernet Controller (EPTPC)	
		1628	36.2.78 Asymmetric Delay Setting Registers (DASYMRU, DASYMRL), changed
		1645	Table 36.17 List of the Registers for which Settings are Required, changed
		1675	36.6.2 Wait Cycles for Register Access, changed
		38. USB 2.0 FS Host/Function Module (USBb)	
		1717	Table 38.1 USB Specifications, changed
		1722	Table 38.4 Status of USB Data Bus Lines (D+ Line, D. Line), changed
		39. USB2.0 Full-Speed Host/Function Module (USBA)	
		1828	Figure 39.1 USBA Block Diagram, changed
		1829, 1830	39.2.1 System Configuration Control Register (SYSCFG), changed
		1832, 1833	39.2.3 System Configuration Status Register (SYSSTS0), changed
		1834 to 1836	39.2.5 Device State Control Register 0 (DVSTCTR0), changed
		1837 to 1839	39.2.6 FIFO Port Registers, changed
		1840, 1841	39.2.7 CFIFO Port Select Register (CFIFOSEL), changed
		1842, 1843	39.2.8 D0FIFO Port Select Register (D0FIFOSEL) D1FIFO Port Select Register (D1FIFOSEL), changed
		1844, 1845	39.2.9 CFIFO Port Control Register (CFIFOCTR) D0FIFO Port Control Register (D0FIFOCTR) D1FIFO Port Control Register (D1FIFOCTR), changed
		1846	39.2.10 Interrupt Enable Register 0 (INTENB0), changed
		1847	39.2.11 Interrupt Enable Register 1 (INTENB1), changed
		1848	39.2.12 BRDY Interrupt Enable Register (BRDYENB), changed 39.2.13 NRDY Interrupt Enable Register (NRDYENB), changed
		1849	39.2.14 BEMP Interrupt Enable Register (BEMPENB), changed
		1850	39.2.15 SOF Output Configuration Register (SOFCFG), changed
		1851	39.2.16 PHY Setting Register (PHYSET), changed
		1852, 1853	39.2.17 Interrupt Status Register 0 (INTSTS0), changed
		1854 to 1856	39.2.18 Interrupt Status Register 1 (INTSTS1), changed
		1857	39.2.19 BRDY Interrupt Status Register (BRDYSTS), changed 39.2.20 NRDY Interrupt Status Register (NRDYSTS), changed
		1858	39.2.21 BEMP Interrupt Status Register (BEMPSTS), changed
		1859	39.2.22 Frame Number Register (FRMNUM), changed
		1860	39.2.23 USB Address Register (USBADDR), changed
		1863	39.2.28 Default Control Pipe Configuration Register (DCPCFG), changed
		1864	39.2.29 Default Control Pipe Maximum Packet Size Register (DCPMAXP), changed
		1868	39.2.31 Pipe Window Select Register (PIPESEL), changed
		1869 to 1871	39.2.32 Pipe Configuration Register (PIPECFG), changed
		1872, 1873	39.2.33 Pipe Buffer Register (PIPEBUF), changed
		1874	39.2.34 Pipe Maximum Packet Size Register (PIPEMAXP), changed
		1875	39.2.35 Pipe Cycle Control Register (PIPEPERI), changed

Rev.	Date	Description	
		Page	Summary
1.00	Jul 31, 2014	1876 to 1880	39.2.36 Pipe n Control Register (PIPEnCTR) (n = 1 to 9), changed
		1881	39.2.37 Pipe n Transaction Counter Enable Register (PIPEnTRE) (n = 1 to 5), changed
		1882	39.2.38 Pipe n Transaction Counter Register (PIPEnTRN) (n = 1 to 5), changed
		1883	39.2.39 Device Address m Configuration Register (DEVADDm) (m = 0 to A), changed
		1884	39.2.40 Low Power Control Register (LPCTRL), changed
		1885	39.2.41 Low Power Status Register (LPSTS), changed
		1886, 1887	39.2.42 Battery Charging Control Register (BCCTRL), changed
		1888, 1889	39.2.43 Function L1 Control Register 1 (PL1CTRL1), changed
		1890	39.2.44 Function L1 Control Register 2 (PL1CTRL2), changed 39.2.45 Host L1 Control Register 1 (HL1CTRL1), changed
		1891, 1892	39.2.46 Host L1 Control Register 2 (HL1CTRL2), changed
		1893	39.2.48 Deep Standby USB Suspend/Resume Interrupt Register (DPUSR1R), changed
		1894	39.3.1.2 Selecting the Controller Function, changed
		1895	Figure 39.2 PHY Clock Settings in Classic-Only Mode, changed
		1896	Figure 39.3 PHY Clock Settings When Not Using Classic-Only Mode, changed
		1897, 1898	39.3.5 Interrupts, changed
		1898	39.3.5.1 Setting the USBAR Interrupt Signal Output Method, changed
		1899	Figure 39.5 Interrupt Association Diagram, changed
		1901, 1902	39.3.6.1 BRDY Interrupt, changed
		1904	39.3.6.2 NRDY Interrupt, changed
		1906	Figure 39.7 NRDY Interrupt Generation Timing (In Function Controller Operation), changed
		1907	39.3.6.3 BEMP Interrupt, changed
		1909	39.3.6.5 Control Transfer Stage Transition Interrupt in Function Controller Operation, changed
		1910	39.3.6.6 Frame Number Refresh Interrupt, changed
		1911	39.3.6.18 L1RSMEND Interrupt, changed
		1912	Table 39.18 Pipe Setting Items, changed
		1914	39.3.7.4 Setting the Maximum Packet Size, changed
		1917	39.3.8.1 FIFO Buffers, changed
		1918	39.3.8.2 Clearing the FIFO Buffers, changed
		1919	39.3.8.3 FIFO Port Functions, changed
		1920	39.3.8.4 DMA/DTC Transfers (D0FIFO and D1FIFO Ports), changed
		1921	39.3.8.5 Allocating the FIFO Buffers, changed Figure 39.12 Example of a FIFO Buffer Memory Map, changed
		1922	39.3.9.1 Control Transfer in Host Controller Operation, changed
		1924	39.3.11 Interrupt Transfer for Pipes 6 to 9, changed 39.3.12 Isochronous Transfer for Pipes 1 and 2, changed
		1926	39.3.12.2 Data PID, changed
		1927	Figure 39.13 Token Issuance When IITV[2:0] Bits are 000b, changed Figure 39.14 Token Issuance When IITV[2:0] Bits are 001b, changed
		1928	Figure 39.15 Relationship between Frames and Token Reception Expectance When the IITV[2:0] Bits are 000b, changed Figure 39.16 Relationship between Frames and Token Reception Expectance When the IITV[2:0] Bits are 001b, changed
		1929	(b) When the selected pipe is for isochronous IN transfers, changed (4) Isochronous Transfer Transmit Data Setup in Function Controller Operation, changed
		1930	Figure 39.17 Example of Data Setup Function Operation, changed
		1931, 1932	(5) Isochronous Transfer Transmit Buffer Flush in Function Controller Operation, changed
		1933	39.3.13 SOF Recovery Function, changed
		1934	39.3.14 Pipe Schedule, changed
		1935, 1936	39.3.15.1 Processing in Function Controller Operation, changed Figure 39.20 Processing as a Portable Device, changed
		1937 to 1939	39.3.15.2 Processing in Host Controller Operation, changed
		1940, 1941	39.3.16 Link Power Management Processing, changed
		1942 to 1945	39.3.17 USB External Connection Circuit, added
		40. Serial Communications Interface (SCIg, SCIh)	
		1946	Table 40.1 SCIg Specifications (2/2), changed
		1971, 1972	40.2.8 Serial Control Register (SCR) (2) Smart Card Interface Mode (SCMR.SMIF = 1), changed
		2041	40.6.5 SCI Initialization (Smart Card Interface Mode), changed

Rev.	Date	Description	
		Page	Summary
1.00	Jul 31, 2014	42. I2C Bus Interface (RIICa)	
		2154	42.2.7 I2C Bus Status Enable Register (ICSER), changed
		2168	Table 42.5 Examples of ICBRH/ICBRL Settings for Transfer Rate, changed
		2176	42.3.4 Master Receive Operation (5), changed
		2181	42.3.5 Slave Transmit Operation (3), changed
		2196	42.8.2 NACK Reception Transfer Suspension Function, changed
		2198	42.9.1 Master Arbitration-Lost Detection (MALE Bit), changed
		2206	42.11.2 Extra SCL Clock Cycle Output Function, changed
		2208	42.12 SMBus Operation, changed
		2209	42.12.3 SMBus Host Notification Protocol (Notify ARP Master Command), changed
		2210	Table 42.6 Interrupt Sources, changed
		43. CAN Module (CAN)	
		2248	BLIF Flag (Bus Lock Detect Flag), changed
		2253	Figure 43.9 Transition between CAN Operating Modes, changed
		2255	Table 43.8 Operation in CAN Reset Mode and CAN Halt Mode, changed
		44. Serial Peripheral Interface (RSPIa)	
		2284	44.2.8 RSPI Bit Rate Register (SPBR), changed
		45. Quad Serial Peripheral Interface (QSPI)	
		2351	Figure 45.1 QSPI Block Diagram, changed
		2359	45.2.8 QSPI Bit Rate Register (SPBR), changed
		2364 to 2366	45.2.13 QSPI Command Register n (SPCMDn) (n = 0 to 3), changed
		2369	45.2.16 QSPI Transfer Data Length Multiplier Setting Register n (SPBMULn) (n = 0 to 3), changed
		2386	(c) Sequence Control, changed
		47. Serial Sound Interface (SSI)	
		2405, 2408	47.2.1 Control Register (SSICR), changed
		2409	47.2.2 Status Register (SSISR), changed
		2411	47.2.3 FIFO Control Register (SSIFCR), changed
		2426	Figure 47.19 Transmission Using the DMAC/DTC, changed
		2427	Figure 47.20 Transmission Using Interrupts, changed
		50. MultiMediaCard Interface (MMCIF)	
		2511	50.2.8 Response Register 0 (CERESP0), Response Register 1 (CERESP1), Response Register 2 (CERESP2), Response Register 3 (CERESP3), changed
		58. 12-Bit D/A Converter (R12DA)	
		2699	58.3 Operation, changed
		63. Flash Memory	
		2726	Table 63.1 Specifications of Code Flash Memory and Data Flash Memory, changed
		2728	63.2 Structure of Memory, changed Figure 63.2 Mapping of the Code Flash Memory, changed
		2732, 2733	63.5 Overview of Functions, changed
		2739	63.11.2 ROM Code Protection, changed
		2746	Table 63.16 Commands Available in the Command Waiting Phase, changed
		2748	Table 63.17 Common Transfer Sequence, changed
		2750	63.12.9 Device Type Acquisition Command, changed
		2759	63.12.16 Blank Check Command, changed
		2760	63.12.17 Erase Command, changed
		2761	63.12.18 Area Erase Command, added
		2762	63.12.19 Programming Command, changed
		2764	63.12.20 Read Command, changed
		2782	63.12.35 Configuration Clear Command, changed
		2783	63.12.36 TM Setting Command, added
		2784, 2785	63.12.37 TM Acquisition Command, added
		2786	63.12.38 Simple Addition Checksum Command, changed
		2789	Figure 63.17 Example of Flash Memory Configuration Information, changed
		2793	63.12.41 Flow for Erasure when Programming Commands are Prohibited, changed Figure 63.20 Flow for Erasure when Programming Commands are Prohibited, changed
		2798 to 2803	63.16 Trusted Memory, added
		2804	(6) Items prohibited during programming and erasure, changed

Rev.	Date	Description	
		Page	Summary
1.00	Jul 31, 2014	64. Electrical Characteristics	
		2806	Table 64.2 DC Characteristics (1), changed
		2808	Table 64.4 DC Characteristics (3), changed
		2817	Table 64.15 Clock Timing (Except for Sub-Clock Related), changed
		2825	Figure 64.18 External Bus Timing/Normal Read Cycle (Bus Clock Synchronized), changed
		2826	Figure 64.19 External Bus Timing/Normal Write Cycle (Bus Clock Synchronized), changed
		2827	Figure 64.20 External Bus Timing/Page Read Cycle (Bus Clock Synchronized), changed Figure 64.21 External Bus Timing/Page Write Cycle (Bus Clock Synchronized), changed
		2847	Table 64.33 RSPI Timing, changed
		2852	Table 64.35 QSPI Timing, changed
		2854	Table 64.36 RIIC Timing (1), changed
		2855	Table 64.37 RIIC Timing (2), changed
		2856	Figure 64.56 RIIC Bus Interface Input/Output Timing and Simple IIC Bus Interface Input/Output Timing, changed
		2859	Table 64.39 MMC Host Interface Timing, changed
		2867	Table 64.42 On-Chip USB Low Speed (Host Only) Characteristics (DP and DM Pin Characteristics), changed
		2873	Table 64.50 Power-on Reset Circuit and Voltage Detection Circuit Characteristics, changed

Classifications

- Items with Technical Update document number: Changes according to the corresponding issued Technical Update
- Items without Technical Update document number: Minor changes that do not require Technical Update to be issued

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Oct 24, 2016	All	Terms unified: GPTa → GPTA LQFP → LFQFP (DTC) DTC Start Enable → DTC Transfer Request Enable (DMAC)(EXDMAC) DMA activation sources → DMA request sources DMAC activation request → DMA transfer request DMA activation, DMAC activation → DMA request DMA start → DMA transfer EXDMAC activation is enabled, EXDMAC activation enable → EXDMAC module start EXDMAC activation → trigger	
		Features		
		69	AES key lengths, changed	TN-RX*-A122A/E
		1. Overview		
		70	Table 1.1 Outline of Specifications (1/9), changed	TN-RX*-A127A/E
		73	Table 1.1 Outline of Specifications (4/9), changed	
		78	Table 1.1 Outline of Specifications (9/9), changed	TN-RX*-A122A/E
		96	Figure 1.5 Pin Assignment (176-Pin LFQFP), changed	
		116	Table 1.7 List of Pin and Pin Functions (145-Pin TFLGA) (2/5), changed	
		117	Table 1.7 List of Pin and Pin Functions (145-Pin TFLGA) (3/5), changed	
		120	Table 1.8 List of Pin and Pin Functions (144-Pin LFQFP) (1/5), changed	
		123	Table 1.8 List of Pin and Pin Functions (144-Pin LFQFP) (4/5), changed	
		126	Table 1.9 List of Pin and Pin Functions (100-Pin TFLGA) (2/4), changed	
		127	Table 1.9 List of Pin and Pin Functions (100-Pin TFLGA) (3/4), changed	
		131	Table 1.10 List of Pin and Pin Functions (100-Pin LFQFP) (3/4), changed	
		3. Operating Modes		
		164	3.3.5 USB Boot Mode, changed	
		5. I/O Registers		
		171	(4) Notes on Sleep Mode and Mode Transitions, added	
		173	Table 5.1 List of I/O Registers (Address Order) (2 / 69) 0008 1200h, 0008 1201h, 0008 1204h, 0008 1208h, added	TN-RX*-A127A/E
		208	Table 5.1 List of I/O Registers (Address Order) (37 / 69) 0008 C296h, added	
		210	Table 5.1 List of I/O Registers (Address Order) (39 / 69), changed	TN-RX*-A152A/E
		211	Table 5.1 List of I/O Registers (Address Order) (40 / 69), changed	
		212	Table 5.1 List of I/O Registers (Address Order) (41 / 69), changed	
		220	Table 5.1 List of I/O Registers (Address Order) (49 / 69) 000C 0438h, 000C 046Ch, deleted	
		234	Table 5.1 List of I/O Registers (Address Order) (63 / 69), changed	
		240	Table 5.1 List of I/O Registers (Address Order), Note 6 added	TN-RX*-A152A/E
		6. Resets		
		253	6.4 Usage Notes, added	TN-RX*-A123A/E
		7. Option-Setting Memory		
		264	7.2.6 TM Enable Flag Register (TMEF), changed 7.2.7 TM Identification Data Register (TMINF), changed	
		265	7.3.1 UB Code A, changed 7.3.2 UB Code B, changed	
		9. Clock Generation Circuit		
		299	9.2.4 PLL Control Register (PLLCR), Note 1 added	
		313	9.2.16 Main Clock Oscillator Wait Control Register (MOSCWTCR), changed	
		314	9.2.17 Sub-Clock Oscillator Wait Control Register (SOSCWTCR), changed	
		315	9.2.18 Main Clock Oscillator Forced Oscillation Control Register (MOFCR), changed	TN-RX*-A150A/E

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Oct 24, 2016	317	9.3.1 Connecting a Crystal Resonator, changed Figure 9.2 Example of Crystal Connection, changed	TN-RX*-A150A/E
		318	Table 9.4 Crystal Characteristics (Reference Values), changed	
		327, 328	9.8 Clock Source Switching, changed	
		333	9.10.6 Notes on Using a Low CL Crystal Unit, changed	
		333	9.10.7 Notes on Using Power-On Reset and PLL Circuit Together, added	TN-RX*-A123A/E
		11. Low Power Consumption		
		All	Terms unified: RTC Cycle → RTC Periodic	
		355	11.2.5 Module Stop Control Register D (MSTPCRD), modified (address: 0008 001Ah → 0008 001Ch)	
		383	11.6.3.1 Transition to Software Standby Mode, changed	
		15. Interrupt Controller (ICUA)		
		409	Table 15.1 ICU Specifications (1/2), changed	
		425	15.2.14 Non-Maskable Interrupt Status Register (NMISR), changed	
		473	15.7.5 Setting Non-Maskable Interrupts, changed	
		16. Buses		
		481	Table 16.3 Order of Priority for Bus Masters, changed	
		482	16.2.5 Write Buffer Function (Internal Peripheral Bus), changed	
		514	16.3.21 Bus Error Status Register 1 (BERSR1), changed	
		591	16.9 Usage Notes, added	
		17. Memory-Protection Unit (MPU)		
		599	17.2.5 Memory-Protection Error Status-Clearing Register (MPECLR), changed	
		611	17.4.4 Processing in Response to Memory-Protection Errors, changed	
		18. DMA Controller (DMACAa)		
		All	Terms unified: DMAC activation is disabled → DMAC stops DMAC activation is enabled → DMAC module start	
		613	Table 18.1 Specifications of DMAC, ‘Transfer end interrupt’ changed	
		617	18.2.4 DMA Block Transfer Count Register (DMCRB), changed	
		19. EXDMA Controller (EXDMACa)		
		All	Terms unified: EXDMAC activation is disabled → EXDMAC stops EXDMAC activation is enabled → EXDMAC module start	
		653	Table 19.1 Specifications of EXDMAC (2/2), ‘Transfer end interrupt’ changed	
		658	19.2.4 EXDMA Block Transfer Count Register (EDMCRB), changed	
		659	19.2.5 EXDMA Transfer Mode Register (EDMTMD), changed	
		20. Data Transfer Controller (DTCa)		
		All	Terms unified: a single activation → a single transfer request	
		725	20.2.5 DTC Transfer Count Register A (CRA), changed	
		21. Event Link Controller (ELC)		
		755	Table 21.2 Correspondence between the ELSRn Register and the Peripheral Modules, notes added	
		779	21.3.6 Example of Procedure for Linking Events, Note added	
		780	21.4.1 Setting ELSRn Register, changed	
		22. I/O Ports		
		795	Table 22.3 Handling of Unused Pins, changed 22.5.1 Note on the Port Direction Register (PDR), changed	
		23. Multi-Function Pin Controller (MPC)		
		815	23.2.2 P0n Pin Function Control Register (P0nPFS) (n = 0 to 3, 5, 7), changed	
		825	Table 23.13 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LFQFP, changed	
		833	23.2.17 PFn Pin Function Control Register (PFnPFS) (n = 0 to 2, 5), changed	TN-RX*-A122A/E
		834	Table 23.24 Register Settings for Input/Output Pin Function in 177-/145-/100-Pin TFLGA, 176-Pin LFBGA, 176-/144-/100-Pin LFQFP, changed	
		847	Table 23.27 Register Settings, modified (External bus)	

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Oct 24, 2016	24. Multi-Function Timer Pulse Unit (MTU3a)		
		All	Terms unified: count clock → count clock source input clock → count clock, internal clock, count clock source input clock edge → clock edge counter input clock → count clock	
		863	24.2.5 Timer Mode Register 3 (TMDR3), changed	
		866	Table 24.13 TIORH (MTU0), Note 1 added Table 24.14 TIORL (MTU0), Note 2 added	
		873, 874	Table 24.27 TIORH (MTU0), Note 1 added Table 24.28 TIORL (MTU0), Note 2 added	
		880	Table 24.41 TIORU, TIORV, and TIORW (MTU5), Note 1 added	
		895	24.2.18 Timer Synchronous Registers (TSYRA, TSYRB), changed	
		900, 901	24.2.21 Timer Output Master Enable Registers (TOERA, TOERB), changed	
		908	24.2.25 Timer Gate Control Register A (TGCRA), changed	
		918 to 921	24.2.35 Timer A/D Converter Start Request Control Register (TADCR), changed	TN-RX*-A119A/E
		931	24.2.43 Bus Master Interface, deleted	
		949, 950, 953	24.3.5 PWM Modes, changed	
		968	24.3.7 Reset-Synchronized PWM Mode, modified	
		1013	Figure 24.92 Example of Procedure for Specifying A/D Converter Start Request Delaying Function (MTU3 and MTU4), notes changed	TN-RX*-A119A/E
		1014	24.3.9 A/D Converter Start Request Delaying Function Figure 24.93 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation, changed (3) Period in Which A/D Converter Start Requests are Enabled, added	
		1015	24.3.9 A/D Converter Start Request Delaying Function (4) Buffer Transfer, changed	
		1016	24.3.9 A/D Converter Start Request Delaying Function (5) A/D Converter Start Request Delaying Function Linked with Interrupt Skipping Function 1, changed	
		1050	Figure 24.148 Contention between TCNT Write Operation and Overflow, changed	
		1055, 1056	24.6.28 Usage Notes on A/D Converter Delaying Function in Complementary PWM Mode, added	TN-RX*-A119A/E
		1058	24.7.3 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, changed	
		25. Port Output Enable 3 (POE3a)		
		1088	Table 25.3 Pin Combinations, changed	
		1094	25.2.6 Input Level Control/Status Register 6 (ICSR6), changed	
		1095	25.2.7 Output Level Control/Status Register 1 (OCSR1), changed	
		1096	25.2.8 Output Level Control/Status Register 2 (OCSR2), changed	
		1120 to 1125	25.3 Operation, changed	TN-RX*-A158A/E
		1130	25.3.7 Release from High-Impedance State, changed	
		26. General PWM Timer (GPTA)		
		1144	26.2.4 General PWM Timer Hardware Source Clear Control Register (GTHCCR), changed	
		1145	26.2.5 General PWM Timer Hardware Start Source Select Register (GTHSSR), b15 to b12 changed	
		1157	26.2.14 General PWM Timer Control Register (GTCR), changed	
		1161	26.2.15 General PWM Timer Buffer Enable Register (GTBER), changed	
		1165, 1166	26.2.18 General PWM Timer Status Register (GTST), changed	
		1171	26.2.28 General PWM Timer Dead Time Control Register (GTDTCR), changed	
		1208	26.3.4 Automatic Dead Time Setting Function, changed	
		1223	26.3.7.1 Synchronous Clear Operation, changed	
		1249	26.6.1 Event Signal Output to the ELC, changed	
		1263, 1264	26.9.4 Order of Priority in Event Counter Operation, added	

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Oct 24, 2016	27. 16-Bit Timer Pulse Unit (TPUa)		
		All	Terms unified: TCNT counter → TCNT presetting → setting counter clock, counter input clock → count clock PCLK → PCLK cycles	
		1329	27.10.11 Conflict between Overflow/Underflow and Counter Clearing, changed	
		29. 8-Bit Timer (TMR)		
		All	Terms unified: Frequency dividing clock → internal clock external reset, external reset input → external counter reset signal external clock → external count clock counter clock → count clock counter external reset → external counter reset TCNT input clock → TCNT count clock	
		1380	29.2.4 Timer Control Register (TCR), Note 1 changed	
		1381	29.2.5 Timer Counter Control Register (TCCR), Note 1 changed	
		1382	Table 29.5 Clock Input to TCNT and Count Condition, Note 1 changed	
		30. Compare Match Timer (CMT)		
		1401	Table 30.1 CMT Specifications, changed Figure 30.1 CMT (Unit 0) Block Diagram, Note 1 added	
		32. Realtime Clock (RTCd)		
		1439	32.2.2 Second Counter (RSECCNT)/Binary Counter 0 (BCNT0), changed	
		1440	32.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1), changed	
		1441	32.2.4 Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2), changed	
		1442	32.2.5 Day-of-Week Counter (RWKCNT)/Binary Counter 3 (BCNT3), changed	
		1443	32.2.6 Date Counter (RDAYCNT), changed	
		1444	32.2.7 Month Counter (RMONCNT), changed	
		1445	32.2.8 Year Counter (RYRCNT), changed	
		1446	32.2.9 Second Alarm Register (RSECAR)/Binary Counter 0 Alarm Register (BCNT0AR), changed	
		1447	32.2.10 Minute Alarm Register (RMINAR)/Binary Counter 1 Alarm Register (BCNT1AR), changed	
		1448	32.2.11 Hour Alarm Register (RHRAR)/Binary Counter 2 Alarm Register (BCNT2AR), changed	
		1449	32.2.12 Day-of-Week Alarm Register (RWKAR)/Binary Counter 3 Alarm Register (BCNT3AR), changed	
		1450	32.2.13 Date Alarm Register (RDAYAR)/Binary Counter 0 Alarm Enable Register (BCNT0AER), changed	
		1451	32.2.14 Month Alarm Register (RMONAR)/Binary Counter 1 Alarm Enable Register (BCNT1AER), changed	
		1452	32.2.15 Year Alarm Register (RYRAR)/Binary Counter 2 Alarm Enable Register (BCNT2AER), changed	
		1456, 1457	32.2.18 RTC Control Register 2 (RCR2), changed	
		1458	32.2.19 RTC Control Register 3 (RCR3), changed	
		1460, 1461	32.2.21 Frequency Register H/L (RFRH/RFRL), changed	
		1475	Figure 32.7 Using Alarm Function, changed	
		35. Ethernet Controller (ETHERC)		
		1532	35.2.14 Broadcast Frame Receive Count Setting Register (BCFRR), changed	
		1536	35.2.22 Frame Receive Error Counter Register (FRECR), changed	
		1537	35.2.25 Received Alignment Error Frame Counter Register (RFCR), changed	
		1548, 1549	35.5.3 Handling Errors in Control Information, added	TN-RX*-A125A/E
		36. PTP Module for the Ethernet Controller (EPTPC)		
		1562	36.2.5 STCA Status Register (STSR), changed	
		1580	36.2.21 Negative Gradient Limit Registers (MLIMITRU, MLIMITRM, MLIMITRL), changed	
		1599	36.2.37 SYNFP Status Register (SYSR), changed	TN-RX*-A125A/E

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Oct 24, 2016	1643	36.2.83 PTP Reset Register (PTRSTR), changed	
		1676	36.3.16 Local Clock Counter, changed	
		1682	Figure 36.36 MINT Interrupt Request and IPLS Interrupt Request, changed	
		1683	Figure 36.37 Details on Interrupt Requests of the Pulse Output Timer, changed	
		1684	Figure 36.38 Relation between the Pulse Output Timer and ELC, changed	
		1685	36.6.1 Restrictions on Access to Registers, changed	
		37. DMA Controller for the Ethernet Controller (EDMACa)		
		1694 to 1697	37.2.6 ETHERC/EDMAC Status Register (EDMACn.EESR), b23 changed	
		1698 to 1700	37.2.7 PTP/EDMAC Status Register (PTPEDMAC.EESR), b23 changed	
		1701, 1702	37.2.8 ETHERC/EDMAC Status Interrupt Enable Register (EDMACn.EESIPR), b23 changed	
		1705	37.2.9 PTP/EDMAC Status Interrupt Enable Register (PTPEDMAC.EESIPR), b23 changed	
		1709	37.2.14 Receive Method Control Register (RMCR), changed	
		1718	37.3.1.1 Transmit Descriptor, changed	
		1723	37.3.3 Reception, Note 1 added	
		1727	37.5.2 Stopping the EDMAC during Operations, changed	
		1728	37.5.3 Illegal Address Access Detection, added	
		38. USB 2.0 FS Host/Function Module (USBb)		
		1730	Table 38.2 USB Pin Configuration, Note 1 deleted	
		1738, 1739	38.2.4 CFIFO Port Register (CFIFO) D0FIFO Port Register (D0FIFO) D1FIFO Port Register (D1FIFO), changed	
		1739	Table 38.5 Endian Operation in 16-Bit Access, changed Table 38.6 Endian Operation in 8-Bit Access, deleted	
		1741, 1743	38.2.5 CFIFO Port Select Register (CFIFOSEL) D0FIFO Port Select Register (D0FIFOSEL) D1FIFO Port Select Register (D1FIFOSEL), changed	
		1752	38.2.13 Interrupt Status Register 0 (INTSTS0), changed	
		1779	38.2.32 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9), changed	
		1789	38.2.36 PHY Cross Point Adjustment Register (PHYSLEW), changed	
		1795	Figure 38.3 Functional Connection of USB Connector in Self-Powered State, Note 1 changed	
		39. USB2.0 Full-Speed Host/Function Module (USBA)		
		1847, 1848	39.2.5 Device State Control Register 0 (DVSTCTR0), changed	
		1849	39.2.6 FIFO Port Registers (CFIFO, D0FIFO, D1FIFO), modified	
		1853	39.2.7 CFIFO Port Select Register (CFIFOSEL), changed	
		1855	39.2.8 D0FIFO Port Select Register (D0FIFOSEL) D1FIFO Port Select Register (D1FIFOSEL), changed	
		1864	39.2.17 Interrupt Status Register 0 (INTSTS0), Note 5 changed	
		1866, 1868	39.2.18 Interrupt Status Register 1 (INTSTS1), changed	TN-RX*-A156A/E
		1893	39.2.38 Pipe n Transaction Counter Register (PIPEnTRN), changed	
		1907	39.3.4 Notes on Stopping Clock, changed	
		1908	Table 39.15 List of Interrupt Functions (1/2), changed	
		1909	Table 39.15 List of Interrupt Functions (2/2), changed	
		1921	39.3.6.17 LPMEND Interrupt, modified	
		1927	Table 39.20 Buffer Status Indicated by the INBUFM Flag, changed	
		1932	39.3.9.1 Control Transfer in Host Controller Operation, modified	
		1934	39.3.12 Isochronous Transfer for Pipes 1 and 2, changed	
		1939	39.3.12.3 Interval Counter (3) Interval Counting and Transfer Control in Function Controller Operation (b) When the selected pipe is for isochronous IN transfers, changed	
		1943	39.3.13 SOF Recovery Function, changed	
		40. Serial Communications Interface (SCIg, SCIh)		
		1969	40.2.7 Serial Mode Register (SMR), b7 changed	
		1974 to 1977	40.2.8 Serial Control Register (SCR), changed	
		1978, 1981	40.2.9 Serial Status Register (SSR), changed	
		1992	Table 40.22 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 372), deleted	

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Oct 24, 2016	1996, 1997	40.2.13 Serial Extended Mode Register (SEMR), changed	
		2021	40.3.5 CTS and RTS Functions, changed	
		2022	40.3.6 SCI Initialization (Asynchronous Mode), changed	TN-RX*-A130B/E
		2037	40.5.2 CTS and RTS Functions, changed	
		2089	Table 40.31 Interrupt Sources, changed	
		2102	40.14.15 Note on Stopping Reception When Using the RTS Function in Asynchronous Mode, added	TN-RX*-A151A/E
		41. FIFO Embedded Serial Communications Interface (SCIFA)		
		2109	41.2.7 Serial Status Register (FSR), changed	
		2121	41.2.12 Serial Port Register (SPTR), changed Table 41.13 Controlling the TXD Pin, changed	
		2122	Table 41.14 Controlling the CTS# Pin, changed Table 41.15 Controlling the RTS# Pin, changed	
		2123	41.2.13 Line Status Register (LSR), changed	
		2131	Figure 41.7 Transmitting Serial Data in Asynchronous Mode, Note 1 added	
		2136	Figure 41.13 SCIFA Initialization in Clock Synchronous Mode, changed	
		42. I^2C-bus Interface (RIICa)		
		All	Terms unified: I^2C bus → I^2C-bus transferred frames → transferred bytes receive frames → receive bytes address frame → address byte standard → specification [Sm], [Fm], [W], [R] → (Sm), (Fm), (write), (read),	
		2169	42.2.9 I^2C-bus Status Register 1 (ICSR1), HOA flag changed	
		2206	Figure 42.31 Suspension of Data Transfer When NACK is Received (NACKE = 1), changed	
		43. CAN Module (CAN)		
		2236, 2238	43.2.6 Mailbox Register j (MBj) (j = 0 to 31), Note added Table 43.4 CANi Mailbox Memory Mapping, Note added	TN-RX*-A152A/E
		44. Serial Peripheral Interface (RSPIa)		
		All	Terms unified: transmission buffer → transmit buffer reception buffer → receive buffer mode-fault errors → mode fault errors	
		2288, 2289	44.2.4 RSPI Status Register (SPSR), changed	
		2291	44.2.5 RSPI Data Register (SPDR), changed	
		2297	44.2.9 RSPI Data Control Register (SPDCR), changed	
		2305	44.2.14 RSPI Command Registers 0 to 7 (SPCMD0 to SPCMD7), SPB[3:0] bits changed	
		2307	Table 44.5 Relationship between RSPI Modes and SPCR Settings and Description of Each Mode, changed	
		2326	44.3.6.1 Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0), changed	
		2327	44.3.6.2 Transmit Operations Only (SPCR.TXMD = 1), changed	
		2328, 2329	44.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts, changed Figure 44.26 Operation Example of SPTI and SPRI Interrupts, changed	
		2330	44.3.8 Error Detection, changed Table 44.8 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function, changed	
		2330	44.3.8.1 Overrun Error, changed Figure 44.27 Operation Example of SPRF and OVRF Flags, changed	
		2332	Figure 44.28 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 1), changed Figure 44.29 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 0), changed	
		2335	44.3.9.1 Initialization by Clearing the SPE Bit, changed	
		2336, 2343, 2345	44.3.10.1 Master Mode Operation, changed	TN-RX*-A147A/E
		2343	Figure 44.36 Flowchart in Master Mode (Transmission), changed	
		2344	Figure 44.37 Flowchart in Master Mode (Reception), changed	

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Oct 24, 2016	2345	Figure 44.38 Flowchart for Master Mode (Error Processing), changed	
		2346, 2349	44.3.10.2 Slave Mode Operation, changed	
		2348	Figure 44.40 Flowchart in Slave Mode (Transmission), changed Figure 44.41 Flowchart in Slave Mode (Reception), changed	
		2349	Figure 44.42 Flowchart for Slave Mode (Error Processing), changed	
		2350	44.3.11.1 Master Mode Operation, changed	
		2354	44.3.11.2 Slave Mode Operation, changed	
		2358	Table 44.13 Interrupt Sources of RSPI, changed	
		2361	44.5.4 Notes on the SPRF and SPTEF flags, added	
		45. Quad Serial Peripheral Interface (QSPI)		
		2369	45.2.5 QSPI Data Register (SPDR), changed	
		2376, 2377	45.2.13 QSPI Command Register n (SPCMDn) (n = 0 to 3), changed Table 45.4 SPI Modes Definitions, added	
		2385	Figure 45.2 Clock Setting and Transfer Timing, changed	
		2407	45.4.3 Notes on Using the Serial Flash Memory, added	TN-RX*-A154A/E
		47. Serial Sound Interface (SSI)		
		2418 to 2421	47.2.1 Control Register (SSICR), changed	
		2420	Table 47.3 SSITXD0, SSIRXD0, and SSIDATA1 Pin States, changed	
		2421	Table 47.4 Setting of SCKP Bit and Signal Timing, added	
		2422	47.2.2 Status Register (SSISR), changed	TN-RX*-A133B/E
		2424, 2425	47.2.3 FIFO Control Register (SSIFCR), changed	
		2426, 2427	47.2.4 FIFO Status Register (SSIFSR), changed	TN-RX*-A133B/E
		2431	47.3.2 Non-Compressed Mode (7) Operating Settings Related to Word Length, changed	
		2436	47.3.3 WS Continue Mode, changed	
		2443	Table 47.7 SSI Interrupt Sources, changed	
		2444 to 2446	47.5.4 Notes Regarding Clearing the Status Flag, added	TN-RX*-A133B/E
		48. Sampling Rate Converter (SRC)		
		2459	48.3.2 Data Input, changed Figure 48.3 Flowchart for Data Input, changed	TN-RX*-A155A/E
		2460	Figure 48.4 Flowchart for Data Output, changed	
		2463	48.5.6 Conversion immediately after the Data Set in the SRCID Register, added	
		49. SD Host Interface (SDHI)		
		2464	Table 49.2 Pin Configuration of the SDHI, changed	
		2469	49.2.5 Response Register 10 (SDRSP10), Response Register 32 (SDRSP32), Response Register 54 (SDRSP54), Response Register 76 (SDRSP76), changed	
		2480	49.2.11 Transfer Data Size Register (SDSIZE), changed	TN-RX*-A122A/E
		2483	49.2.13 SD Error Status Register 1 (SDERSTS1), changed	
		2514	Table 49.8 Interrupt Sources, changed	
		2515	49.4.1 DMA Transfer Triggered by Interrupt Requests, changed	
		52. Boundary Scan		
		2602	52.3.2 List of Commands, modified	
		2603	52.4 Usage Notes, modified	
		57. 12-Bit A/D Converter (S12ADC)		
		2645	57.2.18 A/D Group Scan Priority Control Register (ADGSPCR), Note 2 added	TN-RX*-A124A/E
		2686	57.3.4.3 Operation under Group-A Priority Control, changed	
		2700	Table 57.11 Times for Conversion during Scanning (in Numbers of Cycles of the ADCLK and PCLKB), modified	
		2705	Figure 57.41 Procedures for Clear Operation by Software through the ADCSR.ADST Bit, changed	TN-RX*-A124A/E
		2707	57.5.10 Notes on Noise Prevention, added 57.5.11 Notes on Operating Modes and Status Bits, added	
		58. 12-Bit D/A Converter (R12DA)		
		All	Terms unified: output amplifier → output buffer amplifier (except for used as a register name)	
		2715	58.3 Operation, changed	

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Oct 24, 2016	2716, 2717	58.3.1 Measure against Interference between D/A and A/D Conversion, changed Figure 58.4 Example When the 12-Bit D/A Converter Cannot Capture the 12-Bit A/D Converter Synchronous D/A Conversion Enable Input Signal, deleted	
		2719	58.6.5 Initial Setting Procedure when the Output Buffer Amplifier is Used, changed	TN-RX*-A122A/E
		59. Temperature Sensor (TEMPS)		
		2720	Table 59.1 Specifications of Temperature Sensor, changed	TN-RX*-A127A/E
		2721	59.2.2 Temperature Sensor Calibration Data Register (TSCDR), added	
		2722, 2723	59.3.1 Preparation for Using the Temperature Sensor, changed	
		2723	Figure 59.2 Error in the Measured Temperature, added	
		61. RAM		
		2734	Table 61.1 Specifications of RAM, changed	TN-RX*-A127A/E
		2739	61.2.10 RAM Operating Mode Control Register (RAMMODE), added	
		2740	61.2.11 RAM Error Status Register (RAMSTS), added 61.2.12 RAM Error Address Capture Register (RAMECAD), added	
		2741	61.2.13 RAM Protection Register (RAMPRCR), added	
		2742	61.3.3 Parity Checking, added	
		2743	61.3.4 RAM Error Interrupt Function, changed	
		2745	61.3.6 Interrupt Source, changed 61.4 Usage Notes, added	
		63. Flash Memory		
		All	Terms unified: code flash → code flash memory data flash → data flash memory	
		2747	Table 63.1 Specifications of Code Flash Memory and Data Flash Memory, changed	TN-RX*-A127A/E
		2752	63.3.2 Unique ID Register n (UIDRn) (n = 0 to 2), changed	
		2756	Table 63.6 Lists of Security Functions, changed	
		2760	63.10.2 Boot Mode (for the USB Interface), changed	TN-RX*-A157A/E
		2761	63.11 ID Code Protection, changed 63.11.2 ID Code Protection, added	
		2768	Table 63.14 Commands Available in the Command Waiting Phase, modified	
		2770	Table 63.15 Common Transfer Sequence, modified (Erase command → Block Erase command)	
		2813	Figure 63.18 Example of Reprogramming Procedure, modified (Erase command → Block Erase command)	
		2815	Figure 63.20 Flow for Erasure when Programming Commands are Prohibited, modified (Erase command → Block Erase command)	
		2827, 2828	63.17 Usage Notes, changed (2) Suspension During Programming/Erasure, added (9) Programming/Erasure in Low-Speed Operating Modes 1 and 2, added	
		64. Electrical Characteristics		
		2829	Table 64.1 Absolute Maximum Rating, changed	TN-RX*-A160A/E
		2830	Table 64.2 DC Characteristics (1), changed	TN-RX*-A159A/E TN-RX*-A160A/E
		2831	Table 64.3 DC Characteristics (2), changed	TN-RX*-A159A/E
		2873	Figure 64.48 RSPI Timing (Master, CPHA = 0) (Bit Rate: PCLKB Division Ratio Set to 1/2), changed	
		2896	Table 64.49 Temperature Sensor Characteristics, changed	TN-RX*-A159A/E
		2902	Figure 64.84 Battery Backup Function Characteristics, changed	
		2903	Table 64.53 Code Flash Memory Characteristics, changed	TN-RX*-A146A/E
		2904	Table 64.54 Data Flash Memory Characteristics, changed	

RX64M Group User's Manual: Hardware

Publication Date: Rev.0.50 Jul 24, 2013
Rev.1.10 Oct 24, 2016

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 4.0

RX64M Group

Renesas Electronics Corporation

R01UH0377EJ0110